Audio Thumbnailing of Popular Music Using Chroma-Based Representations

Mark A. Bartsch, *Member, IEEE,* and Gregory H. Wakefield, *Member, IEEE*

***Abstract*—With the growing prevalence of large databases of multimedia content, methods for facilitating rapid browsing of such databases or the results of a database search are becoming increasingly important. However, these methods are necessarily media dependent. We present a system for producing short, representative samples (or "audio thumbnails") of selections of popular music. The system searches for structural redundancy within a given song with the aim of identifying something like a chorus or refrain. To isolate a useful class of features for performing such structure-based pattern recognition, we present a development of the chromagram, a variation on traditional time-frequency distributions that seeks to represent the cyclic attribute of pitch perception, known as chroma. The pattern recognition system itself employs a quantized chromagram that represents the spectral energy at each of the 12 pitch classes. We evaluate the system on a database of popular music and score its performance against a set of "ideal" thumbnail locations. Overall performance is found to be quite good, with the majority of errors resulting from songs that do not meet our structural assumptions.**

***Index Terms*—Audio summarization, chroma, feature extraction, musical structure, popular music.**

I. 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 returned selections 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. For instance, a common method of browsing a database of images uses smaller, downsampled versions ("thumbnails") of the original images. Downsampling reduces the cost of delivery and display, while the size reduction reduces audition time by increasing the number of selections that may be displayed simultaneously. Alternatively, if a database is predominantly comprised of audio recordings of speech, the most useful encapsulation of a selection is likely a text transcript or similar summarization [1]. Not only is textual information far more compact from a storage and transmission standpoint, but it also can be rapidly skimmed for relevant content.

A database of musical waveforms presents a much different problem. The primary inefficiency that arises from browsing selections from a musical database comes from the time required to listen to each selection. If we simply perform time compression by downsampling the signal, the resulting sound is highly distorted and becomes unintelligible at higher downsampling factors. For speech waveforms, we can sidestep this problem by developing a symbolic representation for the relevant textual content of the signal. An analogous system for music would perform transcription to produce a score representation of the selection. This approach, however, has a number of problems. First, musical transcription is an extremely difficult problem and may be intractable for the general class of musical signals [2]. Even if transcription could be accomplished, the ability to understand a score representation of music, much less to relate it to the aural experience, is relatively uncommon. Finally, one can make strong arguments that a good deal of music is not well represented by a musical score. This is especially true in popular music, where very important elements of a song may include a singer's idiosyncratic vocal style or the particular instruments and processing effects used.

Both the image thumbnail and speech transcript methods effectively produce a representation of the entire selection with reduced detail. For music, a better approach is to produce a short clip of the selection which is in some sense representative of the entire selection. Drawing an analogy to image thumbnails, which quickly convey the "gist" of an image, we call these short clips "audio thumbnails." In classical music, a representative sample might include the introduction of a prominent theme or motif. Popular music, though, is often based on a much simpler structure that might, for instance, alternate between verses and a repeated chorus or refrain. A reasonable strategy for selecting thumbnails in this simpler case involves the location and identification of these repeated sections. Because of its relative simplicity of form, structure-based analysis is more readily accomplished for the class of popular music, and for this reason we restrict our attention to that class.

In this work, we present an algorithm for automatically generating audio thumbnails for selections of popular music. We propose that this problem can reasonably be reduced to the

Manuscript received August 30, 2001; revised April 27, 2003. This work was supported by a National Science Foundation Graduate Research Fellowship awarded to the first author 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 associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hong-Jiang Zhang.

M. Bartsch was with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA. He is now with ATK Mission Research, Beavercreek, OH 45430 USA (e-mail: mbartsch@umich.edu).

G. H. Wakefield is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: ghw@umich.edu).

Digital Object Identifier 10.1109/TMM.2004.840597

1520-9210/$20.00 © 2005 IEEE

problem of isolating repeated musical structures in an audio waveform. We make use of a pattern recognition framework for audio streams, in which the signal is segmented into frames and each frame is described by a set of features. The complexity of the features used varies by application; some commonly used features are described in [3]. This feature-based approach has been applied to general sound classification [4], speech/music discrimination [5], and musical instrument identification [6]. The framework has also been employed for similarity-based musical content retrieval [7], [8]. Finally, a number of systems use these techniques to perform automatic segmentation of audio [9], [10].

This feature-based pattern recognition framework has also been previously applied to the problem of audio thumbnailing. The work of Logan and Chu [11], in particular, employs hidden Markov models and clustering techniques to audio represented by mel-frequency cepstral coefficients (MFCCs). MFCCs are a set of perceptually based spectral features that have been used with great success in speech processing [12]. Foote [13] has suggested audio "gisting" as an potential application for his measure of audio novelty. This measure is calculated from a similarity matrix, which compares features calculated from different frames of audio. Though Foote does not specify the use of any particular set of features, he does recommend the use of MFCCs for computing audio novelty [14].

Standard pattern recognition methods for audio generally rely on broad feature classes that describe some aspect of the timbre or texture of the sound (i.e., brightness, loudness, rate of spectral variation, etc.) or features that model the spectral response of the human auditory system for speech applications (like MFCCs). When dealing with music, however, it is appropriate to use features that specifically address the properties of the musical signals. One of the most salient aspects of musical signals is equivalence of octaves in both melody and harmony. Here, we employ a novel feature class that uses octave equivalence to represent the harmony of a signal. This feature class is fundamentally based on the cyclic attribute of pitch perception, known as chroma.

II. Chroma as a Cyclic Representation of Frequency

In the early 1960s, Shepard reported that two dimensions rather than one are necessary to represent the perceptual structure of pitch [15]. He determined that the human auditory system's perception of pitch was better represented as a helix than as a one-dimensional line, and coined the terms *tone height* and *chroma* to characterize the vertical and angular dimensions, respectively. Fig. 1 shows an illustration of this helix with its two dimensions. In this representation, as the pitch of a musical note increases, say from C1 to C2, its locus moves along the helix, rotating chromatically through all of the pitch classes before it returns to the initial pitch class (C) one cycle above the starting point. According to Shepard's results, the perceived pitch, p, of a signal can be factored into values of chroma, c and tone height h as

$$p = 2^{h+c}. \tag{1}$$

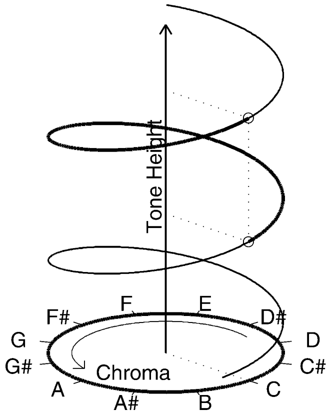

Fig. 1. Illustration of Shepard's helix of pitch perception. The vertical dimension is tone height, while the angular dimension is chroma.

For this decomposition to be unique, it is sufficient for $c \in [0, 1)$ and $h \in \mathbb{Z}$. Linear changes in c result in logarithmic changes in the fundamental frequency associated with the pitch. By dividing the interval between 0 and 1 into 12 equal parts, the 12 pitches of the equal-tempered chromatic scale can be obtained. The implication of Shepard's representation is that the distance between two pitches depends on both c and h, rather than on p alone.

Shepard's factoring is quite intuitive from a musical perspective. In Western music, there is a strong tradition of placing special emphasis on octave relationships between notes of the musical scale. In fact, music theorists use the terms *pitch class* and *octave number* as analogous to Shepard's chroma and tone height. The distinction between pitch class and chroma arises from the discretization of the continuous range of chroma values into 12 distinct pitch classes. It is precisely this relationship between chroma and traditional musical structure that we seek to exploit in the creation of a chroma-based feature class.

A more radical interpretation of Shepard's work was presented in the 1980s by Patterson. In the process of developing his Auditory Image Model in computational audition, Patterson generalized Shepard's results to frequency [16]. Even though he substituted the Archimedian spiral for Shepard's helix as a basic representation of frequency in the auditory system, the mapping from one dimension to two remains effectively the same. His pulse-ribbon model transforms each temporal frame of the auditory image into an activity pattern along a spiral of temporal lags, such that lag values along the same "spoke" of the spiral are octave multiples of each other. This yields a model for frequency that is structurally equivalent to Shepard's decomposition of pitch, such that frequency f is also decomposed as

$$f = 2^{h+c} \tag{2}$$

where we again restrict $c \in [0, 1)$ and $h \in \mathbb{Z}$. Alternately, we can calculate chroma from a given frequency using

$$c = \log_2 f - \lfloor \log_2 f \rfloor \tag{3}$$

where $\lfloor \cdot \rfloor$ denotes the greatest integer function. Thus, chroma is simply the fractional part of the base-2 logarithm of frequency. Similar to ideas of pitch, certain frequencies under this system share the same chroma class if and only if they are mapped to the same value of c. Thus, 200, 400, and 800 Hz all share the same chroma class as 100 Hz, but 300 Hz does not.

A. Chromagram

We have already noted that a strong relationship exists between chroma and the structure of Western music. The previous example, though, suggests that octave-invariant properties of chroma may have utility for a broader class of signals – namely, those composed of harmonic series. In this section, we present a signal analysis technique that exploits the octave-invariant characteristic of chroma. In the following section, we will use this development to examine the properties of such an octave-invariant analysis tool.

With this goal in mind, we first present two definitions. First, we define the *chroma spectrum*, $S(c)$, to be a measure of the strength of a signal with respect to a given value of chroma. This is analogous to the standard Fourier power spectrum of a signal. Then, just as we can extend a spectrum in time to create a time-frequency distribution (TFD) $s(t,f)$, we can also define a "time-chroma" distribution, $s(t,c)$. This distribution, which we call the *chromagram*, is a joint distribution of signal strength over the variables of time and chroma. We define the chromagram as a remapping of a traditional TFD, such as the spectrogram or Wigner distribution, through an aggregation function G, producing

$$s(t,c) = G(s(t,f); \forall f = 2^{c+h}). \tag{4}$$

where $c \in [0,1)$ and $h \in \mathbb{Z}$.

Immediately we see that two independent design decisions are associated with the use of the chromagram. First, we must choose an appropriate aggregation function G. We have found summation to be a good choice, motivated in part by an analogy to spectral aliasing. With such a choice, the chromagram mapping behaves similarly to the mapping from continuous frequency to discrete, cyclic frequency. The primary difference is the logarithmic warping required to map from frequency to chroma. Under such a mapping, we can rewrite the chromagram as

$$s(t,c) = \sum_k s(t, 2^{c+k}) \tag{5}$$

where k ranges over an appropriate set of integers. This computation reduces to a simple weighted sum of frequency-scaled distributions, and it only requires the calculation of a single distribution.

It is also necessary to choose an appropriate TFD, on which to base the chromagram. For many applications, and this work in particular, the spectrogram is a good choice. However, we have also had success with specialized time-frequency distributions such as the modal distribution [17], which permits high-resolution analysis of signals composed of harmonic series.

It is important to note that the octave-invariant properties of the chromagram cannot be obtained from wavelet transformations or time-frequency kernel design. On the one hand, scale-based transformations are octave invariant, but they are also invariant to any other frequency scaling factor. This destroys all chroma relationships within a signal. Kernel design, on the other hand, allows for frequency translation, but this property cannot be sufficiently restricted within the octave. Wakefield [18] presents a detailed derivation of the chromagram which shows the relationship between it and both the scale transform and TFD kernel design.

B. Discrete Chromagram and the Chroma Feature Class

The above section presents the chromagram for continuous time-frequency distributions. When working with sampled signals, however, the TFD is no longer defined on the entire plane but rather is restricted to a regularly spaced lattice in frequency and time. This presents a problem for the logarithmic mapping from a standard TFD to the chromagram. One method for performing this warping in discrete time involves the application of constant-Q transformations; Nelson [19] has presented one such transformation, called the Mellin-wavelet transformation. The resulting warped spectrum translates easily to a chroma-based representation. Another alternative involves the use of centroid calculations to re-localize ridges on the time-frequency surface of spectrally sparse signals, providing improved amplitude and frequency estimates. These refined estimates then allow the accurate computation of associated chroma values.

We have chosen a simpler and more approximate method for several reasons. First, we are seeking a reduced set of features on which we may perform pattern recognition, so a more exact solution is unlikely to be necessary. Additionally, in this work we are examining signals which are highly dynamic and spectrally quite dense; careful refinement of spectral peaks is unlikely to be fruitful in such a context. Finally, our system involves the processing of significant amounts of data at a high sampling rate, so speed of processing is an important factor.

In order to achieve a useful data reduction, we propose a novel feature class based on a coarse quantization of the chromagram. Rather than examine the entire continuum of chroma values in [0, 1), we collect spectral energy into a small number of "chroma bands." All of the spectral energy in a signal, as measured by the spectrogram, is assigned to a chroma band. The energy in each chroma band forms a set of *chroma features* which we can use for pattern recognition. Since we are primarily working with Western music, it seems natural to choose 12 chroma bands, one for each of the 12 traditional pitch classes of the equal-tempered scale. With this in mind, we separate each of our chroma bands by one semitone. Note, however, that this quantization is motivated by the structure of the musical signals being analyzed rather than by the perceptual capabilities of a listener; one semitone is a far coarser quantization than the one required to represent all noticeable differences in chroma.

The chromagram and this quantization in particular have some interesting properties for encoding information about harmonic signals. Let us consider a simple signal composed of a single harmonic series with a fundamental frequency of 220 Hz. The fundamental frequency has a chroma value that is associated with the pitch class A. This same chroma value is shared by the second, fourth, eighth, and 16th harmonics (along with higher octaves of the fundamental). The third harmonic has a chroma value very close to the chroma value of an E; the sixth and 12th harmonics (as well as higher octaves of the third harmonic) share this chroma value. Similarly, harmonics built on the fifth have a chroma value very close to a C♯ while harmonics built on the seventh have a chroma value very close to a D. This mapping is one important feature of the chromagram, and it yields a sort of compression property. Harmonic series with large numbers of different frequencies are represented by many fewer chroma values; further, the first few chroma values represent a proportionately larger number of harmonics than do the later values in the chroma series.

Specifically, consider a harmonic series with 20 components. All of the components of this series are mapped to ten different chroma values, while 13 of these 20 components are mapped to only four different chroma values. The resulting distribution forms a sort of "chroma template" for the harmonic series. Transposition of the harmonic series results in a circular shift of this template around the space of chroma values. One could easily imagine using this property to identify the fundamental chroma value for a particular signal; when coupled with an octave identification scheme, this provides good fundamental frequency recognition in certain cases.

The pitch-class based quantization that we use to generate our feature vectors also has some utility in terms of the above compression property. If we again look at the 20-component harmonic series, we see that all but four of the components have chroma values that fall within 15 cents of the "ideal" chroma values. Thus, relatively little error is introduced by the nominally very coarse quantization of chroma into 12 semitones. Further, the quantization allows the "chroma template" behavior to extend to polyphonic signals. In the continuous chromagram mapping for an A chord, for instance, many components (such as the third harmonic of the chord's root and the fundamental harmonic of the chord's fifth) will have chroma values which are very close but not identical. Under our chromagram quantization, however, these components are mapped into the same chroma band. The resulting representation is significantly simplified, and forms a sort of "chord template" that is mostly invariant to octave or chord inversion. Thus, for instance, we can relate one major chord to another by a simple circular shift of the chord template. In this way, the chroma features can encode and represent harmonic relationships within a particular musical signal.

The representational properties of a feature class based on the quantized chromagram discussed above suggest its utility in structural pattern recognition of popular music. In the following sections, we present and evaluate an algorithm that employs this feature class to derive an "audio thumbnail" from a selection of popular music by isolating repetitive structure in the selection.

III. Algorithm Description and Implementation

The algorithm that we have developed for audio thumbnailing uses a coarsely quantized chromagram as a set of features for pattern recognition. This chromagram is based on the log magnitude spectrogram of the signal under investigation. Essentially, the algorithm identifies extended regions of similarity between different segments of the signal, using feature correlation as a similarity metric. The algorithm consists of five steps: frame segmentation, feature calculation, correlation calculation, correlation filtering, and thumbnail selection. These steps are detailed below.

A. Frame Segmentation

In the first step of the algorithm, we must first define a frame-level segmentation of the song. Rather than using a uniform frame size, as is common for audio processing, we choose to employ beat-synchronous frame segmentation. This allows our frame sampling to track the rhythm of the song. This provides some measure of invariance to tempo changes and tends to align frames in a uniform way. To produce such a segmentation, we preprocess the song of interest using a beat-tracking algorithm. We employ an algorithm developed by Dixon [20], which is particularly suited to popular music. This algorithm is effectively a multi-agent oscillator that responds to impulsive acoustic events (such as a drumbeat). Over our database, the beat-tracking algorithm tends to produce frame sizes between 0.25 and 0.56 s.

It is worthwhile to note that the use of beat-synchronous segmentation is not crucial to the operation of the system, especially since most popular music has a relatively steady tempo. We have performed experiments with this algorithm using a more traditional, uniform frame spacing, and the results are similar for many songs. However, frame segmentations that are not beat-synchronous may produce alignment errors. For instance, if two identical segments of audio are not be separated by an integral number of frames, the misalignment will reduce the computed similarity between these two segments. The resulting reductions in similarity may affect the system's overall performance. The reasonable performance of the system under uniform frame spacing contributes to the robustness of the system with respect to errors of the beat-tracker.

B. Feature Calculation

The second step of the algorithm is feature calculation, during which we calculate a 12-element *chroma feature vector* for each frame. This calculation for the t^{th} frame is based on the logarithmic magnitude of the discrete Fourier transform (DFT), $F_t(n)$. The length of the DFT used is equal to the first power of 2 greater than or equal to the length of the longest frame in the song. The elements of the chroma feature vector for the t^{th} frame $\mathbf{v_t}$ are calculated using the equation

$$\mathbf{v}_{\mathbf{t},k} = \sum_{n \in S_k} \frac{F_t(n)}{N_k}, \quad k \in \{0 \dots 11\} \tag{6}$$

where each $S_k \in \mathbb{Z}$ defines a subset of the discrete frequency space for each pitch class and N_k is the number of elements in S_k. In words, we take the arithmetic mean of all log magnitude DFT bins within a given set S_k. Additionally, we normalize each feature vector by subtracting the scalar mean of that vector's 12 features. Because we are operating in a logarithmic amplitude domain, this operation normalizes the frame in amplitude.

The 12 sets S_k are generated by associating each DFT bin with one of the 12 pitch classes. To make this association, we calculate a DFT bin's associated frequency, then calculate its chroma value using (3). The bin is then associated with the pitch class with the nearest chroma value. For simplicity, we have

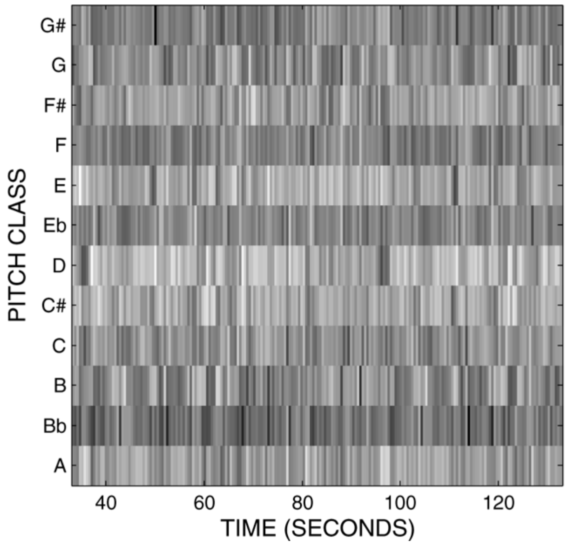

Fig. 2. Portion of the chroma features for Jimmy Buffet's "Margaritaville," showing the energy located in each pitch class for each frame.

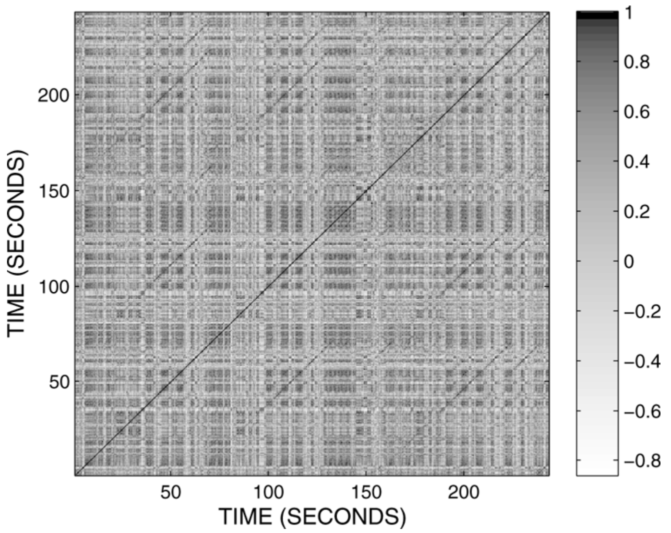

Fig. 3. Similarity matrix C for Jimmy Buffet's "Margaritaville," showing the correlation between the features for individual frames of the song.

reassigned chroma values (which are otherwise arbitrary) so that the pitch class A is centered at a chroma value of 0; then, the other pitch classes are centered at chroma values of $k/12$.

Additionally, we restrict the included range of the spectrum. We set a lower bound at 20 Hz, to correspond with the lower limit of human hearing. An upper bound is set at 2000 Hz. This upper bound is chosen for two reasons. First, the critical bands of the auditory system become broad enough to possibly admit multiple partial frequencies of a harmonic series, which some have argued can effect the perception of chroma. Practically, such a limit is also necessary to prevent the introduction of biases at higher frequencies. These biases arise because the pitch classes immediately below the cutoff have more high-frequency (and, almost universally, low amplitude) bins than the pitch classes immediately above it. This tends to drive the mean for certain features negative. These biases become visibly apparent in the chroma features between 4000 and 8000 Hz.

Fig. 2 shows a portion of the features calculated for Jimmy Buffet's "Margaritaville." Each chroma feature has been labeled with its corresponding pitch class. There are a number of interesting observations that can be made regarding this feature matrix. For instance, one can see a number of chroma features with relatively strong amplitudes (such as A, C♯, D, E, and F♯), as well as several with relatively small amplitudes (such as B♭, E♭, and F). This arrangement of tonal energy is quite consistent with the fact that the song is in the key of D major. Additionally, one can examine the feature vectors themselves and find that many of the vectors manage to loosely track the harmonic transitions in the song. Finally, with careful observation we can visually identify repetitive patterns in the feature vectors of this matrix. For this section of the song, repeats occur with a lag of roughly 60 s.

C. Correlation Calculation

The third stage of the algorithm is the calculation of a similarity matrix, C. Each element of the similarity matrix is equal to the correlation between two feature vectors. This provides a measure of similarity between the corresponding frames in the original signal. In particular, the $(i,j)^{\mathrm{th}}$ element of C is calculated as

$$C_{i,j} = \mathbf{v_i}^T \mathbf{v_j}. \tag{7}$$

We note that the diagonals of C are lines of constant lag in the signal. Thus, extended regions of similarity along any diagonal indicate extended regions of similarity between two portions of the signal.

The similarity matrix for "Margaritaville," C, is shown in Fig. 3. Each element of the matrix indicates the correlation between the chroma vectors for two frames. Here, we can more clearly see interesting structural information about the song. Most obvious are the lines of high correlation along several of the diagonals of the matrix. The main diagonal, as expected, shows unity correlation under zero lag. There are also segments of high correlation along different diagonals of the matrix. These segments indicate repetitions within the song. The block-like structure of the correlation matrix further suggests that there is other structure that we might be able to extract from this matrix. Such an investigation, however, is beyond the scope of this paper.

D. Correlation Filtering

In the third stage of the algorithm, we calculate the similarity between extended segments of the original song that are separated by a constant lag. This is accomplished by filtering along the diagonals of the similarity matrix. Also, the resulting matrix, T, is "rotated" so that the diagonals are oriented vertically. This calculation can be described by the formula

$$T_{i,j} = \sum_{k} C_{i+k,i+j+k} w(k) \tag{8}$$

where $w(k)$ is the windowing function that defines the impulse response of the filter. The $(i,j)^{\mathrm{th}}$ element in T indicates the similarity between the segment of the signal beginning at the i^{th} frame with the segment beginning at the $(i+j)^{\mathrm{th}}$ frame. Thus,

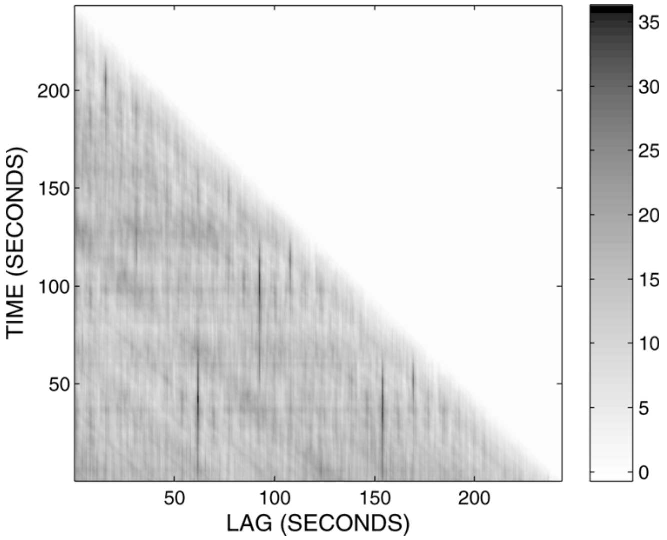

Fig. 4. Filtered time-lag matrix T for Jimmy Buffet's "Margaritaville," showing the similarity between one segment of the song and a segment *lag* seconds ahead of it.

i indicates a time offset, while j indicates the lag that separates the two segments. For this reason, we call T the *filtered time-lag matrix*.

The support size of $w(k)$ determines the length of the segments which are compared, and thus also the length of the selected thumbnail. We have investigated the use of various window functions, both symmetric and nonsymmetric, and have found that a symmetric rectangular window (that is, a uniform moving average filter) yields the best performance in almost all cases. We have left the window's size (and resulting thumbnail length) as a parameter of the system. In our algorithm evaluation, we investigate windows that range from five to 60 s in length.

The filtered time-lag matrix for "Margaritaville," T, is shown in Fig. 4. For this figure, the window length is equivalent to a thirty second interval. Time and lag are plotted along the vertical and horizonal axes, respectively. In this matrix, we can identify strongly repeated sections of the song from vertical ridges on the surface. Though it is not immediately visible in this figure, each of these ridges has a peak near its center that corresponds to the location of greatest similarity.

E. Thumbnail Selection

The final step of the algorithm selects the segment of the song that will be used as the audio thumbnail. To do so, the algorithm selects the maximum value in T subject to a few constraints. The row index of this value and the length of the filtering window $w(t)$ uniquely define the position and length of the selected audio thumbnail. Given the most similar pair of audio segments, our algorithm selects the first of the two audio segments under the assumption that earlier refrains in popular music tend to be less embellished and thus more representative than later ones.

The constraints on our thumbnail selection were identified from errors that the algorithm tended to make on our database of songs (see Section IV). First, music often has a certain degree of local self-similarity that we are not interested in capturing (such repetitions during the song's introduction, for instance, or within one repetition of the chorus). These local repetitions will appear at low lag values, so we place a lower threshold on acceptable lag. We have empirically determined that a reasonable lower limit on lag is one-tenth the length of the song. Additionally, popular music often contains a "fading repeat" at the end of the song; we wish to reduce the system's susceptibility this stylistic technique. To do so, we have found a reasonable upper limit on the starting time of the selection to be three-fourths of the song's length.

The location of the maximum correlation for "Margaritaville" subject to these constraints is found at a starting time of 42 s and a lag of 62 s. Thus, the most strongly repeated section begins 42 s into the song, and this section repeats 62 s later. Since the window used for filtering was 30 s long, we select the portion of the song between [42, 72] as the final thumbnail. This particular thumbnail happens to be the complete chorus of "Margaritaville," and thus it is an ideal selection for this song. In general, however, the system does not always return such perfect results, nor can the selected window size always be equal to the length of the song's chorus. In the next section, we characterize the performance of the system with respect to a large collection of popular music.

IV. Algorithm Evaluation

In order to gain an insight into the performance of this system on a large number of different selections, we have collected a database of popular music for use as a test set. The database is comprised of ninety-three selections of popular music. The database is somewhat biased toward rock music from various eras, but it also contains music from a number of other genres, including folk, country-western, and dance. A number of artists are represented by several selections. To offset the structural ambiguity of some popular music, we have also included in the database a number of contemporary Christian hymns with a 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 a single listener hand-selecting portions of each song as "truth" intervals. The majority of songs contain multiple truth intervals, each of which delimits one repetition of the song's chorus or refrain. 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.

Given a set of truth intervals for a particular song, we can consider how to score the system's selected output interval with respect to these truth intervals. Ideally, the output interval would perfectly match one of the truth intervals. This rarely occurs in practice, so we instead define two criteria that we would like our output intervals to satisfy. First, we would like the output interval to contain as much of a single truth interval as possible. Second, we would like to limit the portion of the selection outside of that truth interval to be as small as possible. We

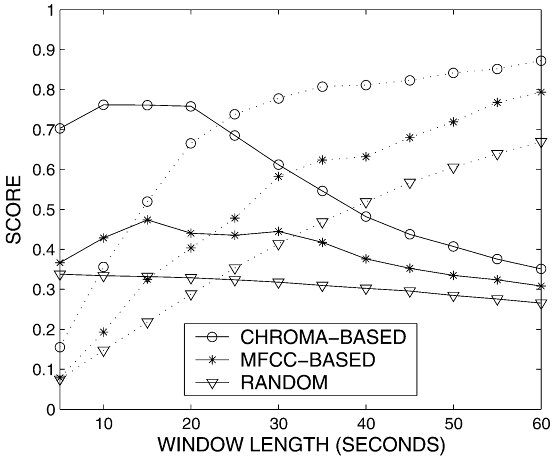

Fig. 5. Average frame-level precision P_p (solid) and recall P_r (dotted) scores plotted versus thumbnail length for the chroma-based algorithm, MFCC-based algorithm, and random thumbnail selection.

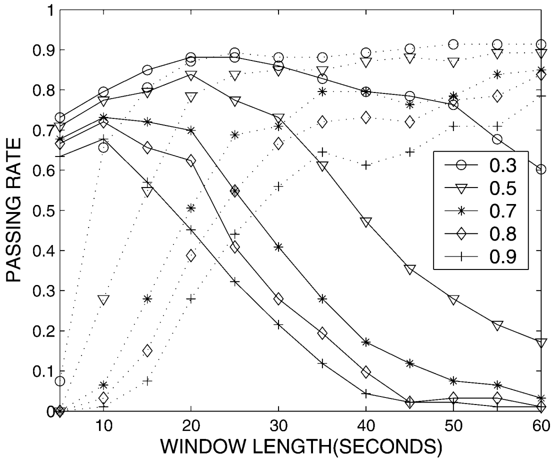

Fig. 6. Percentage of songs with precision, P_p (solid) and recall P_r, (dotted) scores that exceed various thresholds plotted versus thumbnail length.

define both of these criteria based on a single truth interval to discourage intervals that, for instance, are large enough to contain two repetitions of a chorus or contain the end of one chorus and the beginning of an adjoining repetition.

The two criteria regarding interval overlap translate directly into two scoring functions. Given an output interval x and a set of k truth intervals for a particular song, $\{z_i\}_{i=1}^{k}$, we can express the frame-level recall, P_r, as

$$P_r = \frac{|x \cap z_i|}{|x|}, \qquad i = \underset{j}{\text{argmax}} \, |x \cap z_j| \tag{9}$$

where we use $|\cdot|$ to denote the length of an interval. Similarly, we can express the frame-level precision P_p as

$$P_p = \frac{|x \cap z_i|}{|z_i|}, \qquad i = \underset{j}{\text{argmax}} \, |x \cap z_j| \, . \tag{10}$$

Both of these scores will always lie within the interval [0, 1].

In order to examine the performance of our thumbnailing algorithm, we examine the precision and recall scores over a range of windows lengths for each song in our database. The resulting scores are averaged to provide a mean precision and recall score for each window length. We also compared the chroma-based algorithm to two alternative thumbnailing algorithms. The first employs the algorithm presented in Section III, but substitutes the commonly-used mel-frequency cepstral coefficients [12] for our chroma features. The second is a random algorithm that selects any thumbnail with a particular length with equal probability. This random algorithm is run 1000 times on each song and the results are averaged to find the mean score that results under a "chance" decision. Fig. 5 shows the mean precision and recall scores for three audio thumbnailing systems plotted versus desired thumbnail length.

From this figure, we can see that both the chroma-based and MFCC-based implementations of the algorithm perform significantly better than chance for both scores and over all of the window lengths plotted here. Additionally, it is clear that the chroma-based algorithm produces significantly higher precision and recall scores than the MFCC-based algorithm.

This figure also shows the clear tradeoff between the precision and recall scores with window length. One interesting feature is the intersection of the score curves for all three algorithms. All three sets of curves intersect around 23 s, which is the mean length of all truth intervals in the database. This should be expected; when the selection interval is equal to the corresponding truth interval, the expressions for P_p and P_r become equivalent. Therefore, if we wish to maximize both of these scores simultaneously over a given database, we should choose a window length equal to the mean length of choruses and refrains in that database.

Another useful measure of performance is the fraction of the songs with a score exceeding some threshold, or the "passing rate." This provides more detailed information about the performance of a given algorithm. Fig. 6 displays the chroma-based algorithm's passing rate for both P_p and P_r 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 s. 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 20–25 s.

So far, we have seen statistics regarding the system's performance, and we have seen one example where the system performs very well. It is also instructive to examine the system's failure and the causes of such. One trivial reason for failure (which occurs exactly once in our database) is the case in which the chorus is clearly distinguished from the verses, yet is not repeated. This violates our original assumptions, and as such we cannot expect correct selections in these cases. A far more 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. A slightly less common version of this same error occurs when the repetition of some "uninteresting" portion of the song has a high enough correlation to overshadow the repetitions of the chorus. This is often an instrumental section, such as the introduction of the song.

V. 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. It is reasonable to argue that the scoring methodology is somewhat too restrictive while simultaneously being rather arbitrary. After all, if the system selects a verse because it repeats strongly, then one could make the argument that the verse may be a good (if not necessarily ideal) choice after all. Much of this confusion results from the attempt to quantify something which is fundamentally a subjective evaluation.

The most important conclusion that can be drawn from these results relates to the potential of chroma-based representations for encoding musical structure. We have seen that the system successfully highlights repeated passages in a selection, and the success of the system in the general case indicates that the chroma-based representation is sufficient to represent redundant structures within a given song. A detailed subjective examination of the chroma features indicates that the chroma features do in fact encode harmonic relationships. Further, our results have shown that the presented algorithm has superior performance when employing chroma-based features than when using MFCCs, which are often recommended for use in audio segmentation and classification. This shows that chroma-based representations can be very useful for structural analysis in music.

One might question how well this system would work on other collections of music, possibly with music of other styles and genres. The easiest way to address this question is to refer back to the initial assumptions on which the system was built. Does a particular type of music have strongly repeated sections which identify important portions of the music? Consider the majority of Western classical music. Much of this music is characterized by the repetition of themes within various musical contexts. Because of how these contexts vary, the system will generally not perform well on such music. Similarly, the improvizational emphasis in jazz and blues will cause degradation in the system's performance on these types of music, despite a strong repetitive structure. In these alternative types of music, the criteria by which we would wish to select a "thumbnail" may be quite different. If a selection of 12-bar blues simply repeats over and over, what distinguishes a "good" thumbnail? Further, how should one quantitatively evaluate such a selection?

There are a number of areas for future work with regard to this system. First, some means of optimizing over window size would be useful. Another valuable addition would allow the system to use more information than simply the strongest pairwise match, as it currently does. Such a modification could allow the detection of multiple repetitions within a song and use this information to make a better decision about the most important selection. This would also be a first step in extending this system beyond just thumbnailing to the segmentation of a song based on its musical structure and would complement systems that perform segmentation based on sound type [10] or sound "texture" [9].

Another valuable extension of this work would generalize it beyond simple measures of internal redundancy to measures of similarity between different songs. Currently, we have not investigated the use of our chroma features for cross-song comparison. However, such comparison has clear applications for audio database search-and-retrieval systems. One possibility would be to find songs that are similar based on harmonic and timbral structure as encoded in a chroma-based representation. Another possible line of investigation would examine the usefulness of chroma-based representations in comparing "sophisticated" queries to musical databases. It is conceivable that the chroma features employed in this system may encode sufficient information from a singer-plus-harmony query to allow for full-audio search and retrieval.

VI. Conclusion

We have presented a system which uses a novel chroma-based representation of sound to isolate repetitions 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. In particular, for audio thumbnailing chroma-based features are shown to have superior performance to the commonly-used mel-frequency cepstral coefficients. This system provides a first step toward using chroma-based representations as an important element of more sophisticated analysis systems, including segmentation and search-and-retrieval.

Acknowledgment

The authors would like to thank members of the MusEn Project, W. Birmingham, B. Pardo, C. Meek, and M. Mellody, for their comments and contributions to this work.

References

[1] J. Hirschberg, S. Whittaker, D. Hindle, F. Pereira, and A. Singhal, "Finding information in audio: A new paradigm for audio browsing and retrieval," in *Proc. ESCA Workshop: Accessing Information in Spoken Audio*, Cambridge, U.K., 1999, pp. 117–122.

[2] D. B. Gerhard, "Computer Music Analysis," Simon Fraser Univ. School of Comput. Sci., Surrey, U.K., Tech. Rep. CMPT TR 97-13, 1997.

[3] G. Tzanetakis and P. Cook, "A framework for audio analysis based on classification and temporal segmentation," in *Proc. EUROMICRO; Informatics Theory and Practice for the New Millennium*, Milan, Italy, 1999, pp. 61–69.

[4] J. Foote, "A similarity measure for automatic audio classification," in *American Association for Artificial Intelligence: Intelligent Integration and Use of Test, Image, Video and Audio Corpora*, Stanford, CA, 1997, pp. 1–7.

[5] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, Munich, Germany, 1997, pp. 1331–1334.

[6] K. Martin, E. Schreirer, and B. Vercoe, "Music content analysis through models of audition," in *ACM Multimedia Workshop on Content Processing of Music for Multimedia Applications*, Bristol, U.K., 1998.

[7] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, pp. 27–36, 1996.

[8] M. Welsh, N. Borisov, J. Hill, R. von Behren, and A. Woo, "Querying Large Collections of Music for Similarity," Univ. California Berkeley Comput. Sci. Division, Berkeley, CA, Tech. Rep. UCB/CSD00-1096, 1999.

[9] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for browsing and annotation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 1999, pp. 103–106.

[10] D. Kimber and L. Wilcox, "Acoustic segmentation for audio browsers," in *Proc. Interface Conf.*, L. Billard and N. I. Fisher, Eds., Sydney, Australia, 1996, pp. 295–304.

[11] B. Logan and S. Chu, "Music summarization using key phrases," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, Istanbul, Turkey, 2000, pp. II–749–752.

[12] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[13] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. I, 1999, pp. 452–455.

[14] ——, "Visualizing music and audio using self-similarity," in *ACM Int. Multimedia Conf.*, Orlando, FL, 1999, pp. 77–80.

[15] R. N. Shepard, "Circularity in judgements of relative pitch," *J. Acoust. Soc. Amer.*, vol. 36, pp. 2346–2353, 1964.

[16] R. D. Patterson, "Spiral detection of periodicity and the spiral form of musical scales," *Psychol. Music*, vol. 14, pp. 44–61, 1986.

[17] W. J. Pielemeier and G. H. Wakefield, "A high-resolution time-frequency representation for musical instrument signals," *J. Acoust. Soc. Amer.*, vol. 99, pp. 2382–96, 1996.

[18] G. Wakefield, "Mathematical representation of joint time-chroma distributions," in *Proc. SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations IX*, vol. 3807, 1999, pp. 637–645.

[19] D. Nelson, "Mellin-wavelet transform," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 1995, pp. 1101–1104.

[20] S. Dixon, "A lightweight multi-agent musical beat tracking system," in *Proc. Pacific Rim Int. Conf. Artificial Intelligence*, R. Mizoguchi and J. K. Slaney, Eds., Melbourne, Australia, 2000, pp. 778–788.

Mark A. Bartsch (S'96–M'04)) recieved the B.E.E. degree (*summa cum laude*) from the University of Dayton, Dayton, OH, in 2000, and the M.S.E. and Ph.D. degrees in electrical engineering systems from The University of Michigan, Ann Arbor, in 2002 and 2004, respectively.

He is currently with ATK Mission Research, Beavercreek, OH. His research interests include pattern recognition and machine learning for signal processing applications.

Gregory H. Wakefield (M'85) received the B.A. degree (*summa cum laude*) in mathematics and psychology, the M.S. and Ph.D. degrees in electrical engineering, and the Ph.D. degree in psychology, all from the University of Minnesota, Minneapolis, in 1978, 1982, 1985, and 1988, respectively.

In 1986, he joined the faculty of the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, where he is currently an Associate Professor. His research interests include time-frequency representations, music signal processing, auditory systems modeling, psychoacoustics, sensory prosthetics, and sound quality engineering. He serves as consultant to various industries in sound-quality engineering and time-frequency representations.

Dr. Wakefield received the National Science Foundation's Presidential Young Investigator Award in 1987 and the IEEE Millennium Award in 2000.