Automatic Music Summarization via Similarity Analysis

Matthew Cooper
FX Palo Alto Laboratory
3400 Hillview Avenue
Bldg. 4
Palo Alto, CA 94304 USA
cooper@fxpal.com

Jonathan Foote
FX Palo Alto Laboratory
3400 Hillview Avenue
Bldg. 4
Palo Alto, CA 94304 USA
foote@fxpal.com

ABSTRACT

We present methods for automatically producing summary excerpts or thumbnails of music. To find the most representative excerpt, we maximize the average segment similarity to the entire work. After window-based audio parameterization, a quantitative similarity measure is calculated between every pair of windows, and the results are embedded in a 2-D similarity matrix. Summing the similarity matrix over the support of a segment results in a measure of how similar that segment is to the whole. This measure is maximized to find the segment that best represents the entire work. We discuss variations on the method, and present experimental results for orchestral music, popular songs, and jazz. These results demonstrate that the method finds significantly representative excerpts, using very few assumptions about the source audio.

1. INTRODUCTION

As digital audio collections grow in size and number, audio summarization, or "thumbnailing" has become an increasingly active research area. Audio summaries are useful in applications such as e-commerce and information retrieval, because of the large file sizes and high bandwidth requirements of multimedia data. Quite often it is not practical to audit an entire work, for example if a music search engine returns many results each lasting several minutes. A representative excerpt that gives a good idea of the work is thus is desirable. Similarly, e-commerce music sites often make short song segments available to preview before purchase. In an audio retrieval system, it may make sense to judge the similarity of representative excerpts of a work rather than the work as a whole, especially if the analysis is computationally expensive. There is no point analyzing an entire symphony if a reasonable index can be derived from a ten-second excerpt.

For these applications, the segment must be a good representation of the longer work. However, existing segmentation and excerpting algorithms do little to guarantee this. Indeed, some approaches can be as crude as to present, for example, the first thirty seconds of an audio track as representing the whole work. This can be highly unsatisfactory if, for instance, the bulk of a particular track bears little resemblance to an idiosyncratic introduction.

We present a method for automatically producing excerpts of linear media (where "linear" implies a function of a one-dimensional variable). Examples include audio and video, which are functions of time, and text, which is a discrete function of file position. We construct summaries using self-similarity analysis, which allows us to study the structure in an audio file by measuring the pairwise similarity between audio instants.

Here we assume the optimal excerpt is most similar, in an average sense, to the piece as a whole. This approach does not rely on semantic content that can't be automatically extracted, and thus cannot be considered optimal in that sense. For example, a summary of the first movement of Beethoven's Fifth Symphony without the famous four-note theme would not be ideal by most standards. To rectify this, the process can be weighted to reflect any available semantic information. Another possible desiderata for an audio summary is that it contain all representative portions. For example, a popular song containing verses, refrains and a bridge should arguably be summarized by an example containing portions of all three segments. This is generally not possible with a short, contiguous excerpt. In this paper, our summaries will be continuous excerpts that are typically much shorter than the source audio.

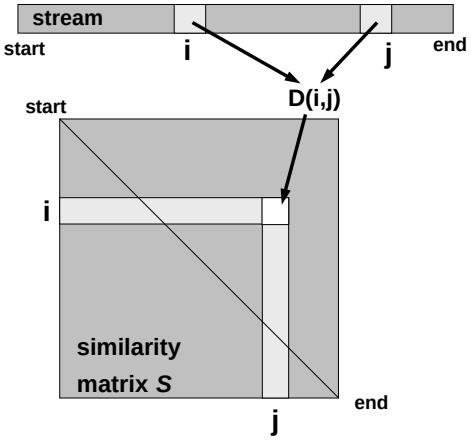

Figure 1: Embedding pairwise similarity data in the similarity matrix.

1.1 Related Work

There is a great deal of related work on summarization techniques for text, audio, and video. Most summarization or excerpting techniques start with an analysis of the structure or semantics of the source material. The work on statistical text summarization uses term frequency/inverse document frequency (tf/ idf) to select paragraphs [1], sentences [9], or key phrases that are both representative of the document and differentiate it from other documents. Audio summarization techniques typically use a segmentation phase followed by extraction of a representative excerpt from each segment. A subset of these excerpts are combined to summarize the audio [6]. The work on scene transition graphs is a typical approach to abstracting video [8]. After video frames are clustered, the keyframe closest to each cluster center is chosen to represent that cluster. Other approaches attempt to summarize video using various heuristics, often derived from an analysis of accompanying closed captions [2]. In contrast, our summarization method is not based on any prior segmentation or segment clustering. The resulting summaries are selected to maximize quantitative measures of

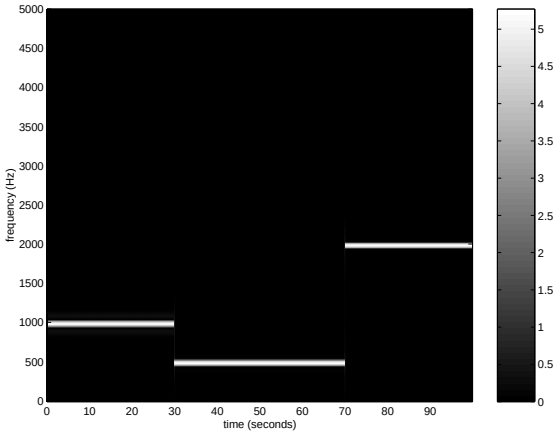

Figure 2: Spectrogram data computed from artificial three-tone audio data.

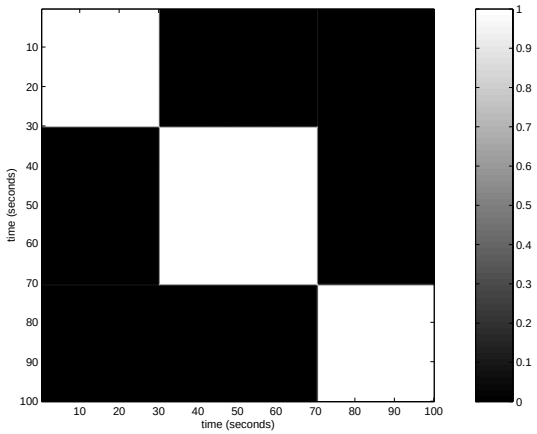

Figure 3: Similarity matrix for synthetic signal of Figure 2.

the similarity between candidate excerpts and the source audio as a whole. Summaries of any desired length can be extracted, to support browsing at varying levels of detail.

2. SELF-SIMILARITY ANALYSIS

In our analysis, the first step is to parameterize the audio. This is typically done by windowing the audio waveform. Currently, we convert the source audio to a 22.05 KHz mono format by resampling or decoding a compressed format like MP3 (MPEG Layer 2 Level 3). We then subdivide the audio into 2048 sample (92.87 ms) "frames" at a 10 Hz rate. Each frame is then windowed with a Hamming window, and parameterized using Mel-Frequency Cepstral Coefficient (MFCC) analysis (e.g. [7]). We have also successfully employed standard spectrogram-based parameterizations. We achieve similar performance by reducing the dimension of uniformly-spaced 64 bin spectrograms via singular value decomposition (SVD). We are currently using MFCCs as they provide a low dimensional data-independent parameterization, which may be efficiently calculated using freely available software. Many compression techniques such as MP3 contain similar spectral representations which could be used directly, avoiding the expense of audio decoding and reparameterizing. Regardless of the parameterization, the result is a compact vector of parameters for every frame. Figure 2 shows the spectrogram for a synthetic three-tone test signal. This was generated by concatenating 30 seconds of a 1 kHz sine wave, 40 seconds of 500 Hz, and 30 seconds of 2 kHz to result in a one-hundred second signal. Because the 500 Hz portion is the longest, the ideal summary should consist primarily of the 500 Hz signal as opposed to the shorter 1 KHz and 2 KHz segments.

2.1 Distance Matrix Embedding

Once the signal has been parameterized, it is then embedded in a two-dimensional representation [3, 4]. The key is a measure d of the (dis)similarity between pairs of parameter vectors v_i and v_j calculated from frames i and j. A simple distance measure is the Euclidean distance in the L-dimensional parameter space:

$$d_e(v_i, v_j) = \sqrt{\sum_{l=1}^{L} (v_i(l) - v_j(l))^2} \quad . \tag{1}$$

Another useful similarity measure is the scalar (dot) product of the vectors. The dot product can be normalized to give the cosine of the angle between the parameter vectors:

$$d_c(v_i, v_j) = \frac{< v_i, v_j >}{\|v_i\| \ \|v_j\|} \quad . \tag{2}$$

This is the cosine of the angle between the vectors and has the property that it yields a large similarity score even if the vectors are small in magnitude. For most applications, this significantly improves performance over the Euclidean distance measure by removing dependence on signal energy.

The distance measure is a function of two frames, hence instants in the source audio. We consider the similarity between all possible instants in a signal by embedding the distance measure in a two-dimensional similarity matrix, as depicted in Figure 1. The matrix $\mathbf{S}$ contains the similarity computed between all frame combinations, such that the element at the i^{th} row and j^{th} column is

$$\mathbf{S}(i, j) = d_c(v_i, v_j) \quad . \tag{3}$$

In general, $\mathbf{S}$ will have maximum values on the diagonal (because every window will be maximally similar to itself); furthermore if d is symmetric then $\mathbf{S}$ will be symmetric as well.

2.2 Visualizing Similarity Matrices

We visualize $\mathbf{S}$ by assigning a brightness proportional to the similarity measure $d_c(i, j)$ to each pixel (i, j). The resulting image reveals the structure of the source audio. Regions of high self-similarity appear as bright squares along the main diagonal. Repeated sections produce bright off-diagonal rectangles. If the work has a high degree of repetition, this will be visible as diagonal stripes or rectangles, offset from the main diagonal by the repetition time. The similarity matrix for the synthetic three tone signal is shown in Figure 3. Each portion of the signal is visible as self-similar white squares on the diagonal. For example the 500 Hz tone extends from 30 seconds to 70 seconds on both time axes.

3. AUTOMATIC SUMMARIZATION

To find the segment of a work that best represents the entire work, we wish to find the segment with maximum similarity to the whole. In popular music, which commonly contains repeated elements such as verses or choruses, we expect that the song's most-repeated or longest element will appear in the summary. This element is determined from the similarity matrix.

A simple example will illustrate the approach. Given the sequence ABBBCC, we wish to find the most representative subsequence of length three. For simplicity, the similarity measure is chosen to be one if the sequence members match and zero otherwise. We can compute $\mathbf{S}$ using a Hamming-like metric such that the distance between two sequence elements is one if the elements are the same,

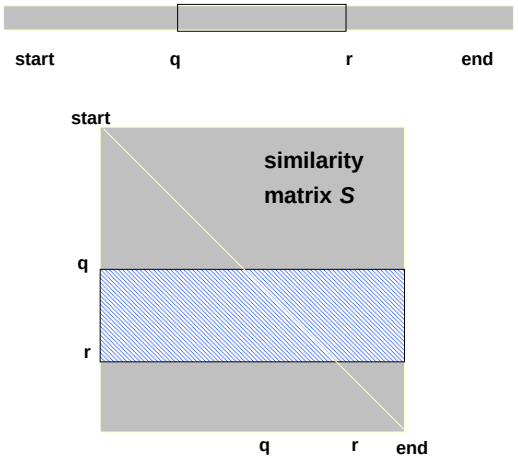

Figure 4: Calculating $\bar{\mathbf{S}}(q,r)$ by summing the similarity matrix over the support of the excerpt $q,\cdots,r$.

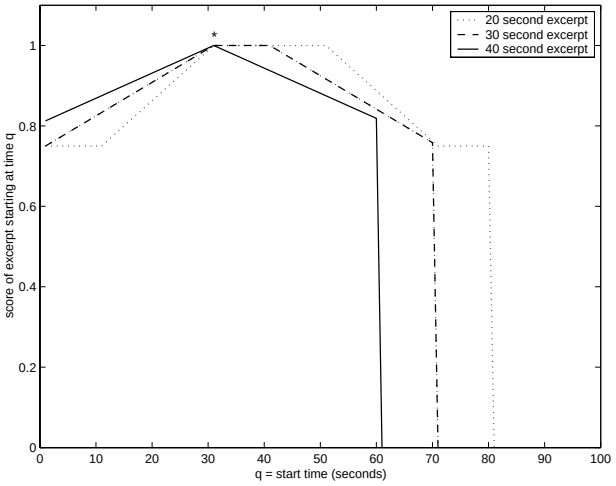

Figure 5: Summary scores $Q_L(i)$ computed from the similarity matrix of Figure 3 for $L = 20, 30,$ and 40 seconds.

and zero otherwise:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{4}$$

For any subsequence, the average similarity between the subsequence and the entire sequence can be found by summing the columns (or equivalently, rows) of $\mathbf{S}$ corresponding to that subsequence. In our example, we want to find the three-element subsequence with maximal average similarity. There are four possible subsequences: ABB, BBB, BBC, and BCC with column sums seven, nine, eight, and seven, respectively. The highest scoring subsequence is BBB, with a score of nine. This is the optimal three-element contiguous summary of the sequence ABBBCC. Note that this contains all the most frequent members (B) of the longer sequence. The runner-up sequence is BBC with a score of eight, which contains both the most frequent and second-most frequent members. The score can be normalized by the subsequence length so that summaries of different lengths can be compared.

The previous example can be generalized to arbitrary sequence lengths. Given a segment starting at q and ending at r, the average similarity of the segment is calculated as the sum of the self-similarity between the segment and the entire work, normalized by the segment length:

$$\bar{\mathbf{S}}(q,r) = \frac{1}{N(r-q)} \sum_{m=q}^{r} \sum_{n=1}^{N} \mathbf{S}(m,n), \tag{5}$$

where N is the length of the entire work (width and height of $\mathbf{S}$). This is shown schematically in Figure 4. A simple interpretation of $\bar{\mathbf{S}}(q,r)$ is the average of similarity matrix rows over the interval $q,\cdots,r$ (or equivalently, the columns). Thus intervals with large similarity to the work as a whole will have a larger average $\bar{\mathbf{S}}(q,r)$.

If a weighting function w is known, it can be applied to find a weighted average as:

$$\bar{\mathbf{S}}_w(q,r) = \frac{1}{N(r-q)} \sum_{m=q}^{r} \sum_{n=1}^{N} w(n)\mathbf{S}(m,n). \tag{6}$$

This can be maximized to find the optimal weighted summary. Typical weighting functions might include a w that decreases with time, so segments at the beginning of the work are weighted more highly than those at the end. Alternatively, w might include a measure of loudness, favoring generally louder sections such as *tutti* (all instruments playing) or choruses rather than verses. Any other information known *a priori* or deduced can be incorporated into w.

To find the optimal summary of length L, we find the excerpt of that length with the maximum summary score (Eq. (5)). Define the score $Q_L(i)$ as

$$Q_L(i) = \bar{\mathbf{S}}(i,i+L) = \frac{1}{NL} \sum_{m=i}^{i+L} \sum_{n=1}^{N} \mathbf{S}(m,n) \tag{7}$$

for $i = 1,\cdots,N-L$. The best starting point for the excerpt is the time q_L^* that maximizes the summary score:

$$q_L^* = \underset{1\le i\le N-L}{\text{ArgMax}}\, Q_L(i). \tag{8}$$

The best summary is then the excerpt of length L starting at q_L^* and ending at time $q_L^* + L$.

Figure 5 shows values of $Q_L(i)$ versus start time i, for summary lengths L of 20, 30, and 40 seconds. All show a maxima or peak at $q^* = 30$, which is the start time of the 500 Hz tone (the most representative segment of the work). The $L = 20$ curve has a maximum that extends from 30 seconds to 50 seconds, because any 20 second excerpt starting within that interval will consist solely of the 500 Hz representative tone. Thus for this example and segment lengths, picking any maximal point q_L^* results in an excerpted segment that consists completely of the 500 Hz tone. This is the desired behavior in a quasi-probabilistic sense: any infinitesimally short sample taken uniformly randomly from the the source signal will result in a 500 Hz tone 40% of the time, and 1 or 2 kHz only 30%. Thus the selected segment contains the excerpt most likely to be similar to samples from the original signal.

4. EXPERIMENTS

4.1 Music Visualization

Figure 6 shows a visualization for the Spring (allegro) movement from Vivaldi's *The Four Seasons*. The 22.05 KHz audio was windowed at a 10 Hz rate. For each window, we computed 45 MFCCs. We then ordered the MFCCs according to their variances across the entire piece, and retained the fifteen coefficients with highest variances. We scaled these coefficients to unit variance (hence zero

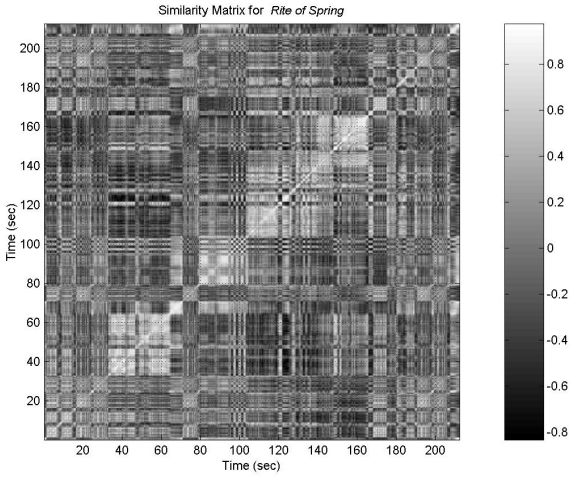

Figure 6: Similarity matrix computed for Vivaldi's *Spring* using MFCC features and cosine similarity measure. The opening theme is repeated at 72 and 190 seconds.

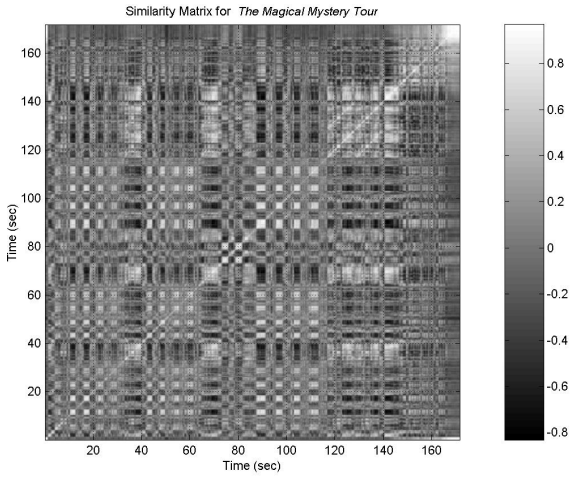

Figure 7: Similarity matrix computed for *The Magical Mystery Tour* using MFCC features and cosine similarity measure.

mean) across the piece, and then calculated the pairwise similarity using the cosine distance of (2). (We discarded the low variance coefficients as they provide poor discrimination between the structural elements of a piece. Scaling them to unit variance will generally amplify noise and in turn degrade the similarity analysis.)

The resulting similarity matrix shows the familiar opening theme in the first sixteen seconds. It is repeated twice, first *forte* (loudly) then a quieter repeat eight seconds later. Both repetitions look similar because of the cosine similarity measure. The theme is repeated at seventy-two seconds and one hundred ninety seconds, which can be seen as brighter regions along the bottom of the image. The major structure of the piece is also evident in the blocks along the main diagonal. For example the bright block between 30 and 70 seconds is a soft passage for two violins, with the rest of the ensemble quiet.

Figure 7 shows the similarity matrix computed from *The Magical Mystery Tour* by The Beatles, using the same parameterization as the Vivaldi. The bright white squares of high similarity show repeated instances of the song's familiar chorus ("Roll up, roll up for the Mystery Tour") throughout the song. The piece also features a distinctive coda from 145 - 167 seconds, which differs substantially from the majority of the song.

4.2 Music Summarization

We use the similarity matrix to determine the optimal summaries in two steps. The first step is to evaluate the summary score $Q_L(i)$ of (7). Next, we maximize this to find the best start point q_L^* of (8). Computing the column sums of of $\mathbf{S}$ in advance can reduce computation and storage requirements. For example, in an application where variable length summaries of songs are provided depending on available bandwidth, the column sums for each song can be pre-computed and stored, and $\mathbf{S}$ can be discarded. Given a desired summary length, L, Q_L can be computed using solely the column sums (the inner sum of (5)), and maximized to determine the summary excerpt. The column sums computed from the matrix of Figure 7 appear in Figure 8.

Figure 9 shows the summary scores $Q_{10}(i), Q_{20}(i)$, and $Q_{30}(i)$ computed by summing the columns of the similarity matrix for for *The Magical Mystery Tour* (Figure 8). Locating the maxima in these

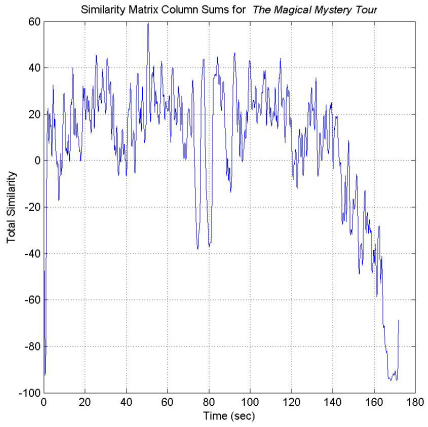

Figure 8: Column sums computed from the similarity matrix of Figure 7.

curves, per (8), produces the optimal summary start points shown in Table 1. The twenty second summary includes the ten second summary and contains the familiar title refrain. The thirty second summary, interestingly, is a repeat of this element of the song, but from later in the piece (after the bridge). This excerpt was selected because it is a longer reprisal of the same title refrain than is available at the beginning of the song.

Table 2 shows the optimal summaries computed for Vivaldi's *Spring*. All three summaries include the memorable introductory theme. The ten second summary is the first 10 seconds of the theme. The 20-second summary includes the last three repetitions. The 30-second summary includes virtually the entire introduction, which exhibits the highest average similarity with the overall piece.

We also present summaries for two additional songs. Table 3 shows three summaries computed for *Wild Honey* by the band U2. All three summaries include the song's longest chorus segments. The chorus is about 15 seconds long, so the first summary only contains a portion of it, while the longer two summaries contain at least one of its repetitions in its entirety. Table 4 shows three summaries computed for *Take the "A" Train* performed by Duke Ellington and his orchestra. Again, all three summaries contain the same portion of the piece; in this case it is a reprisal of the song's main melody at the performance's end. In each of the four cases presented, the

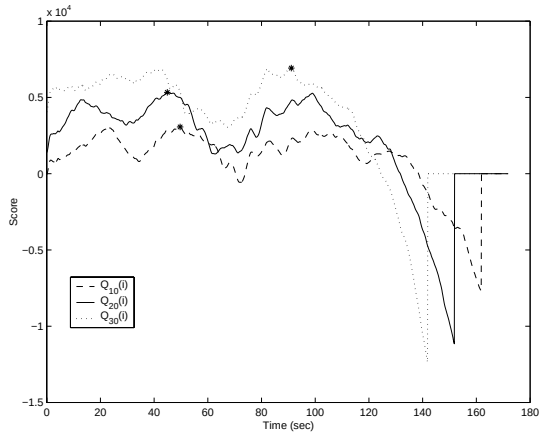

Figure 9: Summary scores, $Q_L(i)$, computed from the similarity matrix of Figure 7 for $L = 10, 20,$ and 30 seconds.

resulting summaries make intuitive sense, and represent significant and memorable elements of the original pieces. These results are promising, and more thorough validation of the approach is planned. The results presented here are available for review on the world wide web[1].

Table 1: Summary times for *The Magical Mystery Tour*.

Segment Length	Start (sec.)	End (sec.)
10	49.7	59.7
20	44.9	64.9
30	91.1	121.1

Table 2: Summary times for *Spring*.

Segment Length	Start (sec.)	End (sec.)
10	4.3	14.3
20	8.4	28.4
30	2.5	32.5

5. CONCLUSION

We have presented a quantitative approach to automatic music summarization which makes minimal assumptions regarding the source audio. (Indeed, we expect this approach to work for MIDI, video, and other time-dependent media as well). The pairwise similarity of the audio feature is embedded in a similarity matrix which reveals the major structure of the audio. By summing the similarity matrix columns, the most representative contiguous portions of the piece can be located and used for summaries of arbitrary length. We have presented experimental results across a variety of genres, and in each case, the resulting summaries represented significant elements of the original piece. While this approach will not always yield intuitively satisfying results, we argue that it will find the summary that is most likely to be similar to the work as a whole. This excerpting criteria is particularly well-suited to popular music which exhibits a relatively high degree of repetition of songs' principal segments.

We are currently integrating this summarization approach with audio segmentation [5] such that summaries will begin and end at meaningful segment boundaries (such as verse/chorus transitions). We are also examining joint segmentation and summarization techniques to develop a more complete structural characterization of digital audio.

[1] http://www.fxpal.com/media/musicthumbnails.html

Table 3: Summary times for *Wild Honey*.

Segment Length	Start (sec.)	End (sec.)
10	197.1	207.1
20	189.6	209.6
30	181.7	211.7

Table 4: Summary times for *Take the "A" Train*.

Segment Length	Start (sec.)	End (sec.)
10	135.2	145.2
20	136.7	156.7
30	135	165

6. ACKNOWLEDGMENT

The authors acknowledge John Boreczky's help developing the original idea for summarization based on similarity matrices.

7. REFERENCES

[1] Abracos, J. and Lopes, G.P., Statistical methods for retrieving most significant paragraphs in newspaper articles, In *ACL/EACL Workshop on Intelligent Scalable Text Summarization*, pages 51-57, 1997.

[2] Christel, M., Stevens, S., Kanade, T., Mauldin, M., Reddy, R., and Wactlar, H. Techniques for the Creation and Exploration of Digital Video Libraries. In *Multimedia Tools and Applications*, B. Furht, ed. Kluwer Academic Publishers, 1996.

[3] Eckman, J.P., et al., Recurrence Plots of Dynamical Systems, in *Europhys. Lett.* **4**(973) (1 November 1987).

[4] Foote, J. Visualizing Music and Audio using Self-Similarity. In *Proc. ACM Multimedia 99*, pp. 77-80, Orlando, Florida, November 1999.

[5] Foote, J., Automatic Audio Segmentation Using A Measure of Audio Novelty, *Proc. ICME 2000*, vol. I, pp. 452-455, 2000.

[6] Logan, B. and Chu, S., Music Summarization Using Key Phrases, in *Proc. IEEE ICASSP*, 2000.

[7] Rabiner, L. and Juang, B.-H., *Fundamentals of Speech Recognition*. Prentice Hall PTR. ISBN 0-13-015157-2, 1993.

[8] Yeo, B.-L., Yeung, M.M., Classification, Simplification, and Dynamic Visualization of Scene Transition Graphs for Video Browsing, in *Storage and Retrieval for Image and Video Databases (SPIE)* pp. 60-70, 1998.

[9] Zechner, K. Fast generation of abstracts from general domain text corpora by extracting relevant sentences, In *Proc. of the International Conference on Computational Linguistics*, 1996.