A Content-Based Music Similarity Function

Beth Logan   *Ariel Salomon*

Cambridge Research Laboratory
Technical Report Series

CRL 2001/02
June 2001

COMPAQ

A Content-Based Music Similarity Function

Beth Logan
Cambridge Research Laboratory
Compaq Computer Corporation
Cambridge MA 02139

Ariel Salomon
Massachusetts Institute of Technology
Cambridge MA 02139

June 2001

Abstract

We present a method to compare songs based solely on their audio content. Our technique forms a signature for each song based on K-means clustering of spectral features. The signatures can then be compared using the Earth Mover's Distance [14] which allows comparison of histograms with disparate bins. Preliminary objective and subjective results on a database of over 8000 songs are encouraging. For 20 songs judged by two users, on average 2.5 out of the top 5 songs returned were judged similar. We also found that our measure is robust to simple corruption of the audio signal and that meaningful visualizations of the data are possible using this similarity measure.

Author email: Beth.Logan@compaq.com, asalomon@mit.edu

CRL Technical reports are available on the CRL's web page at
http://crl.research.compaq.com.

Compaq Computer Corporation
Cambridge Research Laboratory
One Cambridge Center
Cambridge, Massachusetts 02142 USA

1 Introduction

The advent of MP3 and other efficient compression algorithms is changing the world of music distribution. We are moving toward a future in which all the world's music will be ubiquitously available. Additionally, the 'unit' of music has changed from the album to the song. Thus users will soon be able to search through vast databases at the song level.

These changes raise new issues in the field of music retrieval. First, since music will not necessarily be produced as albums, the construction of *playlists* will be important in future systems. Playlists should ideally list songs of a similar genre that 'fit together'. These could be songs to play or songs to buy (and play). Techniques to quickly and automatically construct these lists are needed.

Second, the vast amount of music involved means that users may have difficulty navigating through a database to find new unknown music. Automatic playlist generation may help but alternative user interfaces which visualize the data may be needed to find acoustically remote but desirable songs.

Finally, with more and more music appearing on the Web, content providers are keener than ever to protect their intellectual property. Algorithms to quickly and reliably identify illegal copies of songs will be important.

We believe all these tasks could benefit greatly from a technique to automatically determine acoustic similarity between songs. For example, a playlist could be automatically chosen as list of songs acoustically similar to a favored query song. This could certainly be refined by further processing but the distance measure could give a useful first list. A visual user interface based on an acoustic distance measure may prove compelling. A scheme to detect acoustically similar but not identical copies of songs is attractive as it does not rely on accurate labeling or insertion of additional meta-data such as watermarks. We therefore focus our attention on developing a content-based music similarity measure.

The traditional and most reliable technique of determining music similarity is by hand. However, this is clearly infeasible for large quantities of music. Collaborative filtering techniques are an alternative to solo hand-classification (e.g. [1]). These techniques attempt to produce personal recommendations by computing the similarity between one person's preferences and those of (usually many) other people. However, these methods cannot quickly analyze new music. Also, it may be difficult to obtain reliable information from users.

Many researchers have studied the music similarity problem by analyzing MIDI music data, musical scores or using pitch-tracking to find a 'melody contour' for each piece of music. String matching techniques are then used to compare the transcriptions for each song (e.g. [2], [11], [6]). However, techniques based on MIDI data or scores are limited to music for which this data exists in electronic form. Also, only limited success has been achieved for pitch-tracking of polyphonic music [10] although recent results show much promise [7]. Thus reliably finding the melody in most commercial music is difficult or impossible using current technologies. Also, it seems likely that the type of music similarity required for playlist construction is based on the 'overall sound' of the music rather than simply the complexity of the main melody.

Other work has analyzed the music content directly. Blum et al. present an indexing system based on matching features such as pitch, loudness or Mel-frequency cepstral coefficients (MFCCs)[3]. Foote has designed a music indexing system based on histograms of MFCC features derived from a discriminatively trained vector quantizer [5].

In this paper, we build on the work of Foote to construct a distance measure between music based solely on the music content. We characterize songs using histograms of MFCC features but unlike Foote, the bins of our histograms are local to each song. This implies that the acoustic space for each song is efficiently 'covered' with adequate resolution where needed. Conversely if pre-determined bins are used, some songs may have all their information concentrated in one or two bins and important discriminating detail may be lost.

Our technique has many similarities to an audio retrieval technique described in [8], although we use K-means clustering rather than Gaussian mixture models to characterize each song. We also study the problem of music retrieval rather than the speech-in-audio retrieval problem studied there.

The organization of this paper is as follows. In Section 2 we describe our distance measure. We then describe how this can be incorporated into a playlist generation system. Next, we present results of experiments on a database of over 8000 songs. Finally we present our conclusions and suggestions for future work.

2 Spectral Novelty Distance Measure

Our distance measure captures information about the novelty of the audio spectrum. Conceptually, this corresponds to the type of instruments playing, including whether there is singing, which appears to be related to perceptual similarity. For each piece of music we compute a 'signature' based on spectral features. We then compare signatures using the Earth Movers Distance (EMD) [14]. These steps and the motivation behind them are described in more detail below.

2.1 Obtaining the Spectral Signature

Our spectral signatures attempt to capture the main statistics of a song's spectrum and hence characterize the main types of sounds present in the music. We achieve this by first dividing each song into short, locally stationary sections of 25ms called 'frames'.

For each frame, we then obtain a spectral representation. Many representations are possible so long as a distance measure is available to compare one frame to another such that frames which sound similar are close to each other. In our implementation we use MFCCs (e.g. [13]). These features are prevalent in speech recognition applications and are also useful for modeling music (e.g. [5], [3], [9]). They are based on the discrete cosine transform of the log amplitude Mel-frequency spectrum and can be compared using the Euclidean distance measure. Other spectral measures might include using the amplitude spectrum directly or a representation based on MP3 coefficients.

Given a the set of transformed frames for a song, we then cluster these frames into groups which are similar. The number of clusters may be fixed for every song, in which

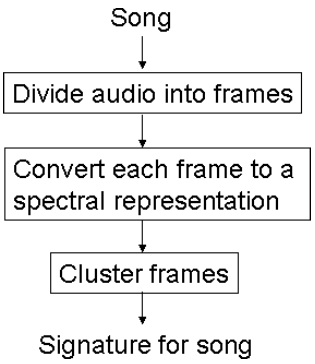

Figure 1: Top level diagram of the process of creating a signature for a song

case standard K-means clustering can be used [4]. Alternatively, the number of clusters chosen can be dependent on the song (e.g. [12]). The set of clusters is characterized by the mean, covariance and weight of each cluster (where the weight is proportional to the number of frames in that cluster). This set of clusters is denoted the ‘signature’ for the song. The whole process is shown in Figure 1.

It is important to note that the clustering is local to each song. Previous systems performed global clustering over an entire music database to obtain a set of representative clusters (sounds) (e.g. [5]). The disadvantage of this as shown by analogous work in image similarity [14] is that resolution is lost as the type and number of clusters is a function of the whole database. If many new songs have unheard sounds then they will be badly represented.

2.2 Comparing Songs

We obtain a spectral signature for every song of interest. These need only be calculated once and stored. We then compare the signatures for two different songs using the EMD[14]. It is crucial to use this technique as other methods cannot accommodate local clustering.

The EMD calculates the minimum amount of ‘work’ required to transform one signature into the other. A useful analogy is to consider the clusters for Song A and Song B as ‘piles of earth’ centered on the cluster means in N-dimensional space. We are interested in how much ‘earth’ or more correctly probability mass we need to ‘move’ to transform Song A’s clusters into Song B’s clusters.

More formally, let $P = \{(\mu_{p_1}, \Sigma_{p_1}, w_{p_1}), \ldots, (\mu_{p_m}, \Sigma_{p_m}, w_{p_m})\}$ be the first signature with m clusters where μ_{p_i} and Σ_{p_i} are the mean and covariance respectively of cluster p_i and w_{p_i} is the weight of that cluster. Similarly, let $Q = \{(\mu_{q_1}, \Sigma_{q_1}, w_{q_1}), \ldots, (\mu_{q_n}, \Sigma_{q_n}, w_{q_n})\}$ be the second signature. Let $d_{p_i q_j}$ be the distance between clusters p_i and q_j. In our work, we compute this using a symmetric form of the Kullback Leibler (KL) distance. For clusters p_i and q_j with means μ_{p_i} and μ_{q_j} and covariances Σ_{p_i} and

Σ_{q_j} respectively this takes the form

$$d_{p_iq_j} = \frac{\Sigma_{p_i}}{\Sigma_{q_j}} + \frac{\Sigma_{q_j}}{\Sigma_{p_i}} + (\mu_{p_i} - \mu_{q_j})^2 \cdot \left(\frac{1}{\Sigma_{p_i}} + \frac{1}{\Sigma_{q_j}} \right). \quad (1)$$

Let $f_{p_iq_j}$ be the 'flow' between p_i and q_j This flow reflects the cost of moving probability mass (analogous to 'piles of earth') from one cluster to the other. We solve for all $f_{p_iq_j}$ that minimize the overall cost W defined by

$$W = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{p_iq_j} f_{p_iq_j} \quad (2)$$

subject to a series of constraints. That is, we seek the cheapest way to transform signature P to signature Q. This problem can be formulated as a linear programming task for which efficient solutions exist. Having solved for all $f_{p_iq_j}$, the EMD is then calculated as

$$EMD(P,Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{p_iq_j} f_{p_iq_j}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{p_iq_j}}. \quad (3)$$

3 Evaluation

Our evaluation focuses on the utility of our similarity measure for playlist generation and music visualization. We also report some general statistics of our measure and examine its robustness to corruption of songs. Our results though preliminary are very promising.

3.1 Experimental Setup

We conduct experiments on an in-house database of over 8000 songs drawn from a wide range of styles. Each song in the database is labeled with the genre, song name, album name and artist name. The genre is one of the following: *Blues, Celtic, Classical, Comedy, Country, Folk, Jazz, Newage, Rap, Rock, Soundtrack, Techno, Various Artists, Vocal, World.* The genres are assigned according to the *All Music Guide* (AMG) database (www.allmusic.com).

For each song, we compute a signature based on K-means clustering of frames of MFCCs. We start with audio sampled at 16kHz and divide this signal into frames of 25.6ms overlapped by 10ms. We then convert each frame to 40 Mel-spectral coefficients and take the logarithm and the discrete cosine transform to obtain 40 cepstral coefficients. Of these, only the first 8-30 are used in the final system. We disregard the zeroth cepstral coefficient which contains magnitude information. Finally, we cluster the sequence of cepstral coefficients into 16 clusters using standard K-means clustering. This set of clusters is the signature for the song. After computing the signature for every song, we then compute the distances between all songs using the EMD as described above.

Nr. MFCC Features	Average distance between all songs	Average distance between songs on the same album
4	0.48	0.21
7	0.52	0.26
12	0.56	0.28
19	0.65	0.35
29	0.70	0.38

Table 1: Statistics of the distance measure

3.2 Simple Similarity Measure Statistics

We first examine some general statistics of our distance measure as a ‘sanity check’. Table 1 shows the average distance between songs for the entire database over a range of different MFCC parameterizations. We also show the average distance between songs on the same album. Our measure is such that the distance between a song and itself is zero (i.e. we have a dissimilarity measure). From Table 1 we see that our measure correctly assigns a smaller distance to songs on the same album which we expect on average to be perceptually more similar than other songs in the database.

3.3 Playlist Generation

In this section we consider the utility of our distance measure in the playlist construction problem. We form playlists as the N closest songs from our database to a given query song according to our similarity measure. Clearly we could devise better schemes to determine a playlist, such as combining the scores from several query songs and incorporating user feedback. This is the subject of ongoing work. The results in this section focus on the quality of playlists generated using our acoustic similarity measure alone.

Our experiments report the average number of relevant songs retrieved in the top 5, 10 and 20 songs. We therefore consider playlists up to 20 songs in length.

3.3.1 Objective Precision

Since user tests are expensive and time-consuming, we first use objective definitions of relevance to tune the parameters of our system and identify trends that are true on average over the whole database. We examine three objective definitions of relevance: songs of the same style, songs by the same artist and songs on the same album.

Table 2 shows the average number of songs returned by our system which have the same genre as the query song. As discussed, we report results for the closest 5, 10 and 20 songs. We see that the majority of songs returned *are* of the same genre as the query song. Note that this result gives only an indication of the performance of our system since several of our genre categories overlap (e.g. *jazz* and *blues*) and songs from both categories might still be perceived as relevant by a human user.

Nr. MFCC	Average number of songs in the same genre		
Features	Closest 5	Closest 10	Closest 20
4	3.02	5.81	11.2
7	3.24	6.17	11.7
12	3.43	6.53	12.4
19	3.44	6.57	12.5
29	3.36	6.44	12.3

Table 2: Average number of closest songs with the same genre as the seed song

Nr. MFCC	Average number of songs by the same artist		
Features	Closest 5	Closest 10	Closest 20
4	0.69	1.06	1.58
7	0.96	1.45	2.10
12	1.13	1.72	2.46
19	1.17	1.80	2.59
29	1.16	1.80	2.64

Table 3: Average number of closest songs by the same artist as the seed song

Tables 3 and 4 show similar results where relevance is defined as songs by the same artist and songs on the same album. From these tables we see that typically, around one song by the same artist or on the same album is one of the top 5 closest songs. Although it seems that these numbers are disappointing we have noticed in many informal tests that because we are using such a large database, our distance measure typically returns songs by different artists which are acoustically more similar than songs by the same artist or on the same album. All tables indicate that 19 MFCC features give the best results although this does not seem to be critical.

Nr. MFCC	Average number of songs on the same album		
Features	Closest 5	Closest 10	Closest 20
4	0.48	0.72	1.00
7	0.71	1.02	1.37
12	0.84	1.21	1.61
19	0.86	1.26	1.68
29	0.81	1.21	1.69

Table 4: Average number of closest songs on the same album as the seed song

Algorithm	Average Number of Similar Songs		
	Closest 5	Closest 10	Closest 20
Random	0.2	0.6	0.9
Proposed	2.5	4.7	8.2

Table 5: Average number of similar songs in playlists generated at random and by our similarity measure as judged by 2 users on 20 queries

3.3.2 Subjective Precision

Since it appears that 19 cepstral features give the best retrieval performance we conduct user tests with this configuration. Our tests compare a playlist generated by our system to a playlist generated at random from the same 8000 song database.

Two independent users participated in the test. They were presented with playlists for 20 randomly selected songs. For each song, a randomly generated playlist and the playlist generated by our system was presented. Users were instructed to rate each song in the playlist as 'similar' or 'not similar' to the query song. Interestingly, although no further instructions were given, both users naturally assumed audio similarity rather than say lyric similarity. There was good agreement between the users as to which songs were similar with only 12% of songs being rated differently.

The average number of similar songs for the first 5, 10 and 20 songs in the playlists is shown in Table 5. Despite the preliminary nature of our tests, the results are very encouraging and confirm what we have noted in many informal tests. On average, 2.5 out of the top 5 songs returned were similar for our system as opposed to 0.2 out of 5 for a random playlist generator.

3.4 Music Database Visualization

A content-based music distance measure presents new possibilities for designing a user interface to a music repository. If the data can be shown graphically, where each song is represented by a point, then the user may be able to navigate through the repository more easily using this graph than by traditional means such as searching for a song by name. This may be especially true for unfamiliar music.

In this section, we do not attempt to provide a solution to the user interface problem. Rather we present a 'proof of concept' that demonstrates that our distance measure can be used to create a data visualization that is in keeping with common sense.

To display our music database graphically, we transform each song to a real two-dimensional point using Multi-dimensional scaling (MDS). MDS (*e.g.* [15]) is a standard technique which transforms a series of objects, about which only relative distance information is available, to a series of K-dimensional points. The mapping attempts to preserve the relative distances between objects such that objects which are known to be 'close' according to the distance measure are mapped to points which are 'close' in K dimensional space.

To construct a visual representation of our database, we construct a matrix of song similarity according to our distance measure. We then perform MDS on this matrix to

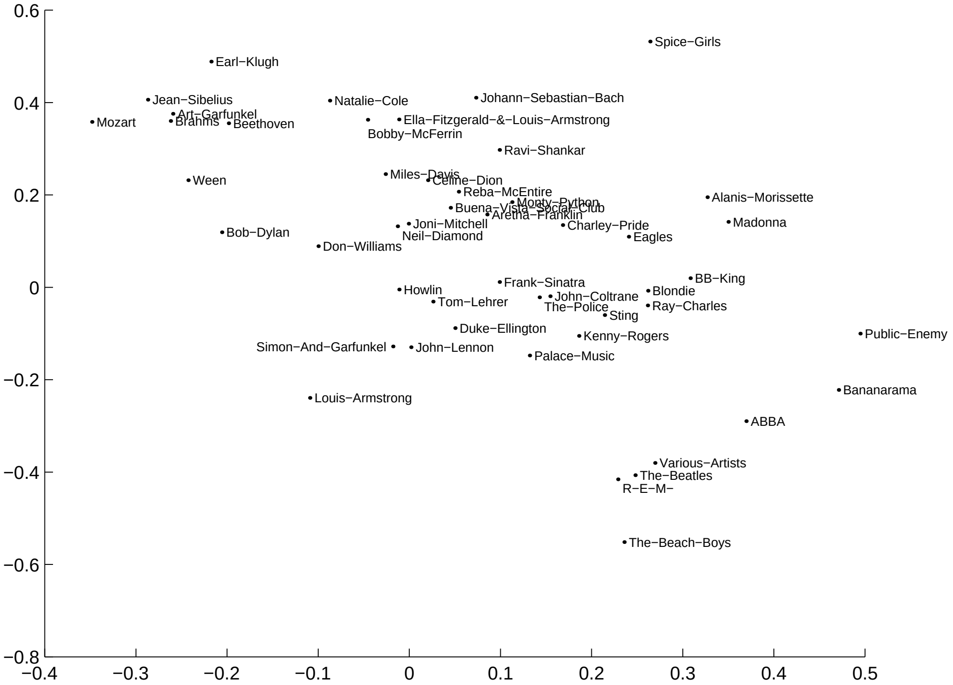

Figure 2: Visualization of a small number (around 50) of songs by well-know artists using MDS on the similarity matrix formed using our technique

obtain the coordinates in 2-D space for each song. We use the same parameterization as described in the previous section with 19 MFCC features.

Figure 2 shows the visualization of around 50 songs by well-known artists. We see that genres are fairly consistent and that in many cases, similar sounding artists are grouped together. Table 7 in Appendix A lists the songs visualized.

Figure 3 shows the visualization of 150 randomly chosen songs from the *Rock*, *Country* and *Classical* categories (50 songs from each category). Again, we see the songs roughly clustered into the genres. Despite the very preliminary nature of these results it seems probable that this technique could form the basis of an interesting user interface (perhaps similar to the web browsing interface at www.webmap.com). At the very least, these results provide visual confirmation of the utility of our music distance measure.

3.5 Robustness to Corruption

Finally, we investigate the robustness of our distance measure to ‘clipping’ of songs. Our interest is piqued by the potential utility of our measure for copyright enforcement and hence wishing to detect potentially slightly corrupted versions of a song.

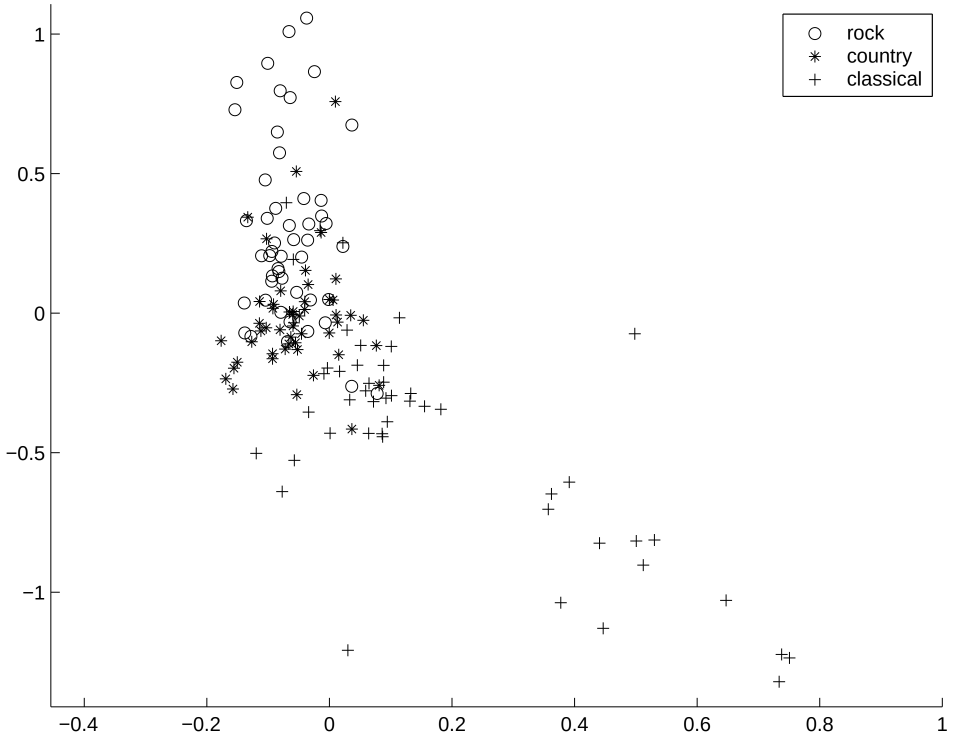

Figure 3: Visualization of 150 randomly chosen songs from the Rock, Country and Classical genres using MDS on the similarity matrix formed using our technique

Nr. MFCC	% of times original song returned within:		
Features	Closest 1	Closest 5	Closest 10
12	98.8	99.2	99.3
19	99.8	100.0	100.0
29	97.2	97.6	97.8

Table 6: Percentage of times the original song is returned as one of the closest 1, 5 and 10 songs when the query is a clipped version of the original

For all songs in our database, we remove a section of random length of up to 30s from a randomly selected place in the song. We then calculate the signatures for each song as before. For each corrupted song, we use our measure to find the closest songs to this in the clean database. Ideally, the original version of each corrupted song should be the first song returned. Table 6 shows the percentage of times the original song is returned as one of the 1, 5 and 10 closest songs when the corrupted version is used as the query. We see that these numbers are quite high indicating that our distance measure has some robustness to this type of corruption.

4 Conclusions and Future Work

We have described a method to compare songs based solely on their audio content. We have evaluated our distance measure on a database of over 8000 songs. Preliminary objective and subjective results show that our distance measure preserves many aspects of perceptual similarity. For 20 songs judged by two users, we saw that on average 2.5 out of the top 5 songs returned are perceptually similar. We also saw that our measure is robust to simple corruption of the audio signal and that it could be used to visualize the data in a meaningful way.

Ongoing work is focused in three main areas. First, we are still refining the parameters of our distance measure over all genres and investigating the effect of different clustering techniques to obtain the song signatures. Second, we are exploring the many heuristics that can be used to select the best playlist given a query song or songs. Finally, we are investigating the augmentation of our similarity measure by other audio and non-audio information.

5 Acknowledgments

We would like to thank Scott Blackwell and Jean-Manuel Van Thong for their help writing the interface for user tests. Thanks are also due to T. M. Murali for bringing the EMD to our attention and Tamara Munzner for advice on visualization techniques. We also thank the members of the CRL speech group for many useful discussions.

References

[1] Workshop on collaborative filtering. Proceedings, University of California at Berkeley, March 1996.

[2] S. Blackburn and D. De Roure. A tool for content based navigation of music. In *ACM Multimedia*, 1998.

[3] T. L. Blum, D. F. Keislar, J. A. Wheaton, and E. H. Wold. *Method and article of manufacture for content-based analysis, storage, retrieval, and segmentation of audio information*. U.S. Patent 5, 918, 223, 1999.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2000.

[5] J. T. Foote. Content-based retrieval of music and audio. In *SPIE*, pages 138–147, 1997.

[6] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming. In *ACM Multimedia*, 1995.

[7] M. Goto. A predominant-F0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[8] Z. Liu and Q. Huang. Content-based indexing and retrieval by example in audio. In *ICME 2000*, 2000.

[9] B. Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*, 2000.

[10] K. D. Martin. Transcription of simple polyphonic music: Robust front end processing. In *the Third Joint Meeting of the Acoustical Societies of America and Japan*, 1996.

[11] R. McNab, L. Smith, I. Witten, C. Henderson, and S. Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Digital Libraries 1996*, pages 11–18, 1996.

[12] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML 2000*, 2000.

[13] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice–Hall, 1993.

[14] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a metric for image retrieval. Technical report, Stanford University, 1998.

[15] F. W. Yound and R. M. Hamer. *Multidimensional Scaling: History, Theory and Applications*. Erlbaum, 1987.

A Well-known Songs

Genre	Artist	Song
Jazz	Bobby McFerrin	Don-t Worry- Be Happy
Jazz	Louis Armstrong	Hello Dolly
Rock	Alanis Morissette	You Oughta Know
Rock	Bob Dylan	Blowin- in the Wind
Rock	John Lennon	Oh Yoko-
Rock	Ween	I-m Holding You
Comedy	Tom Lehrer	The Vatican Rag
Comedy	Monty Python	Lumberjack Song
Classical	Wolfgang Amadeus Mozart	Requiem
Classical	Jean Sibelius	Pelleas et Melisande- Melisande
Techno	Various Artists	Believe
Jazz	Duke Ellington	Midriff
Jazz	John Coltrane	Seraphic Light
Country	Palace Music	Ohio River Boat Song
Vocal	Frank Sinatra	I-ve Got You Under My Skin
Blues	Howlin Wolf	Red Rooster
Rock	R-E-M-	Shiny Happy People
Rock	The Beatles	All My Loving
Rock	Aretha Franklin	Think
Rock	Radiohead	Creep
Rock	Sting	If You Love Somebody Set Them Free
Rock	The Beach Boys	Help Me- Rhonda
Rock	Bananarama	Venus
Rock	Madonna	Like a Virgin
Rock	Spice Girls	Wannabe
Rock	The Police	Message in a Bottle
Rock	Blondie	Heart Of Glass
Rock	Eagles	Hotel California
Country	Charley Pride	After me, after you
Country	Don Williams	Fly Away
Country	Reba McEntire	Between a Woman and a Man
Rap	Public Enemy	B Side Wins Again
Blues	BB King	Sweet Little Angel
Blues	Celine Dion	All By Myself
Classical	Beethoven	Allegretto
Classical	Brahms	Piano Concerto No 2 in B flat, Op 83
Classical	Johann Sebastian Bach	Allegro
Rock	ABBA	Dancing Queen
Jazz	Miles Davis	Blues for Pablo
Jazz	Earl Klugh	Winter Rain
Jazz	Ella Fitzgerald & Louis Armstrong	Cheek to Cheek
Jazz	Natalie Cole	As Time Goes By
Country	Kenny Rogers	The Gambler
Blues	Ray Charles	Hit The Road Jack
Rock	Art Garfunkel	Bright eyes
Rock	Neil Diamond	September Morn
World	Ravi Shankar, Ali Akbar Khan	Raga Palas Kafi
World	Buena Vista Social Club	Candela
Folk	Joni Mitchell	Car On A Hill
Folk	Simon And Garfunkel	Bridge Over Troubled Water

Table 7: Songs with genre and artist which are visualized in Figure 2

CRL 2001/02 **A Content-Based Music Similarity Function** Beth Logan Ariel Salomon

June 2001