Unsupervised Classification of Sound for Multimedia Indexing*

Bruce Matichuk
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
matichuk@cs.ualberta.ca

Osmar R. Zaïane
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
zaiane@cs.ualberta.ca

ABSTRACT

Segmenting audio streams in a significant manner and clustering sound segments objectively, is a significant challenge due to the nature of audio data. This paper presents some preliminary work on clustering sound segments based on frequency and harmonic characteristics. New metrics for comparing the similarity of sound segments are also devised.

Keywords

Multimedia Data Mining, Sound Processing, Classification, Clustering, Similarity comparison

1. INTRODUCTION

Multimedia systems play an increasingly important role in our daily lives. Access to media through the Internet, and by a growing number of electronic media recording and playback devices, is influencing and changing our lives in a profound way. Navigating through the vast amounts of multimedia data being generated is becoming an overwhelming problem. Although automated techniques for accessing electronic media are being developed, the science is still in an embryonic stage. In the area of audio-data, there has been promising early research focusing on supervised learning techniques. The authors argue that unsupervised techniques are required to handle real world retrieval situations. The research presented investigates a clustering technique that provides an automated classification scheme for short sound samples. Recognition of short samples can be combined with well-known pattern recognition algorithms to provide a viable sound retrieval system. The research could be applied to synthetising speech and sound, automatic text to speech conversion and sound compression.

General sound recognition is a difficult problem requiring the division of samples into small recognizable sound segments that can be combined to recognize complex sound patterns. Sounds are composed of signals at multiple frequencies that can be graphed on a Time-Frequency Distribution graph (TFD). Real world objects vibrate with characteristic vibrations and thus produce sound waves with characteristic harmonics that allow people and systems to perform sound recognition. A harmonic is a component frequency of a complex wave that is an integral multiple of the fundamental frequency. A sound segment is analyzed by observing the frequency amplitude pattern that occurs in the segment. Standard pattern recognition techniques can be used to recognize a frequency amplitude pattern including, neural nets, belief nets, decision trees, distance metrics, etc.

The difficulty in analyzing compound sounds lies in the problem of dividing a compound sound into recognizable segments and classifying these segments. Consider the speech recognition problem. Speech can be viewed as a sequence of sound segments that can be recognized individually and therefore combined into compound segments representing phonemes and words. Other real-world sounds can be treated in a similar way. For example, consider the sounds of a dog barking or a child crying. Both represent sounds that can be described by collections of individually recognizable sound segments.

The first task of classification is to decide upon a classification scheme. With sounds, this is difficult because of the large number of variations possible for any given sound segment. Our hypothesis is that clustering techniques can be used to classify sound segments in an unsupervised way. This hypothesis is based on the observation that sound samples are composed of recurring sound segments and that the characteristics of these segments can be determined by clustering segments that seem similar. The segments within each cluster can be analyzed to determine each cluster's representative feature set.

The remainder of the paper is organized as follows. In Section 2, we introduce some related work to audio analysis. In Section 3, We present the methodology adopted for clustering sound segments. A similarity metric for clustering sound is presented in Section 4. Section 5 describes some preliminary experiments on frequency-based and harmonic-based clustering. Finally, we conclude our study and give some pointers to future work in Section 6.

*Work in Progress.
Research is supported in part by the Natural Sciences and Engineering Research Council of Canada.

© The copyright of this paper belongs to the paper's authors. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage.
Proceedings of the International Workshop on Multimedia Data Mining (MDM/KDD'2000), in conjunction with ACM SIGKDD conference. Boston, USA, August 20, 2000
(O.R. Zaïane, S.J. Simoff, eds.)

2. RELATED WORK

Audio retrieval techniques are generally classified as content-based or browser-based [13]. Content-based retrieval allows the search through audio for a segment using some predefined criteria. Browsing allows a user to scan through audio based on some navigational parameters. Both methods require some kind of labeling of audio data that can be used in the search process. Audio analysis to support searching can vary with respect to the semantic nature of the data. Search tasks can be semantically simple such as "Find the next location of 1 second of silence." Search tasks can also be semantically difficult, such as "Find all occurrences of the word 'the'." Some techniques being used to index sounds use a generated feature vector as an index [12]. Some systems can analyze temporal patterns [6]. The main difficulty of currently researched techniques stems from the inability to detect "in-context" high-level semantics of sound. Recent research that uses cognitive models[13] appears promising but is limited by the small number of classifications used to segment sounds. No attempt is made to discover high-level structure in sound in an automated way. Research in [10] explores a broader range of sound classes but also does not attempt to deal with complex structure. Instead, heuristics are used to detect features within sound that can be used to classify sounds in some pre-determined way.

Most research in advanced audio processing has been performed in the context of speech recognition. However, the techniques used are finely tuned toward extracting human speech patterns and detecting pre-determined natural language structures. We are examining a technique that is able to automatically determine the features and the structure of sounds suitable for general recognition and audio retrieval.

3. METHODOLOGY

The real world is composed of objects that generate sounds that vary in pitch and volume. Pitch refers to the fundamental frequency of a wave. Volume refers to the overall energy of a wave. In some cases, pitch and volume convey important information that must be incorporated into the recognition process. Recognizing a song, for example, requires attention to pitch and volume. However, for sounds like parts of speech, recognition requires a mechanism that analyzes sounds in terms of harmonic components. The process of harmonic feature extraction involves several steps. First, convert the signal from the time domain to the frequency domain using the Fast Fourier Transform (FFT) [2]. This is a standard technique used in all sound recognition algorithms. Second, extract a small sample of points that represent local peaks. This allows the selection of features that are important to the recognition of a segment. Third, normalize the sample by choosing the largest sample value and setting it to a value of 1 and scale the other sample values relative to the maximum. This step provides volume independence for the sample. The resulting set of values represents a harmonic feature set that can be used to identify the original sample independent of the volume or pitch.

The kinds of clusters that are found in a signal will depend on the similarity technique used to compare segments. Similarity clustering that is based on harmonics will cluster parts of speech in a way that does not depend on pitch. The reason for this is that the relative strength of the frequencies in a signal are relative to the harmonic characteristics of a vibrating object and tend to be independent of pitch. Consider a person making the sound 'oh' with a low pitch versus a high pitch. The harmonic characteristics in both cases will be virtually identical although the frequency composition will be completely different. Similarity clustering using frequencies will cluster sounds that are similar in pitch regardless of the harmonics. Consider the sound 'oh' versus 'ee' at the same pitch. Both of these sounds will have closely matched frequencies at a given pitch and will thus be indistinguishable based on a frequency by frequency comparison.

A further consideration in sound analysis is the length of time of a sample segment. Shorter segments are more useful in describing complex sounds such as speech. FFTs over shorter segments, however, are less accurate. There is an ideal length for a variety of sound types and optimal length may even vary within a sound sample. Tests regarding timing are not reported in this paper but the toolkit that was developed allows for timing variations. All of our tests were performed using 1/10th-second time segments. The testing software we developed is able to handle arbitrary time segments, however, 1/10th-second intervals seemed to produce the best results. A future paper will report results on varying time segments.

3.1 Clustering Sound Samples

To investigate sound sample clustering, a test system was developed whereby a tester could easily record sounds and apply a clustering algorithm to the sample. Two techniques for comparing sound samples were investigated. One technique was devised to compare two samples on a frequency by frequency basis. Another technique was devised which allows the comparison of harmonic characteristics. A series of tests were performed to determine the effectiveness of the comparison algorithms and their capability for correct clustering. The test bed devised uses a graphical technique for examining the samples and the cluster constituent. A variant of the ROCK clustering algorithm[5] was chosen to cluster sound segments.

Although further investigation using other clustering techniques is required, the ROCK algorithm has some essential features that make it an attractive technique for clustering sound. ROCK does not merely classify items based on similarity alone. Rather, ROCK considers the number of neighbours that appear to be similar to an item a more important clustering metric than the degree of item similarity. Initially all items under consideration by the algorithm are treated as belonging to separate clusters. Based on a similarity "threshold" value, each "cluster" is assigned a list of neighbour clusters that seem similar. The two clusters that seem to share the most neighbours are combined into a single cluster. A repeated evaluation of all clusters is made, combining clusters that share the most neighbours, until a desired number of clusters is achieved or until all discovered clusters have no neighbours. Classification is made of new elements by counting the number of elements within a cluster that are similar to the new element and choosing the cluster with the largest neighbour count scaled relative to overall size of each cluster.

3.2 The Threshold Phase Transition Problem

Unfortunately, the ROCK algorithm is very sensitive to the value of the threshold. Above a particular value, clustering is poor and recognition is not very effective; most points are classified as disconnected. Below a certain threshold, clustering is also poor and recognition is not effective since most points are classified as neighbours. To alleviate these problems, reduced sensitivity to thresholds is required both in clustering and in later classification.

3.2.1 Trimming

To reduce the sensitivity of ROCK to the threshold value, "trimming" reduces the number of allowed neighbours. This allows for a varying value in the threshold while maintaining a reasonable number of neighbours to cluster. A neighbouring pair of clusters is any two clusters that share sufficient neighbour links that allow the classification of both clusters into a single cluster. The neighbour vector for any cluster is ordered according to link counts. The trimming algorithm works by eliminating neighbours from the most linked cluster until pre-determined maximum number of neighbours is achieved. It is important when doing this trimming that neighbours are trimmed in pairs. Without this reciprocal trimming, the rest of the ROCK algorithm will get "out-of-sync" and the algorithm performance will deteriorate. The ROCK algorithm requires all neighbour activity to work in pairs: one action for the link "to" node and a corresponding action for the link "from" node. Therefore, when trimming, for each neighbour that is eliminated from a cluster's neighbour list, the reciprocal entry in the corresponding cluster's neighbour list must also be eliminated. An assumption that was made in choosing the trimming value was to assume that each cluster was relatively similar in size; no cluster was significantly larger than all others. This implies that a good cluster trimming value is approximately the average cluster size. In our case we chose a trimming value calculated using the sample size divided by the desired cluster count.

3.2.2 Threshold Searching

Once the clustering algorithm has completed, the final phase of operation is the assignment of items to clusters. Certain threshold values will cause many items to be unclassifiable. In some cases, this might be acceptable, however, in cases where a classification is required, an alteration to the suggested classification technique is necessary. Consider, for example, the sounds 'aw', 'oh', 'ee', 'ay' and 'eh'. Suppose we are looking for five clusters. Below a certain similarity threshold, ROCK would likely classify new sounds, which were not in the original clustering, as belonging to none of these categories. Above a certain threshold, new sounds that are similar will all be placed in the first category.

The solution is to search for a useful threshold during each assignment. The threshold value must start low. Each cluster is checked for neighbour counts using a low threshold. If an insufficient assignment is made, the threshold value is increased until a similarity count for one of the clusters is above a certain value. To employ threshold searching our modification to the ROCK classification algorithm wraps the original ROCK classification routine with a threshold searching loop. Through each iteration, the threshold is multiplied by 0.9. Our algorithm searched for at least one neighbor link between the sample and one of the cluster samples. Note that the implication here is that the "searching" is not done manually. In fact the "searching" is done automatically within the classification loop.

4. SIMILARITY METRICS FOR CLUSTERING

4.1 Sound Sample Comparison

Sound sample analysis begins with the transformation of a sound segment into the frequency domain using a Fourier Transform. The result of the transform is a series of complex numbers representing the frequency components of the sample. Using an intensity function, each frequency component can be converted into a real number. This set of frequency intensity values for a sound segment can be analyzed, and can be transformed further into a set of characteristics representing the sound. In order to use the ROCK algorithm, a similarity metric was devised which distinguishes between sound samples. The similarity metric used in ROCK clustering must produce a value from 0 to 1: 0 indicating no match, whereas 1 indicates an exact match. The euclidean distance is often used as a similarity metric in clustering. However, another metric that produces similar results is a calculation S that sums up relative ratios. In this case, given two signals composed of frequency vectors X and Y, S can be calculated as follows:

$$S = \frac{\sum_i \frac{min(x_i, y_i)}{max(x_i, y_i)}}{d} \quad (1)$$

where d represents the dimensionality of the data. The value d varies and is calculated using a counter. Pairs, which had values below a certain threshold, were discarded from the calculation. This was necessary since large numbers of low values do not contribute to the analysis.

4.2 Harmonic Filter Based Similarity Metric

To compare harmonics, a metric is desired that is independent from pitch or volume. To provide this comparison, a harmonic component vector is calculated which is comprised of the harmonics of a sample. Natural sounds other than noise consist of a number of harmonic frequencies above a base frequency. By comparing the intensities of the harmonics, one sound can be distinguished from another. Similar to the Frequency Based Metric, the Harmonic Based Metric uses the same metric as S in (1) where d represents the dimensionality of the data. The value d is calculated by adding up all pairs of harmonics where both x and y are above a given cutoff. Each x and y value represents a harmonic pair. The harmonics are ordered such that the first harmonic of sound x is compared with the first harmonic of sound y, etc. Harmonics are found by scanning the FFT of a sample and looking for peaks.

4.3 Calculating Harmonics

To calculate the harmonic intensities the following algorithm was deployed using the FFT sample:

1. Look for a value that is a maximum.

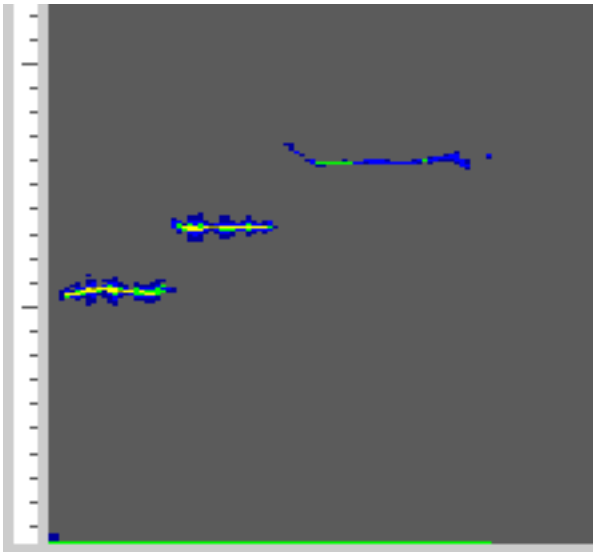

Figure 1: FFT of the recording used in a frequency clustering test.

2. For each successive value determine if the values are declining.

3. If values decline for two simultaneous measurements, record the last maximum and reset it to zero.

5. PRELIMINARY EXPERIMENTS

The test system was built using Microsoft Visual C++ 6.0 and Developer Studio. The recording code, Fourier transform code and the code that draws the frequency histograms was originally developed by a company called Relisoft Inc. based in Seattle. The code was modified to add: the ROCK algorithm, buttons to control sampling and clustering, and extensions to the draw code to allow for the drawing of cluster contents and frequency histograms with harmonic indicators.

The following tests were performed using frequency clustering and harmonic clustering. The first test involved whistling three distinct notes and searching for frequency clusters. The second involved recording the long vowel sounds "A" "E" and "U" and performing harmonic clustering.

In Figure 1 the FFT of the recording used in a frequency clustering test is displayed. The vertical axis indicates frequency. The horizontal axis is time. The tests performed called for 5 clusters from 50 samples. Using a theta value of 0.5, a cutoff of 0.4 and trimming each initial cluster to 10 neighbours, the results obtained are displayed in Figure 2. Figure 3 displays the results of the segment assigment for the entire original frequency test sample using the classification scheme derived from the frequency clustering results.

In Figure 4 the FFT of the recording used in a harmonic component clustering test is displayed. The tests performed also called for 5 clusters from 50 samples. Using a theta value of 0.25, a cutoff of 0.1 and without trimming, the results obtained are displayed in Figure 5. The divisions between phonemes can be visually distinguished by the varia-

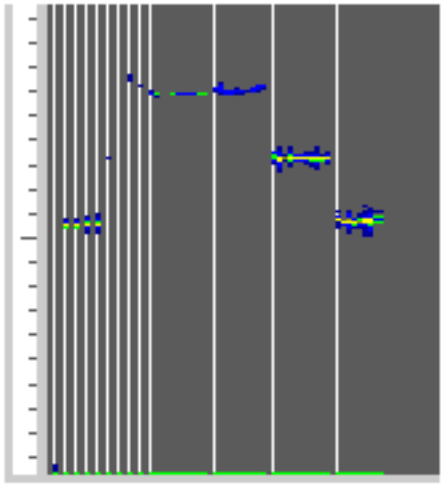

Figure 2: Clusters of frequency-based segment samples from original audio source.

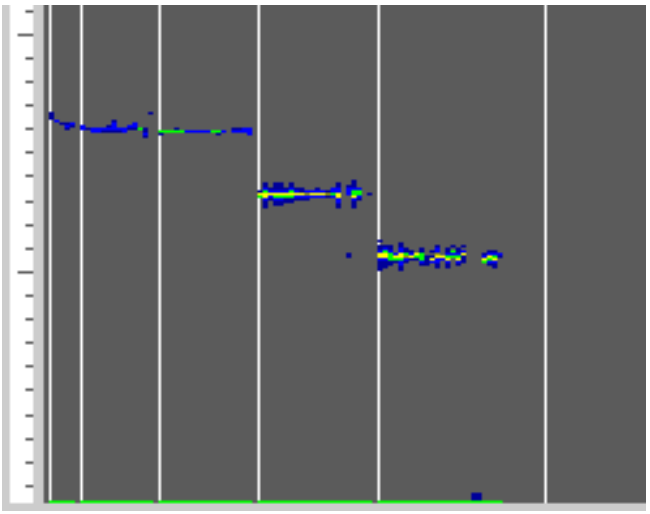

Figure 3: Frequency-based cluster assignments.

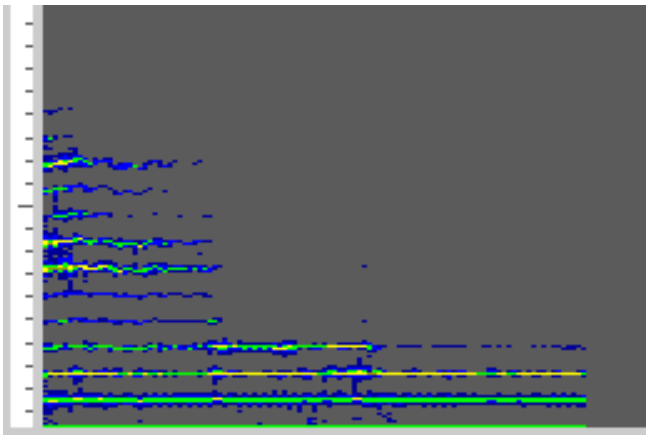

Figure 4: FFT of the recording used in a harmonic clustering test.

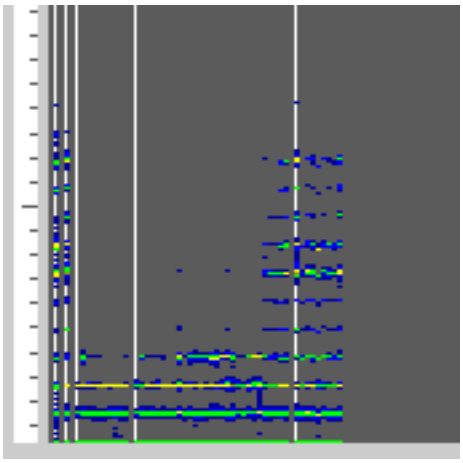

Figure 5: Clusters of harmonic-based segment samples from original audio source.

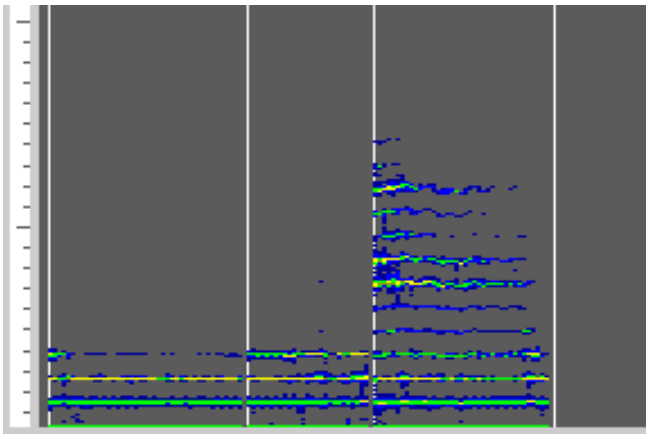

Figure 6: Harmonic-based cluster assignments.

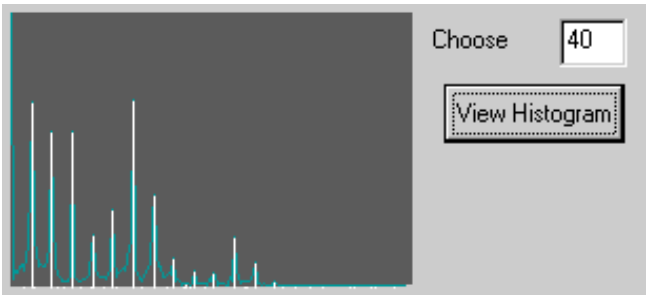

Figure 7: Example of histogram graph.

Figure 8: Histograms for the sounds "Aaa", "Ooo", "Eee".

tion in harmonic components. Figure 6 displays the results of the segment assignment for the entire original harmonic test sample using the classification scheme derived from the harmonic clustering results.

The test illustrated above was clearly very successful in that the clustering technique was able to precisely delineate the 3 vowel sounds.

Figure 7 is an example of the histogram graph. The white lines are used to indicate the locations of peaks. These peak values are used in the harmonic similarity equation.

The graphics in Figure 8 show the frequency histograms for the phrases "Aaa", "Ooo" and "Eee". The white spikes indicate peaks. The vertical axis indicates power. The horizontal axis is frequency.

Although a large number of tests were done using speech, the results reported here represent a small sample of those tests. A later paper will report on these tests and other tests using various kinds of sounds. The similarity technique that we are currently experimenting with does not deal very well with noise. Further work is required to allow noises to be included in the similarity metrics.

6. CONCLUSIONS AND FUTURE WORK

The ROCK algorithm is a very processing intensive technique for clustering. However, the algorithm is very intuitive and is general enough to apply to many different kinds of clustering problems. One major problem with ROCK is its sensitivity to the threshold value. The correct value to use is determined by the nature of the data and the nature of the similarity metric. To alleviate this problem we introduced the notion of "trimming" and the notion of threshold "searching". These techniques greatly reduced the sensitivity of the algorithm to the threshold value. Further research is required to determine if these techniques can be applied to broader problem sets. We were also able to determine

that sound clustering is a viable technique for unsupervised classification of sounds.

Further work will focus on comparing ROCK to other clustering algorithms in the contexct of sound[14, 9, 3, 4, 7, 8, 11]. Other similarity metrics should be investigated including neural nets, decision trees and other pattern recognition algorithms. Additional similarity metrics are required to classify noise and inter-segment transitions [1]. Finally, clustering techniques that combine segments into complex groupings are required to classify higher order sounds. This would require multiple clustering passes that cluster groups corresponding to sound patterns such as rhythms, melodies or speech.

Further research will also investigate classification techniques that will allow segments to belong to multiple clusters. This can be achieved by applying a "fuzzy" similarity threshold value during classification in contrast to the current classification technique which is binary.

7. REFERENCES

[1] A. Czyzewski. Mining knowledge in noisy audio data. In *Proc. Second Int. Conf. on Knowledge Discovery and Data Mining (KDD96)*, pages 220–225, Portland, Oregon, August 1996.

[2] D. E. Dudgeon and R. M. Mersereau. *Multidimentional Digital Signal Processing.* Printice-Hall, 1984.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In *Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, Portland, Oregon, August 1996.

[4] V. Ganti, J. E. Gehrke, and R. Ramakrishnan. CACTUS - clustering categorical data using summaries. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, San Diego, CA, 1999.

[5] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *Proc. 1999 Int. Conf. Data Engineering (ICDE'99)*, pages 512–521, Sydney, Australia, March 1999.

[6] D. Hindus, C. Schmandt, and C. Horner. Capturing, structuring and representing ubiquitous audio. *ACM Transactions on Information Systems*, 11(4):376–400, Oct. 1993.

[7] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)*, pages 58–65, New York, NY, August 1998.

[8] Z. Huang. Extensions to the *k*-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304, 1998.

[9] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A survey. *ACM Comput. Surv.*, 31:264–323, 1999.

[10] K. Melih and R. Gonzalez. Audio retrieval using perceptually based structures. In *Proc. IEEE Intl. Conf. on Multimedia Computing and Systems*, pages 338–347, 1998.

[11] G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pages 428–439, New York, NY, August 1998.

[12] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search and retrieval of audio. In *Proc. 1999 IEEE Multimedia Conf.*, pages 27–36, 1996.

[13] T. Zhang and C.-C. J. Kuo. Heuristic approach for generic audio data segmentation and annotation. In *Proc. 1999 ACM-Multimedia Conf.*, pages 67–76, Orlando, FL, October 1999.

[14] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96)*, pages 103–114, Montreal, Canada, June 1996.