Group Info:
Brian Potetz (bpotetz@cs.cmu.edu) and
Sofia Cavaco (scavaco@cs.cmu.edu).


Project Description:
We propose to create an indexing structure to store sounds and optimize cluster related queries. Modern multimedia databases typically offer the ability to query sound files based on proximity to a given sound file. The measure of distance used is often euclidean distance within some small feature space of the sound files, or may be a more complicated notion of distance based on more computationally expensive comparisons between sound files. However, the notion of distance used is almost always predefined, and independent of the data. Yet knowing how a given data point relates to other data points in the database can often yield insight into what data points should be considered similar. For example, a data point may belong to the tip of an elongated cluster of datapoints. In many circumstances, it is advantageous to report points in the same cluster as being closer than points in other clusters, even if some points in other clusters are closer in euclidean space. Considering a music database, the user can make title related queries or singer related queries but he or she may also need to look for music entries of a certain (unknown) category.

Computationally expensive data mining operations allow users to classify data into clusters. However, these operations require several passes through data, and classifications can be made obsolete when data is inserted, deleted, or modified. We propose to develop an indexing scheme that will facilitate cluster based queries.

While sound files are usually one-dimensional data, we will transform sounds into a multi-dimensional domain and we will then use this new data structure to index sounds transformed in this way.

Apart from the usual insert, delete and search, the data structure will also be equipped with a set of primitives to answer cluster related queries (like requests to return all sounds in a cluster), as well as queries related to the properties of sounds. These would include requests to return:

- the fundamental frequency of a sample;
- the average loudness of a sample;
- samples with a given fundamental frequency, etc..

If time allows, we may extend our project to enhance the speed of queries based on geodesic distance between data points. The idea behind geodesic distance is that data is often highly concentrated along low dimensional manifolds within a high dimensional state space. Our database of audio files is a good example of this. Each audio file contains a recording of a bar being struck with a hammer. Bars of different materials tend to cluster along elongated spiral arms in principal component space. In some places these arms run close to one another. Geodesic distance will yield a large distance for two data points on different clusters, even if they are close in euclidean space. Because recordings of bars of different materials may be added and removed from the database, we cannot rely on a static definition of distance between two data points - our notion of distance depends on the data. Geodesic distance can be approximated by the distance between two data points in a graph where nodes are connected if their corresponding data points are within a certain small distance of one another. Queries based on geodesic distance can be very time consuming if no special indexes are provided.

An index to facilitate queries that rely on clustering of data must optimize two criterion: It must be fast, and it must be accurate. Accuracy can be gauged by querying on data whose true categories are known. Accuracy under a variety of conditions must be measured, since index performance may depend on the number of insertions/deletions since clusters were reorganized. Efficiency must also be gauged under different conditions. Insertion, deletion, and reorganization operations must be measured, as well as a variety of query conditions.

Logistics:
Plan of Attack and Schedule:

The background reading and design phases of our project can be shared equally by both group members. The implementation phase can be broken into segments that can be solved independently: the index data structures and algorithms must be written, a new form of query must be introduced and added to the SQL interface, a new cluster-based relation must be defined, the optimizer must be made aware of the costs of performing cluster based queries, and a method of evaluation must be constructed to test the performance of the algorithm.

Week of 3/7
Continue reading background literature

Week of 3/14
Begin design of index structure for cluster information
Continue reading background literature

Week of 3/21
Continue design of index structure.
Plan implementation of index structure.

Week of 3/28
Begin implementation of index structure.
Build data structures and algorithms for the index.
Complete milestone report.

4/2 - Project Milestones due

Week of 4/4
Continue implementation of index structure.
Build new relation and SQL interface for cluster based queries.
Modify optimizer to estimate the relative speed of cluster based queries.

Week of 4/11
Complete implementation of index structure.
Evaluate performance of cluster index.
Begin working on project write-up.

Week of 4/18
Continue working on project write-up.

Week of 4/25
Complete project write-up and prepare for presentation.

4/30 - Project demos and presentations

Milestone:

By the milestone date we will have completed the design phase and we will have begun building the data structures and algorithms for our index. Our milestone report will include a brief description of the problem, a description of the design of our index (its data structures and algorithms), and some notes on how we intend to implement the index in PostgreSQL.

Literature Search:
Papers related to indices for audio files:


Papers related to indices for non-euclidean queries:


Papers related to geodesic distance:


Resources Needed:

We have all resources needed to complete this project. We will implement it on PostgreSQL.
Getting Started:

As shown in the weekly plan we have started the background literature reading. We will soon start the design phase.