ABSTRACT

Database indices provide faster access to data. Multimedia type data is inherently large, so indexing is a critical aspect of any such database. Feature based multimedia indices are created as the data is entered into the database using an algorithmic procedure. Semantic based indices are created using user-defined descriptions of data. This paper will study theoretical and existing multimedia databases with feature and semantic based indices and show that semantic based indices provide for faster data access than their feature based counterparts. A sample of four of each type, feature and semantic, of indices will be examined to produce an average speed of data retrieval. Big-O notation will be used to measure the speed of retrieval. Due to the lack of commercial multimedia database systems, only four feature and four semantic based indices will be examined for comparison. For the sake of consistency, only databases that hold visual multimedia type data will be studied.

Keywords
Multimedia, Database, Index, Semantic, Feature, Retrieval, Speed.

1. INTRODUCTION

Indices are a valuable part of any database. Multimedia databases most likely contain a substantial amount of data. Serkan Kiranyaz et. al. describe the situation, “During the recent decades, technological hardware and network improvements have caused a rapid increase in the size of digital visual information. Besides the benefits and usages, such massive digital visual information has brought about the problems of storage and management.” It is therefore imperative that indices are provided to the user to locate the information that is queried. Multimedia type data is obviously quite different than traditional type of data that is stored in databases such as integers and strings. The complexity of the data and the types of queries necessary make traditional indices not useful in databases that hold multimedia data. Qian Huang explains, “When the amount of data is small, a user can retrieve desired content in a linear fashion by simply browsing the data sequentially. With the large amounts of data now available, and expected to grow massively in the future, such linear searching is no longer feasible.”

Multimedia type data has physical characteristics like number of pixels, color and size. The data also has semantic characteristics, which describes what is happening in the image or video. Feature based indexing of MM databases is usually based on physical characteristics, also called content-based indexing. Semantic indices are based on the semantic characteristics, called semantic-based indexing. Semantics involves what is actually happening in the image/video. It may be worthwhile for a database to provide both types of indices. It would certainly be helpful to understand which provides faster access to the data. Additional studies as to which type of index provides more accurate retrieval would also be useful. A combination of all these factors will help provide the best indices for commercial multimedia databases that have been and will be developed. For this study, four feature and four semantic type indices will be examined to help determine which type of index provides faster retrieval of multimedia data. Speed of access to data will be examined using Big-O notation. For example, an algorithm with complexity of O(n) would take t units of time to compute n units. The next section of this paper will explain the reason this type study is useful. The following two sections will include a short explanation of the indices studied and determine the complexity of each index, which will indicate how fast the data can be retrieved. The final three sections will describe the results of the study, what conclusions can be drawn and summarize the entire study.

2. PURPOSE

Commercial relational databases offer very little support for multimedia type data due to the complexity of the data. With very few commercial multimedia databases in industry today and the existing ones very specialized, no standard indexing technique exists for multimedia data. The purpose of analyzing the different ways of indexing multimedia data is to determine which method is more time efficient in retrieving data, so that multimedia databases that have been or will be developed in the future can incorporate the optimal index. Speed of retrieval is not the only factor in determining which index is optimal, but it is a very important factor. The answer may not lie in a purely feature or purely semantic based index but some combination of both, but this study should help provide an answer and some of the underlying factors. The results of this and complementary studies can be used to produce optimal multimedia indices that provide fast and efficient data retrieval.
3. FEATURE BASED INDICES

Feature based indexing is performed as data definition and manipulation is performed in the database. Z.J. Zheng defines a feature based index as “…an algorithmic procedure, which establishes a set of invariants that directly correspond to certain aspects of content-based information for shapes and forms.” This definition is accurate but not as comprehensive enough. As will be described, other feature based indexes have been proposed that use both physical and semantic information to produce an index. The purpose of the feature indices is to use characteristics of the complex data and use an algorithm to produce a more traditional type of data such as a single or group of integers. The traditional data type can then be used to store and retrieve the data. John Smith asserts, “Given that multimedia indexing systems are improving capabilities for automatically detecting concepts in multimedia documents, new techniques are needed that leverage these classifiers to extract more meaningful semantic descriptors for searching, classifying and clustering.”

3.1 Ardizzo Index

Edoardo Ardizzo et al. provide their reasoning why feature-based indices are the most effective indexing method for multimedia databases, “…describing semantics of an image is a task too hard to be performed in a fully automatic way with current technologies, and a human action is needed. However, manual content description is usually subjective and time-expensive, thus only the minimum comments necessary to the specific task are usually entered.” The index proposed by Ardizzo involves index pre-segmented video clips. Frames from the video clip are used to produce multiple indices. Three types of features are used to index the video clips. Color features are used to produce a histogram. Motion features are used to produce a measure of optical flow. Finally, a shape index in developed using texture and geometric features. Retrieval of data can be performed by any of the three types of features or by any combination of the three. The user can provide a primary color, motion direction (up, down, pan-right, pan-left, etc) and a shape that can be determined by a set of prototypes. For example, the user can find a clip of a bird diving by providing a primary color like blue and some motion feature like downward motion. Here is our first example of performing a query based on multiple features of data and retrieve only data in the intersection of all results. Because of the multiple indices, only a linear search can be performed on each feature. Additionally, because the index contains multiple vectors, there is no opportunity to perform a binary search to increase speed of retrieval. Clips that match each feature are retrieved to the user. The linear search requirement results in a complexity of O(n) for this indexing technique.

3.2 Zheng Index

Z.J. Zheng et al. proposed the following feature based indexing method for images. They describe their algorithm as follows “The algorithm … partitions a given binary image into feature images dependent on their intrinsic properties of discrete geometry, and the ten clusters represent Inner, Block Edge, Line Connection, Intersection and Noise points for 1 and 0 values respectively.” Simply put, the image is converted into two distinct decimal values. Retrieval of data would follow the following steps. The user would select a feature image with vectors, x₁ and x₂. The user would then provide probability figures that to determine how close the retrieved images are to the feature image. The system would then retrieve a set of images as follows. The system would check the first index vector of each image and determine if it is a close enough match. If so, the system would then check the second index vector to again determine if it is a close enough match. Only those images where both vectors match would the system retrieve the image for the user. This results in a Big-O complexity of O(n) for this index for the following reasons. The index of each image, both vector numbers, must be examined to determine if it is a close enough match. Because of this the system must check each image with no way of reducing the search time with a binary sort the complexity is O(n). In other words, as double the amount of data is added to the database the computational effort of the index is doubled. Anne H.H. Ng et al. explain the difficulty of indices with multiple vectors used to retrieve data, “To answer a query that involves a composite feature vector, a hierarchical approach is often adopted in which each component of the query is applied against an appropriate index in a lower layer…merged and presented to the user a higher level…. This is inefficient in terms of storage utilization and system performance.”

3.3 MUVIS Index

Developed in 1998, MUVIS is one of the few multimedia frameworks that has been deployed commercially. The system consists of four applications, two of which create the database, an application to browse the data and the “DbsEditor”, which is responsible for indexing the data. DbsEditor main task is described as offline feature extraction. The system provides for both visual and audio feature extraction, which are then integrated to develop the index. The unique and interesting concept behind this system is that it allows the user to define the algorithm of their choice for feature extraction. This is accomplished through the Feature Extraction Interface called the “Fex API”. Color Histograms, Gray Level Co-occurrence Matrix and Gabor Wavelet Transform methods are a sample of the algorithms that have been used in the MUVIS application. Additionally an audio feature extraction framework has been added to the system. The AfeX Framework breaks the audio segment into four classifications silence, speech, music and not classified. Silence is removed and the remaining three are used to produce a vector to index the audio data. To determine the complexity of the MUVIS index we must determine how the two feature extracting...
techniques are used. No explanation is provided as to how the user would query the information, but presumably we have another situation where a vector is created for both the audio and visual information and both are used to retrieve the data to the user. Again, this type of multi-vector searching allows for no binary or other more efficient search technique, so the resulting complexity appears to be O(n).

3.4 Ohtsuki Index
The final feature based index to be examined proposed by Katsutoshi Ohtsuki et. al. uses metadata for index creation. This proposed index is very interesting because it is one of the few feature based indices that can somewhat retrieve data based on the semantics of the multimedia data. This is possible due to the very powerful new technology, XML. They describe the process, “...we are working on a multimedia indexing system that automatically creates metadata by combining automatic speech recognition technology and natural language processing technology.”7 The modules of the system interface through XML based files. The models include acoustic segmentation, speech recognition, topic segmentation, and integration to join the results. The acoustic and speech models work together to process the speech data from the video into a metadata format. The topic segmentation model exists solely to extract segments from the video, for example any scene change. An access interface model allows for fast searches on the metadata produced by the speech recognition. Topic words are extracted from speech recognition and extraction is based on statistical modeling of the relevant topic words. The final result of the process is a “cohesion score” developed using a co-occurrence frequency model, which is used to retrieve the relevant data when a query is performed on the database. This index seems to provide the most efficient retrieval of data in all of the feature-based indices that have been studied. The cohesion scores can be ordered within the index, which will allow for a binary search with a complexity of O( log₂ n). The recursion of a binary search allows for much faster data retrieval.

4. SEMANTIC BASED INDICES
Semantic based database indices are described by John R. Smith “The use of manual annotation allows humans to manually ascribe labels to multimedia documents.”3 The user provides a semantic description of what is included in the multimedia data. These type of databases have traditionally been superior at allowing database users to semantically index and retrieve data. As will be described, some indices are created manually, but once a large enough sample of data is provided the index becomes automatic. These have been described as semi-automatic index methods. For the purposes of this study these types of indices will be classified as manual. Only those indices that are fully automatic will be classified as semantic based.

4.1 Smith Index
John Smith et. al. propose a semantic type of index that produces a vector type index. The user will determine different types of “Lexicons” or concepts that might exist in a photograph or video. For example, the picture may include indoors, outdoors, face or people, which would be the different Lexicons. Model pieces of data must then be provided to the index for the different types of features so that the system can “learn” the features of such a concept and develop detectors. Features include “color histograms”, “wavelet texture”, “color correlogram”, etc. As pieces of data are added to the database the detectors will be used to create Vectors. This entire process allows the user to retrieve data based on the concepts that have been provided using the different features. For example, the database can be searched for images or video containing faces based on wavelet texture. As more features are added to the search the more effective the search will become. Again, the complexity of this index involves using a set of features and retrieve only data that match in each case, or the intersection. So, again we have hierarchical structure with a complexity of O(n).

4.2 Iyengar Index
G. Iyengar et. al. propose a very interesting semantic index. They label their index as automatic but, as will be described, the index is really semi-automatic due to the fact that users must provide information to produce the index. The interesting part of this index is that audio information that accompanies video is used to produce a single vector index. Both speech and audio concepts such as explosions or honking horns are used to produce the index. Here is the first time user must provide data to develop the index, the reasoning behind labeling the index as semantic. Audio-based intermediate-level concepts are provided to the system and a Viterbi decoding is used to segment the audio track in a sequence of audio-event labels. Speech content is also used. With a pre-defined set of query terms the speech corresponding to each “shot” is investigated. At this point, the scores from the audio processing are “fused” to produce a final ranking of the video. The fusing of models forms a vector described by Iyengar, “The scores from the intermediate models were combined to form a vector whose dimensionality is the number of intermediate concepts. A Support Vector Machine (SVM) is then trained in this space with positive and negative examples of the high-level concept.”8 Again, the user must provide examples of the high-level concept. Clearly, this index has manual characteristics. However, this index provides an opportunity for more efficient data retrieval. The index boils down to a single vector score. Presumably, the vectors would be stored in a linear fashion. A binary search can be performed on the scores to retrieve the data. This will result in a complexity of O( log₂ n). It is even possible to create a secondary index to further speed retrieval.

4.3 Lienhart Index
Rainer Lienhart believes, “One powerful high-level index for retrieval is the text contained in videos. This index can be built by
detecting, extracting and recognizing such text.\textsuperscript{9} This text would need to be provided at some point prior to insertion into the database. Leinhardt calls the text “artificial text” because it is added somewhere in the post-processing process and is not a natural part of the video. The fact that text needs to be provided to produce the index is the reason the index can be classified as manual. Examples of this text might include actors’ names, broadcast news topics or dates the video was recorded. The text is first segmented then a recognition algorithm is performed. Segmentation involves extracting all pixels that are part of the text and discarding the remaining pixels. Optical character recognition techniques are then used to recognize the text. All words less than three letters are discarded as not important to the semantics. There are likely to be errors in the recognition of words, so the database allows for approximate retrieval based on Levenshtein distance or the least number of differences between the word provided by the user and the words pulled from the video for indexing. Due to the fact that the index is likely to produce multiple words from each video, the only way to search the index is in a linear fashion. The resulting complexity would be O(n).

4.4 Wilcox Index

The final semantic index to be investigated was proposed by Lynn Wilcox and John Boreczky. The authors employ two separate techniques to produce the index for retrieval of data, annotation and segmentation. Annotation will be shown to be a purely manual index technique. Segmentation is another semi-automatic index. When taken together, this index can certainly be classified as manual. The two types of annotation used to develop the index are notes taken viewing the media and transcription produced from the audio of the video. Using annotation will allow the user to search data based on semantic characteristics. Two types of segmentation, audio and video, are used to develop the second index in this system. Both types of segmentation are developed using Markov models, producing feature vectors. To use the Markov models training data must be provided to the system. The features used for video segmentation are a histogram difference and a motion estimator. Audio segmentation is performed based on changes in speakers or sounds. Wilcox and Boreczky describe how data retrieval is performed, “In order to retrieve the desired segments, the user forms a query by specifying a number of keywords. While it would be possible to search the annotations for these keywords and find their corresponding time locations, this would not provide coherent segments of the media for viewing. In the case where annotations are obtained from note-taking, the time at which the words were entered may not correspond to the topic being discussed at the time, since our notes are rarely that well synchronized. For time-aligned data, text and audio are perfectly synchronized, but knowing the times when keywords occur still does not produce meaningful segments of the media. For this reason, we perform keyword search over segmented media.\textsuperscript{10} The keyword(s) provided by the user are then used to compute an inverse document frequency score. Simply, the inverse document frequency is a function of the number of times the keyword(s) appears in the data. While there is no explanation of how the scores are searched, there are multiple possible ways. If the scores are stored in a linear fashion, then again a binary search can be performed on the scores to retrieve the data. This will result in a complexity of O(\log_2 n). It is even possible to create a secondary index to further speed retrieval.

5. RESULTS

Three of the feature-based indices investigated appear to have a complexity of O(n), where doubling the amount of data in the database would double the computational effort. The final index’s complexity in the worse case scenario would be O(\log_2 n). Meanwhile, two of the semantic based indices’ complexity seemed to be O(n). The other two indices’ complexity would be O(\log_2 n) and possibly better. These findings would indicate that semantic based database indices, on average, would allow for faster data retrieval than feature based.

6. CONCLUSIONS

The sample size of four of each type of index used for this study seemed adequate due to the relatively few indices that exist or have been proposed in academic study. The first point of interest regarding these indices is that their main purpose is to convert very complex data into a more conventional type of data that can be used as an index. Multimedia data that includes many physical and semantic characteristics is used to produce a vector of integers or a score in many of the indices examined. This conversion is what provides for faster retrieval of the data. The simpler the resulting data used for the index the faster the retrieval. An index based on a simple data type that provides a robust method of retrieval is what is needed. The results of the study seem to indicate that semantic indices allow for faster retrieval of data than feature based indices. These results are only part of what needs to be considered when looking at multimedia type indices. First, the two types of indices do not provide the same type of data retrieval. Users of the database are better able perform searches based on what is happening in the video or image. This can have good and bad implications. Allowing the user to retrieve a picture of a horse or video of a horse running through a field is very useful. However, the manual aspect introduces the problem of user interpretation. In other words, users of the database are relying on those providing the information for the index to provide accurate data. In the semantic based indices that were studied, if the text in the video, transcriptions does not match what is being seen in the media then the index is not useful. On the other hand, feature indices can provide some semantic-based retrieval. Consider the case of retrieving video of a horse running through a field. Pure semantic based indices will probably allow the user to enter horse, field, and running. This level of specifics might not be possible with an
feature-based index, but it is probably possible to provide the database with an image of a horse to retrieve the desired video.

The second issue that must be considered is the time it takes to develop the two types of indices. While this study asserts that semantic database indices provide for faster retrieval, they rely on information that must be provided by the user prior to insertion into the database. In the case where large amounts of multimedia data going to be stored, the process of providing information for the index might become quite cumbersome. Feature based indices require the user to provide no information prior to insertion of the data. Again, this may be an important factor if the information does not exist and a large amount of data is to be stored.

Additionally, a factor probably as important when considering indices is effectiveness of retrieval. If the index is not providing the correct data to the user, then speed of retrieval makes no difference. A worthwhile endeavor and good compliment to this study would be to research the effectiveness of the indices to determine the percent accuracy of data retrieved.

It is hoped that this research, along with complimentary studies will be used to provide optimal index and query methods to users of databases that store multimedia type data. The result will be indices that evolve convert complex multimedia data into simple terms for fast and efficient retrieval.

7. SUMMARY

This paper studied the speed of data retrieval using the two primary types of indices for multimedia type data in an effort to help develop produce optimal performance. Four of each type of index were examined using Big-O complexity to approximate the speed of retrieval. The results indicated that semantic based indices provide for faster retrieval and that this is one of the important factors that must be considered when developing or introducing an index into a multimedia database.

8. REFERENCES

4. Edoardo Ardizzo, et. al., “Content-Based Indexing of Image and Video Databases by Global and Shape Features”, IEEE 8/96 Volume 3 pp140-144