ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval
(Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

Experiences with Content Based Retrieval of Multimedia Information

Desai Narasimhalu and Mun-Kew Leong

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

Experiences with Content Based Retrieval of Multimedia Information

A. Desai Narasimhalu and
Mun-Kew Leong

Institute of Systems Science, National University of Singapore
Heng Mui Keng Terrace, Kent Ridge, Singapore 119597

Abstract

In the last four years, the Institute of Systems Science (ISS) has developed various content based retrieval engines which work on text, images, and sound. In working with these multimedia, we find they naturally divide into two kinds: encoded and unencoded. We give characteristics of these two kinds of data, and show how they differ with respect to the key issues in multimedia retrieval: feature identification, segmentation, normalization, classification, indexing, similarity measure, filtering, and retrieval. We provide concrete examples of these differences from various content based retrieval applications developed at ISS, specifically, a multilingual freetext search system, a photograph archival system, a facial image recognition system, a trademark archival and retrieval system, and a MIDI audio file retrieval system.

1 Introduction

The Institute of Systems Science (ISS)¹ of the National University of Singapore has been developing multimedia databases and database engines of different kinds for several years. In the past four years, the Archival and Retrieval of Multimedia Information (ARMI) group at ISS has been focusing on developing content based retrieval engines. We have operational prototypes of multilingual free text engine, shape matching engine, phonetic engine, image feature engine, fuzzy retrieval engine, colour retrieval engines and most recently a MIDI melody retrieval engine.

These engines have been tested in real-life applications such as the FACEit face identification system, the STAR System for Trademark Archival and Retrieval, and the MultiSearch multilingual free text product. Most of these applications are presently in the process of scaling up to handle realistic amounts of data, e.g., gigabyte collections of multilingual text, large databases of facial images, and thousands of visual trademarks.

The ARMI group has also been developing a multilingual fuzzy thesaurus management system called Theseus which integrates with many of the engines and allows a greater degree of semantics to be incorporated into the various retrieval processes. And recently, we have started looking into new metrics for measuring the effectiveness of retrieval in such applications.

This paper builds from the experience garnered in the course of working with these multimedia retrieval systems. In Section 2, we suggest that multimedia information is composed of two kinds—*encoded* and *unencoded*—and describe some of their characteristics. The following section lays out the key issues in multimedia information retrieval, specifically, feature identification, segmentation, normalization, classification, indexing, similarity measures, filtering, and retrieval. In Section 4, we describe several systems developed at ISS focused on content based information retrieval and explain how they embody the issues mentioned. We conclude with some thoughts about meta-issues in multimedia retrieval.

---

¹ Please see <http://www.iss.nus.sg/> for more information.

2 Characteristics of Multimedia Information

Multimedia information can be said to be of two broad types—*encoded* and *unencoded*.² What we mean by those terms is not in reference to the *format* used for their representation in physical media, but rather with respect to their natural form, i.e., an extension of the notion of a “glyph” or “word” in a language to include the multimedia. An encoding then is akin to the process of defining a “language” primitive.

As a working definition, we will say that information is encoded in the case where it is represented in a standardized commonly accepted set of descriptors, i.e., the information is normally divided into units of logical meaning where these units are a physical representation of the information in some given collection of symbols. And we will say that information is unencoded if and only if it is not encoded.

Examples of encoded information are text files and MIDI files. In the former case, the information is commonly broken up into units such as words, phrases, and sentences (at least for romanized text), and in the latter, into units based on key presses, pressure levels, and time intervals.

Unencoded information may also be thought of as information that is in its “raw” form, i.e., where there is no commonly accepted set of physical descriptors which fully represent the given information in the general case.³ Examples of unencoded information are image files and video files. In neither case is there any sort of a “language” which can adequately describe the information contained in the files.

Our ensuing discussion on the characteristics of multimedia information will make use of this definition of encoded and unencoded information.

We will also use the term *feature* for those descriptors of particular information which are useful for characterizing that information. Thus, colour may be a feature of an image, rate of change of frequency a feature of an audio file, part-of-speech frequency distributions a feature for a text, etc. A feature is not an innate part of any information, but an interpretation we derive by the imposition of a structure on that information. We may use *feature* to refer both to the feature class and to instances of that class; the context should be unambiguous.

2.1 Characteristics of encoded information

Encoded information has a discrete set of tokens or symbols. Examples are words in a given natural language and MIDI codes which represent parts of synthesized music.

Encoded information can have a large number of symbols. In fact, the class of symbols may be open, i.e., new symbols may be added to it. Examples are words (or their equivalent) in any natural language.

Each symbol or token can be interpreted as a feature. The number of roles that a token can play is limited. For example, in the English language, certain words may assume different part-of-speech roles in different sentences. However, the number of such roles is limited just as the number of parts of speech is limited (in English, normally restricted to noun, pronoun, verb, adjective, adverb, preposition, conjunction, interjection, and article).

Parts of speech cannot themselves be features since the set of values (or words) that they assume do not naturally form a dimensional space which is necessary for non-exact match searching. In other words, in languages such as English, there is no partial ordering defined over the range of words that form the domain values for a part of speech. Note that such an ordering can imposed within a given set, but does not occur naturally. In many cases, to impose such an ordering *is* to do feature extraction.

The value space of encoded information is often linear and one dimensional. This makes for easier indexing and searching, and allows for approximate (neighbourhood) matches.

Encoded information is often represented in a structured manner. For example, documents have chapters, sections, subsections, paragraphs, etc. MIDI codes are also structured, but according to the conventions of music rather than text.

2.2 Characteristics of unencoded information

The characteristics of unencoded information are more complex than for encoded information. It will help to keep in mind some examples of such information, e.g., digital sound, image, and video, when we consider the following.

The features of unencoded information can normally assume a range of values which are continuous. For example, the colour of an object in a photograph, the length of a nose in a facial image. This continuous value space poses

²This difference is analogous to the *symbolic* vs *iconic* distinction found in anthropology and semiotics.

³There might be such physical descriptors understood within a given narrow domain, but not generally.

a challenge when it comes to abstracting (or extracting) indexing features from unencoded multimedia information. Often, the value space of the features have to be quantized and classified for manageable representation and processing. This could create many-to-one mappings which is troublesome and hence undesirable when it comes to indexing such features.

The value space of unencoded information can be non-linear and multidimensional. For example, even something as simple as colour requires more than one dimension for representation. Two schemes for representing colour are by using their Red-Green-Blue values (RGB values), or by using Hue, Saturation and Brightness (HSB values). Another example; if the nose in a facial image is considered as a feature or token, its length and its width at different points along that length are some of the different dimensions that are used to represent the value space of a nose.

The multidimensional nature of unencoded multimedia information allows multiple interpretations by users. These interpretations are often coloured by the knowledge base or structures of a specific user. Hence, unencoded multimedia information is often more context sensitive than encoded information.

Unencoded information is often semistructured or unstructured. For example, an image has no structure.⁴ Even when an image contains an object that is structured, for example, the scanned image of a document, the structure of the document is not captured in (i.e., is not a part of) the image.

3 Key issues in Multimedia Information Retrieval

Multimedia information often consists of both encoded and unencoded information. The major challenge in representing multimedia information in a computer such that it can be efficiently and accurately retrieved is the *granularity* and *accuracy* of representation. The former refers to finding the right level of detail to capture and is a tradeoff of space and time on one side and the completeness of representation on the other. The latter refers to how well the representation captures the information (or particular aspect of the information) for retrieval purposes.

Figure 1 gives the different steps in getting data into a computer in a suitable way for later retrieval on request.

Figure 1: Steps in getting data into a computer for retrieval

The key issues in multimedia information retrieval are feature identification, segmentation, normalization, classification, indexing, similarity measures, filtering and retrieval. We will discuss each of these issues with respect to encoded and unencoded multimedia information.

3.1 Feature identification

Encoded information, by its very definition, has the advantage that features of interest for retrieval normally correspond to the innate encoding scheme, i.e., those features are pre-identified and encoded. Examples are words in texts and permissible values of MIDI codes. In some cases, the innate encoding scheme includes features which are not useful or are considered irrelevant for certain applications, e.g., retrieval. In the case of text, these are the pronouns, prepositions, conjunctions, etc.. In such situations, these features are identified in a “stop list”; elements in the stop list are excluded from further consideration.

Identifying features of interest in unencoded information is much more challenging. There aren’t any predefined schemes which correspond to *all* features of interest. For example, in the case of a collection of images, one may think of using colour as the feature of interest. However, in such a case, a query asking for red cars in that collection will also retrieve a red tomato.

⁴ Again, this may not be the case in a narrow domain, but we are talking in general here.

On the other hand, one has to be careful of features which may be easy to identify in a particular collection, but these may not be reliable (for example, not consistent, not invariant, or not time-insensitive) and should be ignored. For example, in a mugshot (face recognition) application, hair colour and shape are easily extracted from the given image of a face. However, since hair styles and colour are easily altered, that feature is considered to be less reliable.

One also has to be careful of features which are easy to retrieve, but which may take one up the garden path. This occurred in full text retrieval, where the ease in identifying words formed the basis of the Boolean search paradigm. More recently, the full text retrieval community has concentrated more on identifying the *meaning* of a unit of text (as a preferred feature for retrieval) rather than just the words in it, since words are often ambiguous in context, or may be part of a phrase, a frozen idiom, etc.

Hence identifying reliable and robust features is a major task for systems dealing with unencoded information.

3.2 Segmentation

Segmentation is the process of extracting features or tokens from the given multimedia information object. When we refer to segmentation, we normally refer to automatic segmentation by a computer program.

3.2.1 Segmentation in encoded data

Most types of encoded data have well understood delimiters or implicit structures which allow tokens to be extracted from an incoming information stream. Such encoded information can often be segmented using just a single pass tokenizer.

In the text domain, spaces, punctuation, and other “white-space” are examples of delimiters of encoded data. An input text stream is divided into segments, i.e., into words, based on these delimiters. Other types of encoded data such as MIDI codes also have well defined semantics. In MIDI, each different bit or number of bits that form a token can be determined either by predefined length and/or by delimiters.

The exceptions to the above rule aren’t many but are significant. One example of encoded information which does not have comprehensive delimiters of the features of interest is the Chinese script. While it is possible to segment the data into character tokens, those do not correspond to the features of interest (i.e., akin to splitting up an English text stream into letters rather than into words). It also does not have an unambiguous internal structure. Proper segmentation of such encoded data requires multipass tokenizers, dictionary driven systems, semantics driven systems, or combinations of the above.

3.2.2 Segmentation of unencoded data

Segmentation of unencoded data is relatively more difficult than for encoded data. There is the occasional collection where such a case is relatively straight forward. For example, considering images again, if the data are clean bitonal images, then segmentation (the division between the two tones) is straightforward and tokens can be easily generated. Of course, in some cases, it is difficult to decide which tone, and hence which feature, is the useful one. For example in some bitonal (black and white) drawings, white is usually reserved for the background and the features are done in black on this white background, but it is possible to do the reverse, e.g., blackboards.

Most of the time, however, one does not get clean bitonal images, but rather continuous valued characteristics with imprecise divisions between them. An example would be the characteristic of colour in a given photograph. Automatic segmentation of such continuous valued features is very challenging. At the present moment, such tasks require human intervention for placement of registration marks or landmarks, drawing a boundary, etc. It would be more accurate to say machine-assisted segmentation rather than truly automatic segmentation.

In any case, the accuracy of the tokens extracted from such data is almost always in doubt. This is especially true if the boundaries of such features are ill-defined or if they are multidimensional. Such less than perfect extraction will affect the classification, indexing and, hence, the computing of the similarity measures for multimedia information retrieval.

3.3 Normalization

Encoded data, for example text, do not normally need normalization. They may, instead, require verification. This is to ensure that the features extracted by segmentation are within the respective feature spaces. An example of verification

is to make sure that features which are supposed to be dates do actually fall within the syntax of dates, no February 30th, for example.

Unencoded data, on the other hand, often has to be normalized, and sometimes more than once. The normalizations may include some that are dependent and some that are independent of applications.

Examples of normalization in a collection of images which are independent of application are cleaning up the noise in an image, adjusting the gray scale contrasts of all images in the collection such that they fall within a prescribed standard deviation, and maintaining a consistent aspect ratio for each image.

Examples of normalization which are dependent on the application domain are adjusting the tilt of a face in a collection of mugshots or making sure that the text images are right side up in a document imaging system.

Normalization is a key step in the processing of unencoded information and will greatly affect the rest of the steps. Hence, it is important to understand the different normalizations required for a given collection of information and then find the best possible normalization routines before the other modules are developed.

3.4 Classification and Indexing

We have said above that encoded information is almost always easily segmented because it has well understood delimiters or implicit structures. In addition, encoded information is often preclassified. Examples are department number, employee number, and product code. In many cases, these classifications are also the features of interest for retrieval. Thus indexes are built based on this classification information. For example, in an employee database collection, indexes are normally built on *employee number* to pull information about a specific employee out easily.

Again, as for segmentation, unencoded information is not pre-classified. Some types of unencoded information need classification and others do not.

An example of unencoded information that does not need classification is a common attribute such as salary, found in any employee database. Such information is not normally indexed. Some examples of unencoded information which need classification are bibliographic and full text records.⁵

Other examples of unencoded information that normally need classification are features such as irregular shapes, images whose feature space assume continuous values, and images which do not have dominant object(s) in them.

Certain unencoded information may defy crisp classification. In such a situation, one can resort to the use of fuzzy set theory. However, the biggest challenge lies in defining the membership value distributions over the feature space domain. This task is often influenced by the application under consideration.

In any case, the classification of unencoded information is equivalent to some form of encoding. This encoding then serves as a (possibly non-unique) signature for the multimedia information object. This signature is the basis for indexing and retrieval.

Large numbers of multimedia information objects that may have fuzzy membership in different classes are best organized into a classification tree for indexing purposes. Such a classification tree will start with a root node representing the entire collection and will be organized into several child nodes at each following level. Each level will represent the classifications across one dimension.

Such a classification tree is in reality populated only at the leaf nodes. Every intermediate node is represented by a "prototype" belonging to that subclass. The prototype is chosen as that object which is the centre of gravity of the collection for the given feature.

The subclasses at any level in the tree serve as a coarse grained index. This type of organization best fits browsing applications where users can go down one level of the classification tree at a time. However, queries can be translated into feature classes and this will allow direct access to one of the populated nodes of the classification tree.

3.5 Similarity Measures and Retrieval

Encoded information is crisp by its very nature since there are well-defined delimiters and/or internal structure. Hence similarity for such information is an exact match between the value of the qualifier in a query and value of the attribute of interest. This is handled traditionally through indexing techniques.

Unencoded information on the other hand may not all belong to a crisp classification. The classification along different feature spaces provides the signature (as describe above) for a multimedia information object.

⁵ We referred to text before as *encoded* information. This is not an error. In this case, we are talking about *full text*, i.e., documents that are not segmented or classified. In such a case, it is not the words in the text that is the information of interest (and hence encoded) but rather the meaning of the document as a whole (and hence unencoded). For example, an arbitrary document is not easily segmented into its constituent concepts.

Given the known difficulties in being able to specify a query for a multimedia information object such that it is both precise and complete, it is often best to return a neighbourhood result rather than trying for an exact match result.

A neighbourhood result is often derived using some form of distance measure between the signature of the query object and the signatures of the objects in a collection. The objects in the collection are then ranked in the decreasing order of similarity. If the number of objects in a collection is large then some criteria for thresholding is applied to reduce the number to a manageable one.

Additionally, given that there is the possibility the query object has been imprecisely and incompletely described, there may then be a need to iterate through the results shown by picking those objects which are closest to the one being sought. This selection will then be used to refine the query which is then resubmitted and a new set of results will be displayed. This process is often referred to as relevance feedback.

3.6 Context sensitivity and Filtering

Multimedia information such as audio and visual is often amenable to multiple interpretations. Also, when there is a large amount of information available, a simple flooding of all this information to a user will lower the usefulness of this information. It is for these reasons that it is important to be able to understand a user's interest and hence to build a user profile.

A user's profile can serve as one of the means of defining the context. Meta-information or information about the information may serve as another source for establishing the context. However, be it meta-information or user profile, such knowledge has to be captured in some manageable structure. These knowledge structures can then be used for establishing the context of a user's query and that of the information.

Once the contexts of the user query and the multimedia information are established, then suitable filters can be used to direct only the most relevant information to a user. This will make sure that the user is able to receive information that is likely to be of the most interest to him or her. A trivial example from a text application is the disambiguation of word senses, e.g., if the user's profile is oriented towards financial information, then the sense of the word "bank" used in a query would be treated as a financial institution and not the edge of a river.

There is a danger, however, in completely filtering out information that does not fit the context. This is the possibility of alienating a user from new information that may not be of past interest but which may create new interest, i.e., the possibility of serendipitous occurrences. A simple use of contexts and filters impedes the presentation of potentially interesting information to a user.

Hence it may be useful to develop topics not of interest to a user in his or her profile, and give information that may not have been of past interest to that user at a lower ranking. This puts the onus of reading potentially interesting information on the user. The system should also monitor such topics read by the user, and amend the user profile as necessary. One of the ways of developing such new topics is to cluster a large number of user profiles. Within a single cluster, there will be a large overlap (by definition) but there will also inevitably be some "leftover" topics which are part of some profile(s) but not part of others. If we assume that users who have similar profiles may also have similar interests outside those profiles, then those leftover topics are worthwhile exposing to others in the same cluster.

Alternatively, applications can also be built which will actively present such potentially interesting information in a window on the output screen that co-resides with the regular result window. This will ensure that potentially interesting information does not always sit at the bottom of the stack and remain unseen. Such active presentations are a core part of data mining applications.

Finally, it is suggested that only the headlines or titles of potentially interesting information (and some keywords, perhaps) be presented rather than the whole of the information. This would allow the user to browse through a large number of such potentially interesting information and only go into the details if there is a serendipitous match.

4 Multimedia Retrieval Applications in ISS

We have looked at the key issues in the process of getting multimedia data into the computer and retrieving it. We now look at how these issues were addressed and implemented in several content-based multimedia retrieval applications developed at ISS.

4.1 Picture Archival System

The Picture Archival System (PAS) was developed at ISS about five years ago. It archived several hundred images of various sporting activities taken from the 1988 Olympics. Each image was annotated by a text description, and this text description was indexed and used to retrieve the relevant images. Users were presented with pictures corresponding to word or concept based text search, and they could refine their queries using relevance feedback on the selected images.

With regard to the key issues mentioned, feature identification, segmentation, and classification on the unencoded image data was done manually by human indexers. The resulting encoded text data was normalized with respect to the vocabulary used to describe the pictures, and matching was simply word equality. Because of the imprecision of the text description, neighbourhood results (see Section 3.5) were returned in ranked order, and iterative querying (relevance feedback) was employed. The indexing, search, and retrieval components are basically for encoded data; i.e., one meta-level of description higher than the actual multimedia content.

This model is representative of many of the existing commercial database systems which support multimedia data. The binary non-text information is stored as BLOBs (Binary Large Objects) and the meta-information is stored in the database, indexed, and used for retrieval.

4.2 MultiSearch Full Text Search Engine

MultiSearch [4] is a universal multilingual full text search engine (see Figure 2), and the first such system developed

Figure 2: The MultiSearch Multilingual Free Text Search Engine

using the Unicode standard [5, 6]. It is a fairly traditional probabilistic model system with, however, some performance constraints, specifically, that a 128-megabyte single processor server shall be able to support up to twenty full text ranked retrievals on gigabytes of data, returning the results in under thirty seconds. In addition, updates to the indexes shall be reflected in the query results within five minutes of verified entry. It also supports a multilingual thesaurus that provides, for example, multilingual query expansion and concept search.

As we explained earlier, full text information is unencoded. Feature identification occurs at the level of words (terms), with IDF as a weighting factor for the importance of each term. Segmentation is imprecise and done at the level of words (or characters) rather than the concepts or themes which may be said to represent the content (meaning) of the document. This may change, however, as more semantically driven tokenizers (segmentors) are plugged into

the system. They are already required for segmentation of non-ASCII represented information (e.g., Chinese script) and can be expanded to enhance the indexing of, say, English text as well. Normalization is on the range of the IDF and TDF values, and is built into the similarity function. This measure is a weighted function of the co-occurrence of words in the query and in the indexes.

As we can see, it turns out that the issues mentioned above are key not for the class of multimedia data, but, more precisely, for the class of unencoded information.

4.3 FACEit Face Recognition

The next system we describe is currently called FACEit, but had an earlier incarnation as the CAFIIR⁶ system [1, 7]. This system stores and manages facial images oriented towards a criminal identification application. Various input methods, i.e., ways of specifying an image as a query, are possible, including facial composition (see Figure 3), loading

Figure 3: Composing a facial image for query

scanned images, processed images (e.g., age regression), fuzzy querying, and hierarchical browsing.

This system illustrates many of the key issues discussed earlier. Feature identification is based on cephalometry, i.e., on bony landmarks on the face which are relatively invariant over aging and facial expression. Seventeen such landmarks are identified, as shown in Figure 4. and a signature vector formed as a function of the regions created. The landmark registration relies on a classification of the image content into semantic (facial) features such as eyes, chin, nose, etc.

⁶ Computer Aided Facial Image Inference and Retrieval

Figure 4: Landmarks registered for a facial image

No automatic segmentation is currently being done, though this is in development. Normalization is done on the images to ensure a consistent size for the face, and the tilt angle of the face in the image, as well as the colour balance. As expected, the signature vector created from the landmarks is indexed, and retrieval and rank is based on the similarity between the signature of the query image and the signatures in the index. So searching returns a ranked list of images as in Figure 5. The weighting of the various facial attributes can be controlled (the panel at the lower right) and relevance feedback is also possible.

Classification is used in other aspects of the system. For example, in the aging module, certain facial attributes are adjusted to account for general trends due to aging; these, of course, depend on knowing which of the extracted feature landmarks belong to what class of facial attributes. Additionally, the same attributes are ones which are to be defined (perhaps incompletely) during fuzzy querying, and are used as the partitioning dimensions in the creation and browsing of the classification tree.

Context sensitivity is an important issue in this application. Many of the rules which are used in the aging module, or the weights applied to the various landmarks during searching are dependent on the composition of the image collection. There are differences for male and female populations, as well as from different racial genotypes. This is context sensitivity of the system.

There is also context sensitivity in the query side of the system. There are fairly distinct classes of possible users, ranging from untrained witnesses trying to describe a perpetrator to experienced law enforcement personnel who focus on invariant attributes such as the shape of the face rather than easily altered attributes such as hair colour or hair style.

4.4 STAR: System for Trademark Archival and Retrieval

In the previous system, the domain of interest is very focused; on just facial images. This allows a great deal of semantics to be built into the system as explained above, e.g., with very few exceptions, everybody has two eyes, two ears, a nose, a mouth, and all laid out in the same general locations in the face.

The System for Trademark Archival and Retrieval (STAR) starts with the same underlying engine as in FACEit, but adds other techniques to deal with the much greater variety found in handling trademark images. The very nature of trademarks works against common semantics; people who register trademarks tend to either be as different as possible from their competitors (brand differentiation) or as close as possible to a leading competitor (brand transference).

Nevertheless, the STAR system works just as well in its own way. Its input method is almost exclusively image driven (see Figure 6) and requires quite a bit more analysis than in the FACEit system. There are six kinds of similarity

Figure 5: Results from a facial image query

functions which are used in STAR. These are word-in-mark, abbreviation, phonetic, overall meaning, graphic meaning, and component meaning. These correspond to the various ways of feature identification, segmenting and indexing the unencoded image data. The interesting thing is that the first three methods rely on textual properties and are therefore techniques for encoded information. The remainder are techniques for unencoded information.

With respect to our key issues again, feature identification is a manual process, though the segmentation is automatic once identification is done. A given trademark is either a *word-in-mark*, a *device mark*, or a *composite mark*. The first means that the trademark consists solely of words (or characters in some script), e.g., the IBM logo; the second means that the trademark consists solely of some figure without words, e.g., the Mercedes logo; and the third is that the trademark comprises both words and figures, e.g., the Hewlett Packard logo. In the first and third cases, an indexer has to analyze a given image and enter the words (or characters) contained in it. In the second and also in the third case, the indexer specifies what is the primary component image and the background of the contained image and the system will extract the feature. Basically, the extraction comprises a coarse and a fine filter to get a rough shape description and a more precise shape description. Both descriptions are affine functions, i.e., they are invariant with regard to size, orientation, lateral symmetry, transformation (location), etc. These descriptions are the features extracted for the non-text component of the trademark. Note that normalization is built into these descriptions by their affine nature. Again, these features are used as the signatures for the trademark figures and are indexed accordingly.

Classification is also done on the non-text components. There is an international standard for textually describing these components, known as the Vienna Classification. Trademark officers are specially trained to try to describe non-text unencoded content in a consistent manner using this classification, though the fallibility of the human factor is well recognized. This is another example of context sensitivity in the data domain.

The six similarity measures apply wherever possible. As mentioned earlier, the first three make use of the textual nature of the word-in-mark so are basically encoded and are based on exact-match, containment, or phonetic similarity. Overall meaning is a full text description of the graphic elements and can (and has been) treated both as encoded and unencoded data depending on the retrieval method applied. The last two are strictly unencoded and make use of the signatures mentioned above. In Figure 7, only the last three columns are active since the search is entirely on the graphic elements.

Figure 6: Analyzing a trademark query image

Figure 7: Six dimensions of similarity in trademark searching

4.5 MIDI Melody Search and Retrieval

The last system we are describing is in a different realm entirely. It is a system which has been used to index and retrieve MIDI melodies given a query tune (i.e., a set of notes in sequence). Since MIDI control codes are limited to 128 possible combinations, they can be mapped onto a double-byte printable character set for display and analysis. A query in the system is merely a list of notes as shown in Figure 8. Submitting the query returns a ranked list of tunes

Figure 8: Specifying a MIDI query

which match the query.

As we mentioned earlier in the paper, MIDI is an example of encoded information. Nevertheless, the same key issues apply here. A MIDI data stream is basically a series of control codes indicating which keys in some specified MIDI device have been depressed, the velocity at which they were depressed, the time interval to the next event, or the release of a depressed key. Feature selection for tune retrieval consists of ignoring most of these codes and focusing only on which keys have been depressed. Since this is encoded data, that is very simple to do. Classification is implicit and the non-note producing codes are discarded. Incidentally, normalization is unnecessary in this case, which is as expected for encoded information.

The segmentation step, however, is not so simple. The tune or the melody in a song is not easily separated from the accompaniment. The task would be much more difficult for unencoded data, but even in this case, since the MIDI control codes do not specify any semantics, separating the melody from the accompaniment is done via heuristics, e.g., the melody is normally in a higher key than the accompaniment, etc.

The similarity measure for MIDI is qualitatively different than for the other content based engines described above. There the feature space is large but the order of the features is unimportant, i.e., they are static with respect to time. In the case of MIDI, however, the feature space is small (only 128 codes, of which about half are discarded as non-notes) but the arrangements or order of the features is crucial.

The equivalent of full text retrieval for MIDI is very difficult because of the problem of segmenting the music into unambiguous logical units. Other techniques such as digraphs and trigraphs (or di-notes and tri-notes, to be specific) also fail because of the small number of codes used. Falling back onto their encoded nature, it is possible to do a serial match with a useful side-effect which offsets the slow speed of such a matching algorithm. This is that the system is able to qualitatively identify the type of differences between the notes in the query string and the partially matching

data string. For example, viewing the result in Figure 9, we see that the system not only ranks the tunes returned, but

Figure 9: A single midi file returned

identifies the types of errors between the query and the tune returned (the lower right corner of the figure).

5 Conclusion

We have spend most of this paper discussing the key issues in getting multimedia data into a computer and retrieving it on demand. As we have seen, the issues mentioned apply not simply to unencoded multimedia data but also to encoded multimedia data, and unencoded non-multimedia data. In fact, we can safely say that these are rather key issues that must be addressed in any form of content based retrieval. It just so happens that most applications which are termed content-based retrieval happen to focus on unencoded information.

We do not have the space here to discuss some other issues which apply to content-based retrieval. These are the meta-issues related to multimedia information retrieval. They include such issues as benchmarking and testing, evaluation of retrieval methods, data fusion between different (multiple) sources of multimedia evidence (as in the STAR system), and of course, the perennial problem of scaling up to realistic application sizes.

References

- [1] Y. H. Ang, Face Recognition Based on Anatomical Features, ISO/IEC JTC1/SC29/WG11 Working Group on Coding of Moving Pictures and Associated Audio, November 3, 1993
- [2] G. Phanendra Babu, B. M. Mehtre, Mohan S. Kankanhali, Color Indexing for Efficient Image Retrieval, ISS Technical Report TR94-163-0, 1994
- [3] C.P. Lam, J.K. Wu, B.M. Mehtre, Shape Feature Measures and Complex Similarity for Conflicting Trademark Retrieval, submitted to IEEE trans PAMI, also ISS Technical Report, 1995
- [4] M.K. Leong, P. Oh, P. Senthil Kumar, A Universal Multilingual Free Text Engine, proceedings of TREC-4 conference
- [5] The Unicode Consortium, The Unicode Standard Version 1.0, Volume 1 & 2, Addison Wesley, 1991/1992

- [6] The Unicode Consortium, Proceedings of the Seventh International Unicode Conference, The Unicode Consortium, California, USA, 1995
- [7] J. K. Wu, Y. H. Ang, P. Lam, H. H. Loh, A. Desai Narasimhalu, Inference and Retrieval of Facial Images, Multimedia Systems, Vol 2, 1-14, 1994
- [8] J. K. Wu, B. M. Mehtre, Yong Jian Gao, A. Desai Narasimhalu, STAR—A Multimedia Database System For Trademark Registration, International Conference on Applications of Databases (ADB-94), Sweden, June 21-23, 109-122, 1994