Efficient Use of Semantic Annotation in Content Based Image Retrieval (CBIR)

V. Khanaa,Bharath University., Dr.M.Rajani., Director., Research., Bharath University,
K.Ashok Augustine Raj , Asst. Prof., Dept. of Bioinfo Bharath University, Chennai .

Abstract*—With the growing amount of people using the Internet, and creating digital content and information, knowledge retrieval becomes a critical task. Ongoing efforts provide frameworks and standards for annotating digital and non-digital content semantically to describe resources more precisely and processable in comparison to simple descriptive structured and unstructured metadata. Although the MPEG group provides with MPEG-7, a useful and well defined theoretical framework for the creation of semantic annotation, retrieval of the annotations is not discussed. In this paper we present a retrieval process for MPEG-7 based semantic annotations founded on well proved information retrieval techniques, namely query expansion and regular expressions. Additionally NWCBIR, a prototype implementation for semantic search and retrieval will be presented.

***Index Terms*—MPEG-7, NWCBIR, Semantic Search and Retrieval.**

I. INTRODUCTION

In traditional libraries metadata plays a central role, as keywords and taxonomies provide short and meaningful descriptions and cataloguing. It provided for a long time the only alternative way to inspecting all available books of finding what users need within the inventory of a traditional library. In digital libraries this context was "digitized" but remained quite similar to the original concept. Current trends show that the efforts and achievements of the information retrieval research area are integrated to enhance digital libraries ([Lossau2004], [Summan2004]). On the other hand much of the metadata based methods of digital libraries have been adopted in knowledge management, knowledge discovery and information retrieval (i) for providing an application area for techniques like metadata extraction and automatic taxonomy creation and (ii) for enhancing knowledge management, discovery and retrieval by using metadata based retrieval techniques. Especially in the latter field techniques based on query expansion using thesauri or ontologies are very successful. Multimedia retrieval heavily depends on such techniques and the appropriate metadata. Content based image and video retrieval requires the pre-processing and indexing of content before query time, this pre-processing is the extraction of low level metadata. An often discussed topic in content based image retrieval is the semantic gap ([DelBimbo1999], [Smeulders2000]), which defines the difference between automatically extracted image features and the understanding or description of visual information of a user. If the semantic gap can be bridged by retrieval mechanisms no annotation would be necessary. Right now semantic descriptions have to be created, at least in parts, manually. Human Computer Interaction (HCI) methods and information retrieval methods exist, that support the user in the annotation task. Different formats and approaches for the storage and definition of semantic descriptions are currently discussed and in use, wherefrom MPEG-7 is one of them.

II. MPEG-7 BASED SEMANTIC DESCRIPTORS

The standard being used to define the way of handling the metadata has to be a lot more powerful than *EXIF* or for instance Dublin Core [Hunter2000]. *DC* only defines 15 core qualifiers, which can be understood as metadata tags, which can be filled by the user. A combination of Dublin Core and adapted Resource Description Framework structures, *RDF*, would at least permit a structured storage of graphs and a quality rating, although content based image retrieval would not be supported. An import of the *EXIF* information to a *RDF*-based structure is possible. The main proposition against *RDF* is that there exists, at this time, no standardized structure for saving all or most of the metadata defined in the requirements above. Although it would not prove impossible to create such a structure, to gain interoperability with other systems and implementations, agreeing on the same RDF based enhancements with all other developers or vendors is necessary. Based on these facts a much better choice is MPEG-7 [Benitez2002].

III. REALIZATION OF NWCBIR RETRIEVAL TOOL

NWCBIR gives the user the ability to retrieve annotated photos. Due to the fact, that this is experimental software the retrieval mechanism is file system based. All MPEG-7 documents found by NWCBIR in a specified directory and in further sub-directories are searched. Figure 1 shows the simplified UML diagram of the NWCBIR System's Retrieval Tool.

NWCBIR offers three different ways to search for a matching photo:

1. Defining search options through textboxes with various options.
2. Content based image retrieval using the visual descriptors *ColorLayout, ScalableColor* and *EdgeHistogram* defined in the MPEG-7 standard.
3. Searching for a similar semantic description graph.

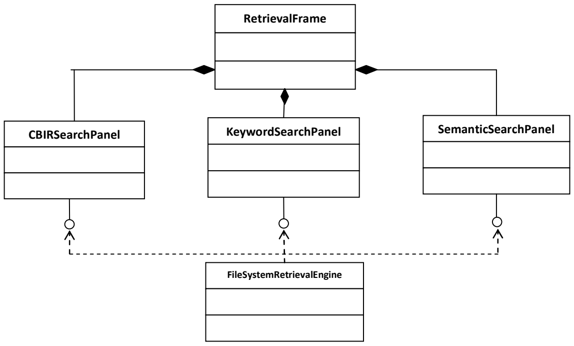

Fig. 1: Simplified UML diagram of the NWCBIR System's Retrieval Tool

IV. Semantic Search

The input for the retrieval process is a semantic description, given by the user. The output lists all relevant semantic descriptions in the database sorted by their relevance compared to the query. To achieve these goals a mathematical and data model for the semantic descriptions has been built and a retrieval strategy has been created.

V. The Model of the MPEG-7 Semantic Description Scheme

All semantic descriptions consist of nodes, which are semantic descriptors extended from the semantic base descriptor, and relations, which interconnect two different nodes. The MPEG-7 Semantic *DS* can be seen as directed graph; whereas the nodes are the vertices and the relations are the directed edges. The graph is not necessarily connected, as relations are not mandatory. As all the nodes and relations are identified by descriptors, a semantic description is a labeled graph, whereas the MPEG-7 descriptors are the labels for the edges and vertices. Screenshots of visual representations are given in figure 2.

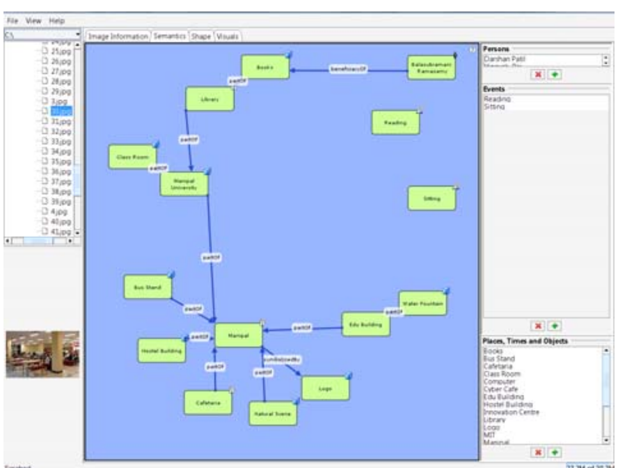

Fig. 2: Example of the representation of a MPEG-7 based semantic description.

For the sake of simplicity two nodes cannot be connected through two different relations and a relation cannot have one single node as start and end node. In graphs based on Semantic *DS* no two nodes can have the same label (the same descriptor), so the node labels are unique. Each directed edge can be inverted as there exists an inverse of each MPEG-7 based semantic relation.

VI. The Term Space For The MPEG-7 Semantic DS

Based on the identified constraints of a directed labeled graph with unique node labels and edges, that can be inverted, an efficient retrieval strategy can be designed as follows: The idea of fast retrieval of graph based structures is not new, as the contribution of Simmons in 1996 [Simmons1966] shows. Common retrieval techniques for graphs are the usage of metrics like the maximum common sub-graph metric, or the graph edit distance. A straight forward implementation using this distance measures results in search time *O(n),* whereas *n* defines the number of graphs in the database. Please note that the distance or similarity calculation between two graphs is *NP*-hard in respect to the number of nodes and edges of the graphs to compare [Valiente2002]. Another approach is the filtering of the database with a fast (less than linear search time) algorithm and the ranking of the results with a slower metric which is described in chapter 12 in [Baeza-Yates1999]. This method has been successfully used for graphs e.g. in [Fonseca2004] for clipart retrieval by using graph eigenvalues as filters like in [Shokoufandeh1999]. A more promising approach for MPEG-7 semantic *DS*, if the usage of an existing text search engine is constraint, is the usage of a path index [Shasha2002]. A path index allows the fast retrieval of graphs based on paths (sequences of nodes connected by edges, whereas the number of edges defines the length of the path) of different lengths extracted from the graphs. The extracted paths can be interpreted as index terms for a graph.

The graph can be expressed using all paths, which are sequences of nodes interconnected by edges, of chosen length. Paths of length 0 are the nodes themselves while paths of length 1 are triples as used in RDF. Paths of length 0 and length 1 have unique string representations for MPEG-7 based semantic descriptions as shown in [Lux2005]. To allow the usage of wildcard nodes at least the paths of length 2 have to be used, for which a unique string representation can be defined as shown below. The graph can be stored using the paths of length 0, 1 and 2 as index terms. Using a query graph all paths of the query graph are extracted and used as search terms. The ranking is done by TF*IDF on the index terms, which are the paths of the graphs.

VII. Implementation

Based on an open source retrieval engine *(Lucene),* an index for node descriptors has been implemented in NWCBIR along with the string representations of paths of length 0 and 1. As the query graph can consist of query strings for the node values, query expansion based on the node descriptors is used as described in [Lux2005]. All path representations are constructed from node IDs, which identify a unique node descriptor in the index, and relation names or wildcards for nodes or relations. For the usage for terms within the retrieval engine *(Lucene),* the path representations were adopted: all white spaces were replaced by '_' and all paths start with a leading '_'. The leading '_' allows the usage of wildcards at the start of a path expression.

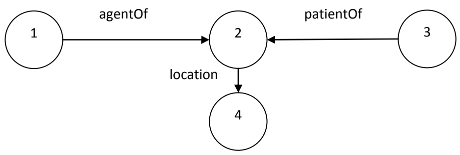

Fig. 2: Example of the representation of a MPEG-7 based semantic description.

For the graph given in figure 3 the terms for paths of length 0 and 1 are given in the following table.

Table 1: Extracted path terms of length 0 and 1 from graph shown in figure 3.

Term	*Path length*
_1	0
_2	0
_3	0
_4	0
_agentOf_1_2	1
_locationOf_4_2*	1
_patientOf_3_2	1

For the creation of terms from paths of length 2, following method has been introduced. The input of the method is either a graph representing a semantic *DS* or a query graph. In a first step all paths of length 2 are extracted from the graph [Valiente2002]. For each of these extracted paths the unique string representation has to be created as follows:

1. Compare the start node of the path with the end node of the path.
2. If the start node is bigger than the end node reverse the path:
   a. Switch end and start node.
   b. Switch and invert first and second relation.
3. Create string in order: start node – first relation – middle node – second relation – end node with '_' as separator.
4. Prepend '_' to the string.

This results in the table 2 as shown below.

Table 2: This table shows all available extracted path terms of length 2 from the graph shown in figure 3.

Term	*Path length*
_1_agentOf_2_patient_3	2
_1_agentOf_2_location_4	2
_3_patientOf_2_location_4	2

All these above shown terms are used to index the semantic description with *Lucene*; all terms are used as *Lucene* tokens without stemming or other pre-processing. For queries the terms are constructed in a similar manner with one exception: Wildcards for nodes and relations can be used. For relations the adoption is straightforward: As *Lucene* supports wildcard queries for a wildcard relation the string '*' is inserted instead of the relation name, e.g. _*_*1_2* instead of _*agentOf_1_2*. To support undirected wildcard relations two relation query terms are constructed and combined with a Boolean OR, like *(_*_1_2 OR _*_2_1)*. For paths of length 2 only the '*' is inserted instead of the relation name as the order of the path only depends on the start and end node.

For nodes in paths of length 0 the query string is omitted. For paths of length 1 and middle nodes in paths of length 2 the node ID is replaced with a '*'. For start and end nodes in paths of length 2 a Boolean query clause has to be constructed as the order of the start and end node cannot be used to identify the term representation, e.g. *(_*_patientOf_2_location_4 OR 4_locationOf_2_patient_*)*. Note that the relations have to be inverted in this case.
A simple example for a wildcard query would be: "Find all semantic descriptions where Balasubramani is doing something at the SMU". In the first step possible candidates for nodes are identified to construct the query graphs. Assuming that for Balasubramani the node with ID 28 has been found, while for SMU the node with ID 93 has been found, the query graph would look like *"[28] [93] [*] [agentOf 1 3] [locationOf 3 2]"*. The numbers within the relations reference the node using their position in the node list. Such a query would result in a query like *"_28_93_agentOf_28_*_locationOf_*_93_28_agentOf_*_locationOf_93"*.

VIII. Retrieval Mechanism for Semantic Descriptions

This section introduces our retrieval model, which is motivated by providing a fuzzy retrieval mechanism for semantic descriptions and an appropriate ranking scheme, including support for wildcards. As there are already well functioning and well tested tools for text retrieval available one major constraint is that we want to focus on the usage of existing text retrieval tools. All of the used techniques should find their source in existing text retrieval techniques to allow the usage of existing tools if possible to rely on their speed and precision.

For the retrieval process of MPEG-7 based semantic descriptions we can assume that, without loss of generality, a set of image exists, where each image is annotated with a semantic description. Thus, our goal is to retrieve a set of semantic graphs best matching a given input graph. In the following section nodes (or vertices) of the graph are denoted as semantic objects and edges of the graph are denoted as semantic relations. Our model is described in three parts, whereas the first part explains the indexing of semantic objects and semantic descriptions, the second part states on the retrieval process and the third part introduces the ranking method.

IX. Indexing of Semantic Descriptions

The first step is creating a data structure for accessing nodes *N* of the graph. As in text retrieval we are using an inverted index as data structure, which holds all semantic objects and offers good possibilities for speeding up the retrieval process as a whole.

In general for every unique semantic object in all semantic descriptions a unique identifier is assigned and an index entry is created, which means the text describing a semantic object is indexed using text retrieval methods. Note that, multiple semantic descriptions can share the same semantic objects. In this case each shared semantic object is treated as one object and obtains the same unique ID within the different semantic descriptions. Figure 4 shows in detail the implementation of the indexing process.

For example if the description shown in figure 5 are processed three different semantic objects with three different IDs are extracted: (i) Balasubramani (Semantic agent, part of description of image A and B, ID: 1), (ii) Talking (Semantic event, part of description of image A, ID: 2) and (iii) Listening (Semantic event, part of description of image B, ID: 3).

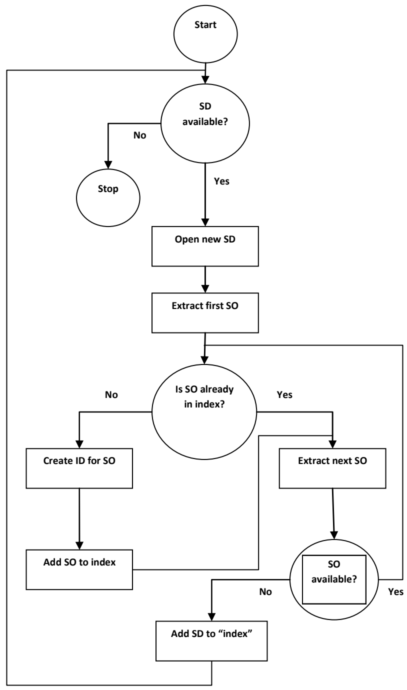

Fig. 4: Flow diagram showing the process of "indexing" semantic descriptions (SD) and semantic objects (SO).

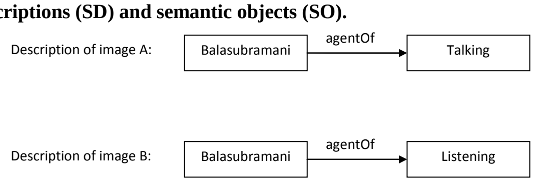

Fig. 5: Examples for semantic descriptions of two different images

After indexing and assigning unique IDs to semantic objects, semantic descriptions are indexed using a string representation. Given the semantic descriptions in figure 5, the string representation of image A is: *[1] [2] [agentOf 1 2]*, for image B the description is *[1] [3] [agentOf 1 3]*. In the first part of the string representation all available semantic objects (vertices) of the semantic description (graph) represented by their ID in square brackets are defined in numerically ascending order. The second part consists of all available semantic relations (edges) in lexicographically ascending order. Each semantic relation is defined in square brackets, whereas the name of the relation is followed by the ID of its source and the ID of its target. Note that the number of possible MPEG-7 based semantic relation types is already limited by the standard itself. Therefore relations do not get unique IDs but are referenced by their names. All possible semantic relations in MPEG-7 are directed edges but have an inverse relation. Based on this fact relations are re-inverted if their inverse relation is used in a graph. It can be seen that the string representation is unique.

X. Retrieval of Semantic Descriptions

Given the fact that above described indices and representations exists, a retrieval mechanism can be implemented as follows:

1. A user provides a query string for each of the k semantic objects he wants to search for and interconnects the nodes with relations. Each query string leads to a node query, q_1 to q_k, which are used to query the inverted index of semantic objects, as described earlier.
2. The retrieval engine searches for a set of available matching node IDs L_{q1} to L_{qk} for each node query q_1 to q_k, sorted by relevance of the matches. The relevance returned for each relevant node is in (0, 1], whereas a relevance of 1 indicates an optimal match. The relevance is obtained from using standard text relevance methods (e.g. vector space model).
3. Based on the sets of matching nodes for each node query the original query is expanded to $|L_{q1}|*|L_{q2}|*...*|L_{qk}|$ queries, for which the node IDs and the relevance of the nodes are available. This means that every node returned by q_i is combined with every node returned by q_j having i ≠ j. Given the semantic relations of the user queries consisting of semantic descriptions can be created.
4. For each of the above $|L_{q1}|*|L_{q2}|*...*|L_{qk}|$ queries the number of matching documents is found through a search in the string representations of the graphs with regular expressions. A relevance value for each matching document is calculated based on the formula presented in the next section.
5. All resulting sets from step 4 are merged in one result set, whereas for documents which are in more than one set, a higher relevance value is assigned.

XI. Relevance Calculation

Taking one specific expanded query q with node set $N^q = \{ n_1^q, n_2^q, ..., n_k^q \} \neq \emptyset$ and relation set $R^q = \{ r_1^q, r_2^q, ... r_l^q \}$ and one specific matching semantic description d resulting from the search in step 4 with node set $N^d = \{ n_1^d, n_2^d, ..., n_r^d \}$ and relation set $R^d = \{ r_1^d, r_2^d, ... r_s^d \}$ with $k, l, r, s \in N \cup \{0\}$. The relevance $r \in (0, 1]$ based on the query nodes relevance values $r(n_1^q), r(n_2^q), ..., r(n_k^q) \in (0, 1]$ is defined by:

$$r = \frac{\min(|N^q| + |R^q|, |N^d| + |R^d|)}{\max(|N^q| + |R^q|, |N^d| + |R^d|)} \cdot \prod_{i=1}^{|N^q|} r(n_i^q) \qquad (1)$$

The calculated relevance takes the relevance of nodes, which result from the query expansion, into account. The relevance value is in the interval (0, 1] because all node relevance values are in (0, 1] and the fraction has to be in (0, 1] because the numerator is smaller or of equal size compared to the denominator. Note that all irrelevant nodes are discarded in step 2 by discarding all nodes with relevance below a specific threshold which leads to a minimum relevance above zero. The relevance of semantic relations is not taken into account as the relations in the query are only matched with relations in the database following the Boolean model, not supporting a partial or fuzzy match.

To express the meaning of the relevance formula in words: The more relevant the nodes of the query expansion are, the more relevant is the matching semantic description. Additionally the smaller the difference in the number of components (nodes and edges) of the query and description graph is, the more relevant is the matching semantic description.

XII. Implementation Details

The above described method was implemented in NWCBIR. NWCBIR uses the *Jakarta Lucene* search engine *[Lucene],* which allows the creation of an inverted index of nodes. The string representations of semantic descriptions are stored in a flat file.

For query formulation a simple query language is used which can be described as follows: All nodes are defined in square brackets, inside these square brackets all features of *Lucene* like fuzzy matching or field matching can be used. Following these node queries the relations are defined using the name of the relation as defined in the MPEG-7 standard followed by the source of the relation and the target of the relation identified by the position in the list of node queries. Following *BNF* expression defines the supported queries:

Query ::= NodeQuery {NodeQuery} {RelationQuery}
NodeQuery ::= "[" NodeQueryString "] "
NodeQueryString ::= (Clause) *
Clause ::= ["+", "-"] [<Term> ":"] (<Term> | "(" NodeQueryString ")")
RelationQuery ::= <MPEG-7_Relation> <Number> <Number>

From each of the expanded queries a regular expression for searching in the file of semantic descriptions is created and executed on each semantic description in its string representation. If the regular expression matches the string, the associated documents are put into the result set and the relevance of the documents is calculated. Finally the result sets are merged following above described parameters and the sorted set of results is presented to the user.

XIII. User Interface

The component of most interest is the panel offering a search mechanism for searching semantic descriptions.

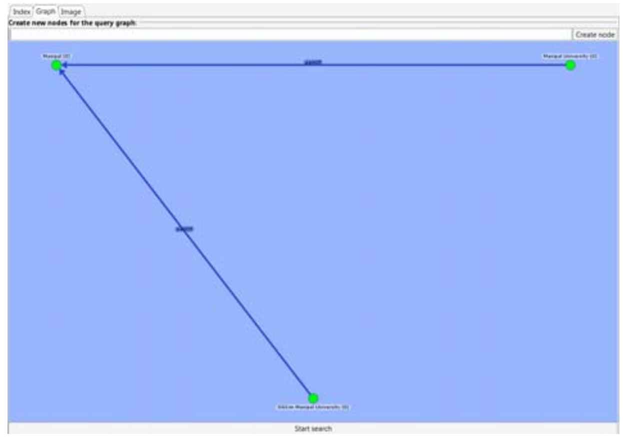

Fig. 6: Starting a semantic search using a graph as input. Three objects and two relations are defined.

This component allows the user to define a graph with minimum one to maximum three nodes and two possible relations. An asterisk is used as wildcard. A search graph which only contains one node with a word defining this node will return each MPEG-7 document wherein a semantic object containing the specified word is found. If two or three nodes and one or two relations are used to define the search graph, the repository of MPEG-7 documents is filtered by the terms defined as objects or relations. If, for example, the graph in figure 7 below is used for search, all documents which contain semantic objects, which contain the terms "Manipal", "SMU" and "MU", and a semantic relation containing the term "partOf" are taken from the repository and checked if there is also a structural match with the given graph.

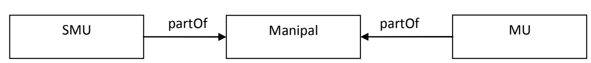

Fig. 7: Starting a semantic search using a graph as input

Fig. 8: Possible search graphs, which are supported in this prototype, are sub graphs of the maximum graph shown in figure 7

The retrieval mechanism follows modular system architecture, an *XPath* statement is given to a class implementing the interface *RetrievalEngine* and the results are received as list of HTML documents, which can be visualized using standard Java Swing components. The only retrieval engine implemented yet is the "*FileSystemRetrievalEngine*", which collects all MPEG-7 documents from a specified directory and its sub-directories and executes the given *XPath* statement. If a matching document is found it is transformed into HTML, using *XSLT*. This HTML result visualization is added to a list of results, which is ordered by relevance. Relevance is calculated using the number of nodes matching the *XPath* statement used as input. Another retrieval engine implementation would connect for instance to an *XML* database, which would result in a significant speedup executing the *XPath* statements.

In case of a content based image search each MPEG-7 document has to be loaded and the required descriptor is located using *XPath*. This descriptor has to be compared to the sample descriptor used as search parameter to calculate relevance. These results are put into a list ordered ascending by relevance, though a relevance of zero would show an exact match. Using a database the comparison of the descriptors has to be implemented on database side like a stored procedure, a server object or a similar mechanism, because of speed issues.

XIV. Results

A common problem with retrieval of *XML* documents is the speed, although Oracle and other big players in creating databases are already working on a possible solution. An also well-known fact is the insufficiency of *XPath* as query language. The upcoming standard *XQuery* represents a possible solution. Nevertheless, most manufacturers and database vendors do not support it yet.

The computation of graph visualizations in a very semantic way with minimum crossings is also complicated in this context. For instance loading a semantic description and reading it does not prove as complicated, but arranging and visualizing the same graph can be quite tricky without generating a complete visual mess, which can only bring confusion to the user. Mostly a semantic description has a central element, take for example an image of a conversation between two persons. The central element is the conversation, therefore it should be placed in the center of the visualization, and the objects representing the persons taking part in the conversation should be placed around this element along with a place, a time and a context for the conversation. Basically a very similar effect can be achieved if visualization with a minimum number of edge crossings is calculated, because the central element takes part in most edges and therefore it is placed in the middle.

Another problem is that different users produce different descriptions for the same media instance. Also MPEG-7 defines an inverse for each semantic relation; as a result a user can choose either a relation or its inverse to create the description. Therefore the retrieval mechanism must take care of these differences, like combined searching for relations and inverse relations and computing a similarity between semantic objects to identify objects which define the same person, thing or state but differ in the way they are described.

Finally the retrieval using a semantic graph as search parameter is not only a simple graph to graph matching based on a simple description standard but has to mind some parameters set from the MPEG-7 standard. In addition to the above mentioned inverse relations, MPEG-7 allows to integrate objects by reference. This means that the objects are used in the graph, but they are not fully described inside the descriptor itself, but are only referencing the object, which is defined in another part of the document or even in another document.

Although MPEG-7 defines similarity measurement for low-level content based descriptors it fails to define those measurement methods for calculating the similarity of two semantic graphs, so a generalized method has to be found and proposed.

We have carried out experiment to compare the recall and precision for retrieving a given set of semantic-linked image networks under the same set of query conditions. Each network contains thirty image nodes.

Figure 9 compares the change of precision and recall with the change of the number of semantic links. Figure 10 shows the recall and precision change with the number of types of semantic links.

We can see that the retrieval efficiency depends not only on the number of semantic links but also the types of the semantic links included in a semantic-linked network. We are carrying out experiments with larger scale and random samples to verify this phenomenon.

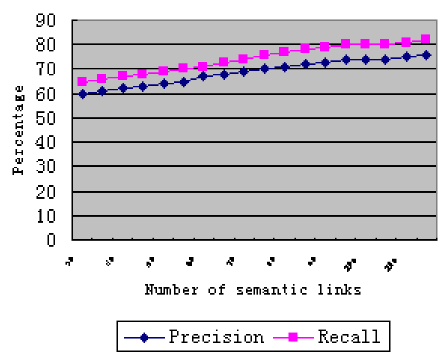

Figure 9: Recall and precision change with the number of semantic links.

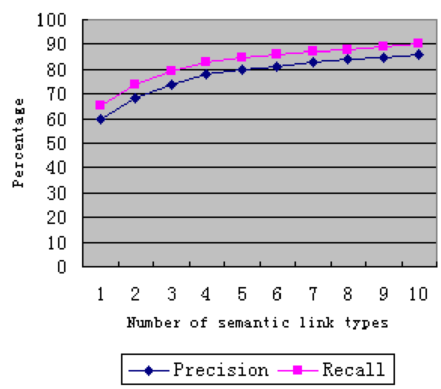

Figure 10: Recall and precision change with the number of types of the semantic links.

The underline premise of the proposed approach is that the image retrieval efficiency depends on the providers' semantic description on the provided images. If no semantic links are established, the proposed approach becomes traditional text-based or content-based approaches. The hyperlink-based approach also depends on the pre-established hyperlinks.

XV. Conclusion

The application areas most likely to benefit from the adoption of CBIR are those where level 1 technique can be directly applied. Users and managers of image collections need to be aware of the capabilities of CBIR technology, and to be capable of making informed decisions about adoption. Specifically:

- Managers of image collections in specialist areas such as fingerprints or trademark images, involving image matching by appearance, should be encouraged to investigate possible adoption of CBIR technology in the near future.
- Managers of video libraries should certainly investigate the possibility of using one of the proprietary video asset management packages.
- Managers of general-purpose image collections such as art galleries or photographic libraries should be encouraged to keep a watching brief on developments in CBIR, through articles in the specialist press and conferences relating to image retrieval particularly to hybrid text/image feature indexing and cross-media retrieval.
- Software developers or information providers with products designed to handle images, which currently lack CBIR capabilities, also need to make informed decisions about whether CBIR would add value to their products.

In conclusion, CBIR is clearly a technology with potential. The next five to ten years will reveal whether this potential can be turned into solid achievement. Our view is that at present the omen is favorable.

References

1. Semantics of Multimedia in MPEG-7, Benitez Ana B. Et al, 2002, International Conference on Image Processing 2002, Proceedings Volume: 1, Page(s): 137-140
2. Image Description and Retrieval Using MPEG-7 Shape Descriptors, Carla Zibreira, Fernando Pereira, ECDL 2000, Proceedings, pp. 332-335
3. Everything You Want to Know About MPEG-7, F. Nack, A. Lindsay, Part 1, IEEE Multimedia, 6(3), July-September 1999, 65-77
4. Multimedia Content Description Interface – Part 5: Multimedia Description Schemes, MPEG, 23.10.2001
5. Proposal for Integration of DublinCore and MPEG-7, Hunter, Jane, October 2000
6. Bormans, Jan, Hill, Keith, "MPEG-21 Overview", Moving Picture Ex-pert Group MPEG, Shanghai, October 2002, URL: http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
7. Del Bimbo, Alberto, "Visual Information Retrieval", Morgan Kaufmann Publishers, 1999
8. Flickner, Myron, Sawhney, Harpreet, Niblack, Wayne, Ashley, Jonathan, Huang, Qian, Dom, Byron, Gorkani, Monika, Hafner, Jim, Lee, Denis, Petkovic, Dragutin, Steele, David, Yanker, Peter, "Query by image and video content: The QBIC system", IEEE Computer, 28(9), pp. 23-32, September 1995
9. IBM Research, "MARVEL: MPEG-7 Multimedia Search Engine", http://www.research.ibm.com/marvel/, last visited: 25.11.2004
10. IBM Research, "VideoAnnEx Annotation Tool", http://www.research.ibm.com/VideoAnnEx/, last visited: 25.11.2004
11. Kosch, Harald, "Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21", CRC Press, November 2003
12. Lossau, Norbert, "Search Engine Technology and Digital Libraries - Libraries Need to Discover the Academic Internet", D-Lib Magazine, Vol. 10, Num. 6, June 2004
13. The Apache Software Foundation, "Jakarta Lucene", a Java based search engine, URL: http://jakarta.apache.org/lucene
14. Lux, Mathias, Klieber, Werner, Granitzer, Michael, "Caliph & Emir: Semantics in Multimedia Retrieval and Annotation", 19th International CODATA Conference, Berlin, Germany, November 2004
15. Martínez, José M., "MPEG-7 Overview", Moving Picture Expert Group MPEG, Pattaya, March 2003, URL: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
16. Mayer, Harald, Bailer, Werner, Neuschmied, Helmut, Haas, Werner, Lux, Mathias, Klieber, Werner, "Content-based video retrieval and summarization using MPEG-7", in Proceedings of Internet Imaging V, IS&T/SPIE 16th Annual Symposium, Electronic Imaging, San Jose, California USA, 2004
17. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R. "Content-based image retrieval at the end of the early years", Pattern Analysis and Ma-chine Intelligence, IEEE Transactions on, Vol. 22, No. 12, pp. 1349-1380, December 2000
18. Summann, Friedrich, Lossau, Norbert, "Search Engine Technology and Digital Libraries - Moving from Theory to Practice", D-Lib Magazine, Vol. 10, Num. 6, September 2004
19. Tsinaraki, Chrisa, Fatourou, Eleni, Christodoulakis, Stavros, "An Ontology-Driven Framework for the Management of Semantic Metadata Describing Audiovisual Information", in Proceedings 15th Conference on Advanced Information Systems Engineering CAiSE 2003, pp. 340-356, Springer, LNCS, 2003
20. Wactlar, Howard D., "Extracting and Visualizing Knowledge from Film and Video", in Proceedings 2nd International Conference on Knowledge Management I-KNOW '02, Journal of Universal Computer Science, July 2002