Supporting Image Search on the Web[1]

Giuseppe Amato

Email: G.Amato@iei.pi.cnr.it
IEI-CNR Via S. Maria 46
56126 Pisa Italy

Fausto Rabitti

Email: F.Rabitti@cnuce.cnr.it
CNUCE-CNR Via S. Maria 36
56126 Pisa Italy

Pasquale Savino

Email: P.Savino@iei.pi.cnr.it
IEI-CNR Via S. Maria 46
56126 Pisa Italy

Abstract

While pages on the Web contain more and more multimedia information, such as images, videos and audio, today search engines are mostly based on textual information. There is an emerging need of a new generation of search engines that try to exploit the full multimedia information present on the Web. The approach presented in this paper is based on a multimedia model intended to describe the various multimedia components, their structure and their relationships with a pre-defined taxonomy of concepts, in order to support the information retrieval process. A prototype of an image search engine, based on this approach, is presented as a first step in this direction, and results are discussed.

1 Introduction

The wide use of the World-Wide Web (WWW) across Internet is making of vital importance the problem of effective retrieval of WWW documents for casual as well as professional users. The documents on the WWW are becoming more and more complex and richer in their content. In these documents, the multimedia components, like image, video and audio, are playing an increasingly important role over the traditional text components. A large number of search engines index the documents on the WWW in order to provide support to their content-based retrieval. However, these systems, such as Altavista [1], Yahoo [25], HotBot [14] and Lycos [18], index the documents considering only their textual content. They periodically scour the Web, record the text on each page and through processes of automated analysis and/or (semi-)automated classification, condense the Web into compact indexes to support searching. The user, by entering query terms and/or by selecting subjects, uses these search engines to find more easily the desired Web documents.

Given the increasing importance of multimedia information – in particular images – there is the need to extend the capability of today WWW search engines in order to access documents according to their multimedia content. At the best of our knowledge two of the most promising prototype systems that address such a topic are Webseek [22,23] and Amore [20,4]. The first one combines physical features extracted from images and videos, and textual information extracted from path-names and alternate text to index the multimedia documents. It classifies documents in a subject taxonomy. Amore indexes Web pages using a standard approach for text as well as visual features extracted from images. Both of them support the retrieval of multimedia Web documents by using physical features such as color, shape, texture. The quality of retrieval is quite unsatisfactory: because these systems do not go beyond the use of pure physical visual properties of the images, represented in feature vectors, these systems suffer the same severe limitations of today general-purpose image retrieval systems, such as Virage and QBIC [5,11]. These systems consider a database of images as independent objects, without any semantic organization in the database or any semantic inter-relationships between database objects, and they result, from the user point-of-view in little more than toy systems [19]. These limitations are due to the generality of these systems with respect to the

[1] This research has been funded by the EC ESPRIT Long Term Research program, project no. 9141, HERMES (Foundations of High Performance Multimedia Information Management Systems).

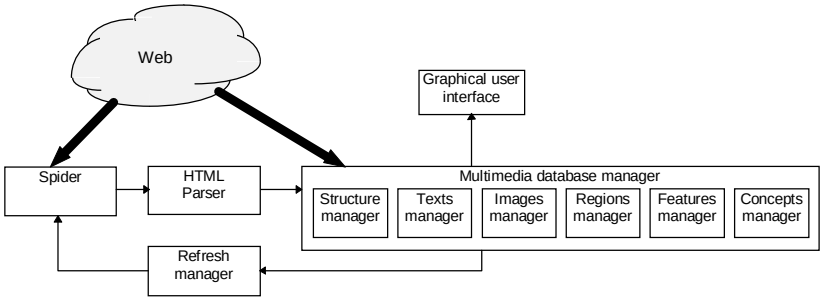


Figure 1: Search engine architecture

application domains and the limitations of general image feature extraction techniques.

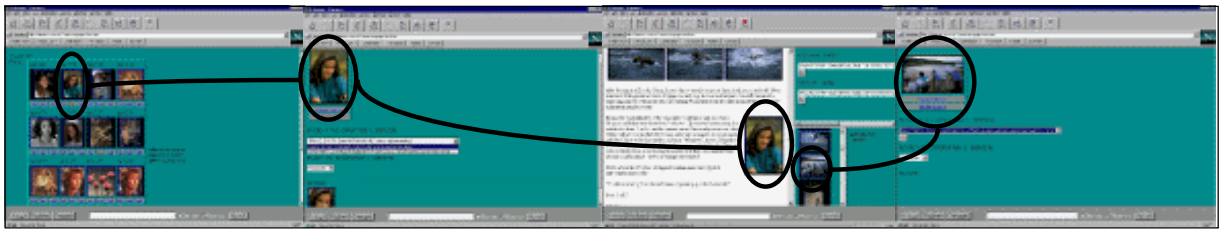
The approach proposed in this paper tries to overcome the limits of existing systems making use of the information about the "context" in which a multimedia object (i.e., an image) appears[15]: the information contained in the same Web document where the multimedia object is contained, the information carried by other Web documents pointing to it, the (possible) internal structuring of the multimedia object (i.e., components of an image). This approach permits to combine, during query processing, the result of efficient information retrieval techniques, working on text components, with less precise results of generalized feature based image search techniques, exploiting the inter/intra-relationships in WWW documents. To describe and make use of this potentially rich information, it is necessary to use a suitable model for the representation of multimedia WWW documents. We have adopted the HERMES multimedia model [2], that is particularly suited to describe their multimedia document content in order to support content-based retrieval. Multimedia objects are represented through features and concepts – concepts provide a description of the semantic content of the object – while the retrieval is based on the measure of similarity between the objects and the query. The power of this model derives from the possibility of defining, according to the specific application needs, what are the features used to represent the physical properties of the multimedia objects (how they can be "measured" and how two feature values can be compared) and how the concepts can be described and recognized [12].

In our approach, the model is being used to represent the structure and content of Web documents and specific concept taxonomies. We have implemented a prototype system that gathers documents from the Web and, after an analysis of its structure identifies the different types of data (e.g. text, images, audio, video) and "classifies" them. Textual information is used to extract the terms to be used for text retrieval purposes; images are analyzed in order to compute the values of the physical visual features and to derive the presence of one of the pre-defined concepts. Retrieval, performed through a visual interface, allows one to express restrictions on the textual information present in the documents, on the presence of images having a certain concept and to use images (or part of images) in order to retrieve documents containing other similar images. Furthermore, the information extracted from Web documents is used to support browsing between documents having similar content. Thus, the resulting prototype system has the functionality of existing systems based on visual features, as well as the possibility of supporting document retrieval based on concepts contained in the images; furthermore, it combines textual and image retrieval capabilities.

In the following, section 2 provides a brief overview of the system architecture and its functionality; section 3 contains a condensed description of the HERMES model; section 4 describes how the search engine has been built exploiting the HERMES model; conclusions illustrate open points for future research work.

2 Architecture of the image retrieval search engine

The prototype system that is presented in this paper addresses all phases of the retrieval of Web documents: it gathers documents from a pre-defined set of Web sites and analyses their content in order to derive the structure and to extract an appropriate description of their content. The existing version uses only text components and images, but it is intended that it can be extended to other multimedia data types. The extracted information is used to "classify" the documents in order to provide support to their content-based retrieval. It is useful to observe that a Web document may contain references to other Web documents, such as images, or other multimedia objects. They are classified independently but the relationship with other documents that refer them is retained. Traditional IR text retrieval techniques are applied on the text part of the documents. Images are processed, using today available technology, in order to extract their visual features which are used to support similarity-based image retrieval. A taxonomy of image categories has been created and the classification process automatically associates an image to one or more categories. Further information can be derived through the associations of image and text components (including URL names [22]). This process is described using the HERMES multimedia model into a suitable multimedia database schema: this is the key for exploiting this information during the retrieval process. Efficient retrieval is



a) b) c) d)

Figure 2: User interface

obtained through access structures which are specific of the different data types used in the multimedia database schema. For example, we can index words belonging to the text components and visual feature representatives for the image components. For image features we use the M-Tree access method [8], developed in the context of the HERMES project. The M-Tree access structure can be effectively used when the similarity of feature representatives is a metric (e.g., in case of image shape comparison) or when the image representatives belong to an high dimensional vector space (e.g. in case of color histograms).

The system architecture is sketched in Figure 1. The *Spider* grabs the content of a set of specified Web sites navigating all their internal pages. It is based on the gatherer of the Harvest Information Discovery and Access System [6]. The retrieved HTML pages are passed to the *HTML parser* which recognizes the hyper-links and distinguishes between references to other pages, references to images and references to in-line images. This information is given to the *multimedia database manager*. The *Refresh manager* periodically verifies that the Web URLs contained in the database are up-to-date and, when needed reconstructs and updates portions of the information about the pages of the Web sites. Finally, the *multimedia database manager*, according to the information passed by the HTML parser, updates the content of the database accordingly to the database schema design.

The multimedia database manager is composed of several internal components. The *structure manager* maintains the information about the hyper-links among the indexed Web documents. The *text manager* handles the HTML pages, each one represented by objects of the model. It co-operates with the structure manager to keep consistent the relations between the pages. The *image manager* temporarily downloads the referred images, in order to have the features extracted by the feature manager and to create a thumbnail of the images. The *region manager* allows to identify and represent relevant regions of the images (a region is any subpart of the image having a significance per-se). It materialises them when they are created in order to have the features extracted and to create a thumbnail. The *feature manager* is the component that manages the features defined for the Web documents. It can extract the features, it can compare feature values and it can perform feature based document retrieval. The *concept manager* allows one to deal with semantic concepts. It allows to define, update, customise concepts and to retrieve documents that match a given concept. In Section 4 we provide a more detailed description on how the concepts can be used.

The user may interact with the search engine using a graphical web interface. Retrieval is supported through text and image similarity. Text retrieval is based on standard IR techniques. Image (as well as region) retrieval can be performed by providing an example and retrieving all similar images. Similarity between images is computed by comparing the values of features as well as concepts that can be extracted from them. The retrieved images are presented in decreasing similarity values. They can be used as a starting point for successive similarity retrieval or, according to the information extracted from Web documents, to browse the database in order to retrieve similar documents.

An example of a query session is shown in Figure 2. We assume that the user has retrieved a set of images Figure 2a). By choosing one of the retrieved images, it is possible to access at the information related to the image itself Figure 2b), such as: the address of pages that point to it or that contain it in-line, the list of regions contained in the image. This information can be used to browse the images: for example, the user can inspect a page in which the image is contained in-line (Figure 2c). This page is displayed together with information about the images referred, the images in-line, the pages pointed and the pages that refer the page. Then, one of the referred images is displayed (Figure 2d). The process can continue by using this image as a starting point for a query or in order to access the page that contains this image in-line.

3 The Hermes model

In this section we will briefly describe the main characteristics of the HERMES multimedia data model [2] in order

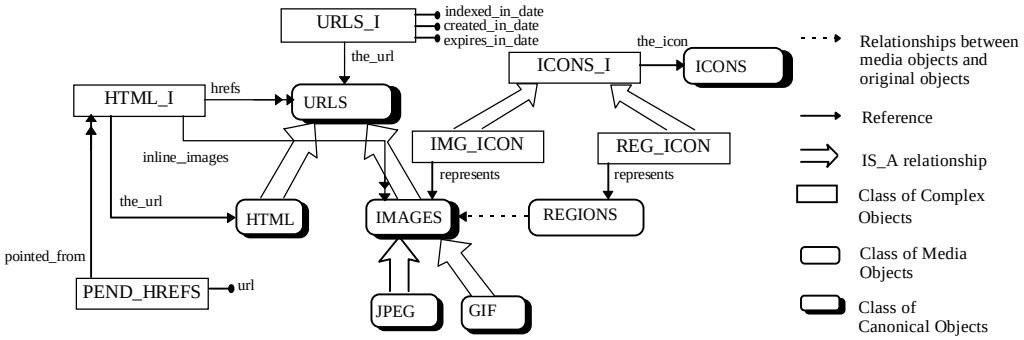


Figure 3: Description schema

to explain the functioning of the search engine prototype. Three different layers can be identified in the overall model: the Multimedia Storage Model layer (MSM), which allows one to define how multimedia information is stored in the database and how it can be accessed; the Multimedia Description Model layer (MDM), which allows one to identify relevant portions of multimedia data; and the Multimedia Interpretation Model layer (MIM), which allows one to specify the semantic interpretation of multimedia objects.

At the lowest level of representation, a *multimedia data* is any unstructured piece of information stored in the multimedia database - taken either from the real world or from other existing multimedia databases. These are the "real" pieces of multimedia data, which in the MSM, are identified by *raw objects*. Raw objects do not contain any specification regarding internal content and internal structure. They contain information about their physical encoding and the storage strategy used to store them.

One of the aims of interpreting a set of persistent multimedia data is to make explicit the structure and content present in the multimedia data in order to support their retrieval. The MDM is used to individuate the objects which should be interpreted. It allows one to specify the structure and the composition of all objects that the multimedia database manages. In the Multimedia Description Model, the unstructured content of a raw object can be conveniently structured by representing portions of it, and assembling such basic components into complex objects. Objects of the Multimedia Description Model are those that can be retrieved, manipulated, and delivered. The values of features, which are defined and used in the MIM, are calculated for the objects of the description model, and queries are performed by using these features and their semantic description as arguments.

To express queries based on the content of objects, the MIM allows the representation of the content of multimedia objects at two levels: the physical content is described by extracting features from multimedia streams, while a semantic description is obtained by associating object features to pre-defined concepts.

The HERMES model also consists of a query language for content base retrieval of multimedia data [3]. It is an SQL–like query language that allows one to express similarity retrieval and partial match queries on the content of multimedia objects.

3.1 The Multimedia Storage Model (MSM)

In the MSM any piece of multimedia information is represented by a *raw object* (RO) which does not contain any description about its semantic content. ROs contain the physical data of the objects, and information about their physical encoding and on the strategy used to store them - which means that it is possible to define the place the object is stored (on local disks, remotely) and the type of access that will be used to retrieve it (sequential, striped, etc.).

The storage strategy of an object is a quadruple *<strname, strtype, file_to_str, str_to_file>*. The name of the storage strategy is given in *strname*; examples are *local_file, local_database_object, striped, off-line, remote-url*, etc. *strtype* is the type of the data structure that allows one to represent and to access the physical content of a RO that uses this storage strategy. *file_to_str* and *str_to_file* are two functions which respectively store a file into a RO and get a RO.

When a new RO is created, its storage strategy and the source file should be specified. Then the content can be transparently managed regardless of the storage strategy adopted.

For example a RO may represent an image encoded into the GIF format and its storage strategy may indicate that the object can be fetched by following a particular web URL. Another RO may represent a video encoded into the MPEG format and its storage strategy specifies that it must be stored by using a striping algorithm. Both objects are accessed by object identifier, regardless the details that involve their corresponding storage strategies.

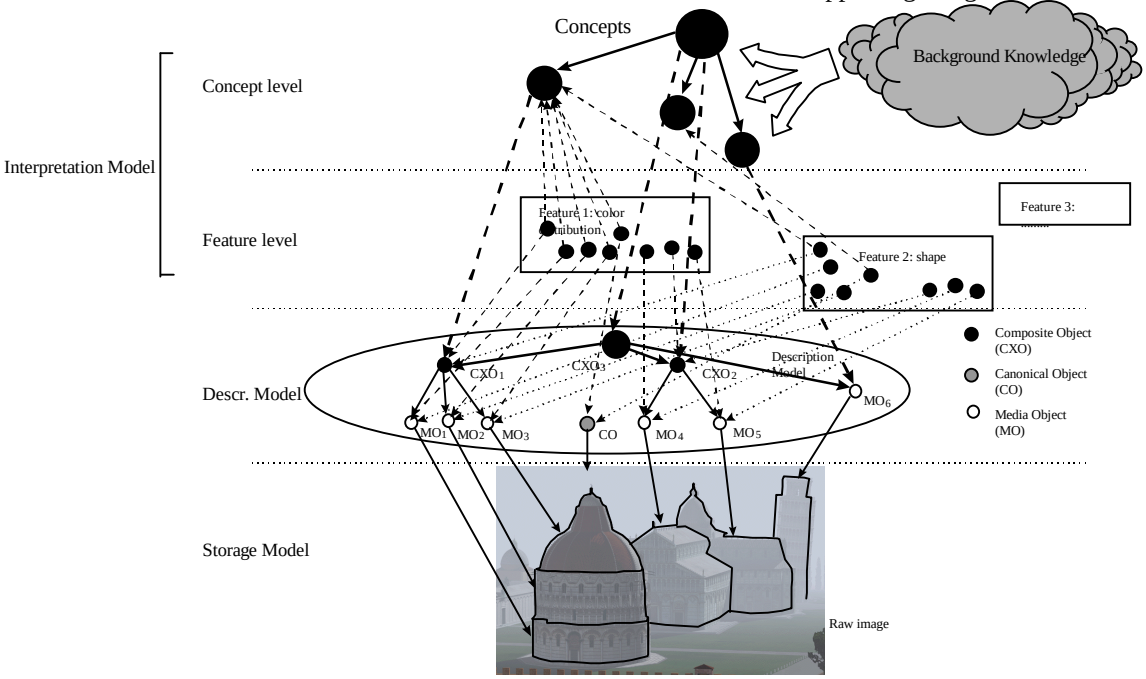


Figure 4: HERMES model: an example

3.2 The Multimedia Description Model (MDM)

At the MDM level, no assumption on the semantic content of the documents is made: the function of the MDM is to provide the mechanisms for defining and manipulating the structure of the information contained in ROs. The semantics is provided by the levels using the MDM, such as the MIM.

The MDM provides three types of objects: the *Canonical Objects* (COs), which represent the entire multimedia document; the *Media objects* (MOs), that represent relevant portions of COs; the C*omplex objects* (CXOs), which provide a way of aggregating COs and MOs (as well as CXOs themselves). For example an image or a video are represented by a CO while the region of an image that contains a person can be represented by an MO. An HTML page containing in-line images and embedded videos can be represented by a CXO that points to an CO that contains the HTML source, to a set of COs that contain the in-line images and to a set of COs that contain the embedded videos. Classes of CO, MO and CXO can be defined to classify different types of documents. The description model schema defined for the search engine prototype is given in Figure 3 and will be described in section 4.

3.3 The Multimedia Interpretation Model (MIM)

The MIM is used to represent the content of COs and MOs, at two levels: the *feature level* – using visual properties of the objects – and the *concept level* – according to the semantic content of the objects.

The *feature level* describes the content of objects of the MDM by measuring the values of some of their physical properties – the *features*: examples of features are *color distribution*, *shape*, *texture*, *motion vectors*, etc. A feature is described by the quintuple <*name*, *dclass*, *ftype*, *extrf*, *simf*> where *name* is the name of the feature, *dclass* is the name of a description model class that contains the objects which the feature can be extracted from, *extrf* is an *extraction function* that is a function used to calculate the value of the feature of an object belonging to *dclass*, and *sim* is a *similarity function* that is a function that compares feature values. The model permits one to define new features according to user and application needs defining the previous quintuple. The measure of similarity between features is used during document retrieval in order to measure the degree of matching between the query and the retrieved documents. Indeed, the process of retrieving multimedia documents is imprecise: the system does not retrieve the documents that *exactly* match the query – it ranks the documents according to their *similarity* with the query. Feature values are automatically extracted when description model object are created in the database. Objects are indexed using as keys their extracted feature values.

The *concept level* describes the semantic content of objects of the description model. It may describe and represent the conceptual entities contained in an object and the relations among entities. This is obtained using an object oriented model extended to cope with the issues of multimedia document description. *Concepts* are represented both by classes and objects. They have the traditional object-oriented behaviour and in addition they also

behave as *fuzzy sets*. In order to obtain that, they are associated with a set of *membership functions*, each related to a different class of the description model schema. These functions, given an object of the description model, return the degree of recognition of the associated concept in the object. Once a membership function has been defined, the associated concept can be automatically recognised inside all objects of the corresponding description model class. We have identified three types of strategies to implement a membership function: (a) an object of the description level can be classified by using a prototype of the concept; this prototype is either compared with the features extracted from the object in order to measure the degree of matching or the correspondence is performed by the user (this corresponds to manual classification); (b) the concept is recognised in the description level object by using some feature values; the degree of recognition of the concept is obtained by comparing these feature values with those extracted from the object of the description level to be classified; (c) the concept is recognised in the description level object because other concepts have already been recognised.

The concept model also behaves like a classical object data model so it even allows information to be represented that is not explicitly contained in multimedia documents. The information contained in a multimedia document is only a partial representation of the information present in the real world. For instance the name of a person or his date of birth cannot be inferred from a picture. This kind of information can be described at the conceptual level by adding the missing information to concepts. Relationships among concepts can be described at this level.

Figure 4 shows all levels of the model used to describe the interpretation of an image. The parts of the image that contain the tower, the Baptistery and the Duomo are considered as the most relevant and are represented at description level as MOs while the entire image is a CO. Feature defined at the feature level are extracted from objects of the description level. These features can be used to recognise concepts at the concept level.

3.4 The query language

The multimedia query language MMQL defined in the HERMES model is an SQL-like language extended with constructs that allow users to deal with multimedia data [3]. MMQL allows users to express range queries and nearest neighbour queries using weighted similarity predicates.

However efficient processing techniques for this kind of queries is still an open research issue. Traditional query processing and query optimisation techniques are not sufficient to cope with them.

For instance consider the following nearest neighbour query: *find the best k objects having (Shape sim 'round') and (Colour sim 'red')*. Each predicate somehow orders objects of the database correspondingly. How can this query be executed without scanning the entire database? Notice that the best *k* objects for the former predicate might be the worst *k* for the latter.

In [9] there is a proposal for solving this problem. Supposing that

- the *and* operator is defined as the standard *fuzzy and* operator using the *min* between the scores of the two predicates,
- the predicates are independent,

the suggested approach is to retrieve *k'* objects from each predicate where $k' = k^{1/m} N^{1-1/m}$. It is shown that retrieving *k'* objects insures that in the intersection of the results of the two predicates there are at least *k* objects. In the previous formula *k* is the number of nearest neighbours for the conjunction, *N* is the number of objects in the database, *m* is the number of conditions in the conjunction.

In [7] there is a proposal for a strategy to solve range queries in which multiple predicates occur. It uses a notion of *Selectivity Statistics* and an ad-hoc cost model to decide in which order the predicates should be processed. The selectivity statistics, given a range *r* and an object O_q used as the term of comparison, provide the expected amount of objects whose distance from O_q is smaller than *r*. The idea is that, if there is a conjunction of predicates, then the best strategy is to begin the computation with the predicate that returns the smallest amount of objects and checking the score of the obtained objects in the other predicates. However the problem of obtaining *Selectivity Statistics* is in most cases a complex task to deal with.

The problem of solving similarity condition with weights has been addressed in [10]. Provided that the grades g_i and the weights w_i for i= 1,2,...m are in the interval [0,1] and that the weights are normalised, the following formula has been proposed to compute scores for the conjunction of *m* atomic conditions with weights:

$$f_{(w_1,\ldots,w_m)}(g_1,\ldots,g_m) = \sum_{i}^{m-1}\left[i(w_i - w_{i+1}) \cdot \min\{g_1,\ldots,g_i\}\right] + m \cdot w_m \cdot \min\{g_1,\ldots,g_m\}$$

In order to compute scores for a disjunction of *m* atomic conditions, a similar formula can be used where the scoring function *min* is substituted by the function *max*:

$$f_{(w_1,\ldots,w_m)}(g_1,\ldots,g_m) = \sum_{i}^{m-1}\left[i(w_i - w_{i+1})\cdot \max\{g_1,\ldots,g_i\}\right] + m\cdot w_m \cdot \max\{g_1,\ldots,g_m\}$$

4 Indexing Web pages using the Hermes model

The prototype of the search engine indexes web pages from a pre-defined set of web sites. This set of web sites can be considered as a distributed remote "repository" of multimedia documents. The search engine uses a local database, which exploits the possibilities offered by the HERMES model, to represent the information contained in that remote repository. Each document, which is contained in any of the indexed web sites, is represented by a local object. The objects of the local database are classified and organised in relationships that reflect the original structure of the specified web sites.

Next subsections will describe the storage strategy used for web documents, the design of the MDM schema, the features used to index web documents, and the concepts have been defined in order to support their retrieval.

4.1 Storage strategy for web documents

The local database does not replicate the information contained into documents of the web sites. Instead, a new ad-hoc storage strategy *web_url* was defined, that allows a local RO to be a representative for a web document. The content of a RO that uses this storage strategy is actually contained in the corresponding remote document. Using the *web_url* storage strategy, remote documents can be classified just accessing local objects and using features and concepts as in a local multimedia repository. Their content is *automatically* downloaded when needed, for instance when features should be extracted. All the other operations, as for example the manipulation of the relationships among objects, require to operate on the ROs themselves.

To define the *web_url* storage strategy we had to define the *strtype* data type, the *file_to_str* and the *str_to_file* functions (see section 3.1 for details about the storage strategies). The chosen *strtype* is a record with three fields (*host*,*port*,*path*) that represent a web URL. The function *file_to_str* is a function that returns that triple given a web URL. The function *str_to_file* downloads the referred document and stores its content locally in the specified file[2], given a triple (*host*,*port*,*path*) and a file name.

4.2 Description model schema

An HTML page can be considered a structured multimedia document. It can contain text, images, videos, and audio; it may also contain references to other HTML pages and other multimedia documents. The MDM offers the mechanisms to represent, store and retrieve this kind of documents.

A web document is represented in the MDM by a Canonical Object (CO). Complex Objects (CXOs) are used to represent the information corresponding to the structure and to the relationships between web documents. Media Objects (MOs) are used to represent relevant components contained into web documents.

The current prototype handles text and images; it represents relevant regions inside images and keeps track of the overall hyper-link structure of an HTML page.

Figure 3 sketches the description model schema used in the prototype. Each web document is represented by a CO of the class URLS. The class URLS has two direct subclasses: the class HTML and the class IMAGES. The class HTML contains COs corresponding to web URLs referring HTML pages while the class IMAGES contains COs corresponding to web URLs that refer images. The class IMAGES has two direct subclasses as well: the classes JPEG and GIF.

In the HTML class we store COs that represent entire HTML pages. However they do not contain any information about the internal structure of a page as for instance the list of referred pages or the contained images. Therefore, the structure of each HTML page is explicitly stored by using CXOs of the class HTML_I (it stands for HTML pages Information). Each object of that class has a reference to the object of the class HTML that represents the actual HTML page, a set of references to objects of the class URLS that correspond to web URLs referred by the <A HREF> tag, and a set of references to objects of the class IMAGES that correspond to the in-line images of the page, i.e. those images referred by the <IMG> tag. Notice that the web URLs referred by the <A HREF> tag may be other HTML pages, images that are not in-line or other types of documents.

[2] Since the content of a RO may be required several times during a processing phase, a trivial implementation of the *str_to_file* function could download the same document repetitively. The *str_to_file* function uses a local cache in order to avoid this behaviour.

Using this structure, it is possible for instance, to express queries like “retrieve all pages that point to this page” or “retrieve all pages that contain or refer an image similar to this”.

The class URLS_I (it stands for URLs Information) contains generic information about a web URL. This information includes the date in which an URL was indexed, when it was originally created in the web site, when it may expire, in order to check it again and eventually to refresh the related information.

Images may contain several subjects. For instance, an image may contain a *person* on the left of a *tree* near a *house*. Relevant regions of an image [16,17], that is regions that contain relevant subjects, are represented by MOs of the class REGIONS. In the current prototype regions are defined manually, using a graphic tool, and consist of the bounding box of a subject. We plan to use some automatic segmentation algorithm to identify regions in images and to generalise the representation of regions by using polygons. The use of regions makes possible to express a query like “*retrieve all images in which there is a region similar to this image*”.

Thumbnails are used to temporarily visualise images and regions: downloading the original (eventually large) images from the original web sites may be too slow. Every remote image and region is associated with a thumbnail. Thumbnails are stored in the local database and are represented by COs of the class ICONS. The storage strategy chosen for ROs corresponding to thumbnails requires to store them in the local database as BLOBs. Objects of the class ICONS are associated with the corresponding objects of the class IMAGES or of the class REGIONS through the hierarchy of classes ICONS_I, IMG_ICON, REG_ICON.

During the indexing phase it is possible to find an HTML page that refers to other HTML pages that have not been inserted in the database. In that case it is not possible to set all references in the object that is going to be inserted in HTML_I, since the objects that have to be referred do not exist. Therefore the pending pages, that is the web URLs of the pages that are not yet in the database, are recorded in the class PEND_HREFS. When a new page is going to be inserted in the database, the system checks if the web URL is contained in PENDING_HREFS. In that case, after the new object of the class URLS is created, all objects of the class HTML_I, corresponding to the referring pages, will be updated adding the pointer to the new CO.

4.3 Features

The HERMES model, as we said in section 3.3, is not tied with a pre-definite set of features. Indeed, provided that feature extraction and feature similarity functions can be supplied, any new feature can be handled by the system depending on the application that should be supported.

In the prototype of the search engine we defined features for objects of the classes IMAGES, REGIONS and HTML.

The basic approach to search for images is to use visual features. The hypothesis is that if two images are similar, their visual features are similar too. Since matching features is simpler that matching images itself, similarity image search is actually obtained by using similarity feature search [24,21,12]. The features used to handle images in our prototype are *global colour*, *local colour*, *texture* and *structure*. The extraction and similarity functions used for these features are provided by the Virage package [5]. These features are defined both for the classes IMAGES and REGIONS. In addition to that, the feature *position* and the functions needed to express constraints on the positions of the regions, as for instance *left_to*, *right_to*, *above_to*, *below_to*, are defined for the class REGIONS.

We have also defined two features for the class HTML: the *title_keywords*, and the *body_keywords*. The similarity functions for these features are traditional text matching functions.

Queries like “retrieve all HTML pages in which there is an image whose global colour is similar to green and contains a region, whose structure is similar to the structure of this image, on the left of a region, whose local colour is similar to the local colour of that image” or “retrieve all images that are contained in an HTML page that is about the moon, whose structure is circular and whose global colour is bright grey” can be expressed exploiting the defined features.

The access method that we used, to index features and to search them by similarity, is the M-Tree [8]. M-Trees can use similarity functions that satisfy just the metric properties.

4.4 Conceptual schema

As described in section 3.3, in the HERMES multimedia model, concepts can be represented both by classes and objects and can be seen as fuzzy sets. Each concept is associated with a set of membership functions related to different classes of the description model schema. Concepts are recognised inside objects of the description model by using the membership functions associated with the concepts themselves. It is possible to define new concepts and to modify and customise the existing ones.

The fuzzy sets characterised by concepts can be represented either *statically*, by explicitly specifying the set of objects of a fuzzy set and their recognition degrees, or *dynamically*, by defining membership functions as a partial match queries that use other concepts and combinations of features. The first approach allows queries that involve

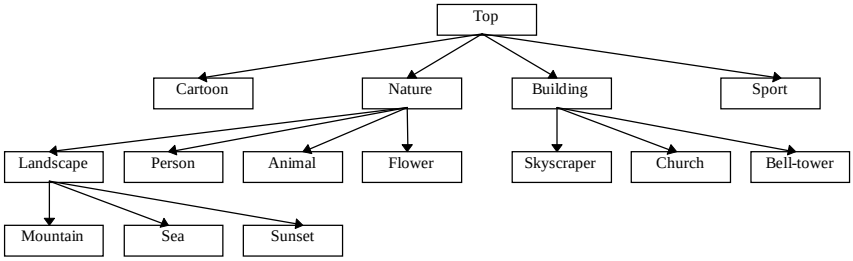


Figure 5: Concepts' taxonomy

concepts to be executed faster but does not allow users to customise and to adapt concepts to their needs. The second approach may be slower at query time, since testing if a document matches a concept actually corresponds to execute a complex similarity query. On the other hand, it allows users to modify the behaviour of the membership functions changing some parameter of the query.

In the prototype system we decided to give a dynamic behaviour to concepts represented by classes , and a static behaviour to concepts represented by objects. Let us consider the concept "person", represented by a class, as an example. Different users may intend different things by using this concept: men, woman, child, black, white, tall, fat, etc. Each of this different interpretations can be obtained customising the membership function of the concept. At the same time all users may agree on the concept "Bill Clinton" represented by an object of the class person. Therefore it is not needed to allow users to customise the concept "Bill Clinton". Furthermore, it is easier to automatically recognise a person in a picture than to distinguish Bill Clinton from all other people, so the membership function of the concept "person" can be defined as a query that use similarity between combination of features while the concept "Bill Clinton" can be defined explicitly specifying the objects in which there is "Bill Clinton".

The taxonomy for the concepts defined up to now is shown in Figure 5[3].

5 Conclusions

In this paper we presented an approach for retrieval of multimedia documents on the WWW. The prototype of a search engine system uses text and images in the retrieval process; in particular, images are retrieved using their visual features as well as semantic information. Similarity retrieval and browsing are supported.

Further research work will proceed in three main directions: (a) evaluation of system performance, (b) improvement of the classification process and (c) management of new types of data.

The first point will require the study of response times during gathering and classification and during the retrieval; it will also require an analysis of the quality of retrieval (effectiveness) at least through a comparison with respect to other approaches. Regarding the second point, we expect to better classify documents using the OCR technology for recognising text in images and encapsulating algorithms for automatic region extraction. The effectiveness of the system will be improved exploiting a query modification strategy that uses content-based relevance feedback and the interaction with the system will be enhanced by allowing users to submit visual queries by example and using a better strategy to visualise the results of queries. Finally, we plan to extend system capabilities by dealing with audio data and video data, in addition to images.

6 References

1. http://www.altavista.com

2. G. Amato, G. Mainetto, P. Savino. "An Approach to a Content-Based Retrieval of Multimedia Data". Multimedia Tools and Applications, 1998, Vol 7, No 1/2, pp. 9-36.

3. G. Amato, G. Mainetto, P. Savino. "A query language for similarity-based retrieval of multimedia data". In proceeding of the First East-European Symposium on Advances on Databases and Information Systems-ADBIS 97, St. Petersburg, September 2-5, 1997, pp. 196-203.

[3] The shown taxonomy should not be confused with the conceptual schema. In fact links between concepts in the taxonomy do not correspond to inheritance relationship between classes. The taxonomy in some sense represents the paths that the user may follow to browse existing concepts.

4. http://www.ccrl.neclab.com/amore

5. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain and C. F. Shu. "The Virage image search engine: An open framework for image management".In Proceedings of the SPIE 96, 1996

6. C. Bowman, P. Danzing, D. Hardy, U. Manber and M. Swartz, "The HARVEST Information Discovery and Access System", In proceedings of the Second International World Wide Web Conference, Chicago, Illinois, October 1994.

7. S. Chaudhuri and L. Gravano. "Optimizing Queries over Multimedia Repositories". In proceedings of the ACM SIGMOD 96, Montreal, Canada, 1996, pp. 91-102.

8. P. Ciaccia, M. Patella, P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces", In proceedings of. 23rd International Conference on Very Large Data Bases (VLDB'97), Athens, Greece, 1997.

9. R. Fagin. "Combining Fuzzy Information from Multiple Systems". In proceedings of the ACM PODS 96, Montreal, Canada, 1996, pp. 216-226.

10. R. Fagin and E.L. Wimmers "Incorporating User Preferences in Multimedia Queries". In proceedings of the International Conference on Database Theory. Delphi, Greece. Lecture Notes in Computer Science, Vol. 1186, Springer, 1997, ISBN 3-540-62222-5, pp. 247-261

11. M. Flickner et alt. "Query by Image and Video Content: the QBIC System", IEEE Computer, Vol. 28 No.9, September 1995

12. V. N. Gudivada and V.V. Raghavan "Content-Based Image Retrieval Systems: Guest Editors' Introduction", IEEE Computer, September 1995.

13. http://www.ced.tuc.gr/Research/HERMES.htm

14. http://www.hotbot.com

15. V. Harmandas, M. Sanderson, M.D. Dunlop. "Image retrieval by hipertext link". In proceedings of ACM SIGIR '97, 1997, pp. 296-303

16. Jose and Harper. "An integrated approach to image retrieval", The New Review of Document and Text Management, Vol 1, 1995, pp. 167-181.

17. Jose and Harper. "A retrieval mechanism for semi-structured photografic collections". In proceedings of DEXXA'97, Springer Verlag, 1997, pp. 276-292 (Lecture Notes in Computer Science No. 1308)

18. http://www.lycos.com

19. C. Meghini, F. Sebastiani, U. Straccia, "Modelling the Retrieval of Structured Documents containing Texts and Images", In proceedings of the First ECDL, Pisa (Italy) September 1997.

20. S. Mukherjea, K. Hirata and Y. Hara. "Towards a Multimedia World Wide Web Information Retrieval Engine" in Proceedings of the 6th WWW International Conference, S. Clara CA 6 - 11 May 1997

21. E. Petrakis and S. Orphanoudakis. "Methodology for the Representation, Indexing, and Retrieval of Images by Content", Image and Vision Computing, , Vol 11, No. 8, 1993.

22. J. R. Smith, and S. Chang. "Visually Searching the Web for Content" IEEE Multimedia, July-September 1997.

23. http://www.ctr.columbia.edu/webseek

24. J.Wu, A. Narasimhalu, B. Methre, C. Lam and Y. Gao. "CORE: A Content-based Retrieval Engine for Multimedia Information Systems", Multimedia Systems, Vol 3, No. 1. 1995.

25. http://www.yahoo.com