Generic Multimedia Database Architecture Based upon Semantic Libraries

Omara ABDUL HAMID, Muhammad ABDUL QADIR,
Nadeem IFTIKHAR, Mohib UR REHMAN, Mobin UDDIN AHMED,
Imran IHSAN
Center for Distributed and Semantic Computing, Mohammad Ali Jinnah University
Jinnah Avenue, Blue Area, Islamabad, Pakistan
e-mail: {omara,aqadir,nadeem,mohib,mobin,imran}@jinnah.edu.pk

Received: June 2006

Abstract. Semantic-based storage and retrieval of multimedia data requires accurate annotation of the data. Annotation can be done either manually or automatically. The retrieval performance of the manual annotation based approaches is quite good, as compared to approaches based on automatic annotation. However, manual annotation is time consuming and labor extensive. Therefore, it is quite difficult to apply this technique on huge volume of multimedia data. On the other hand, automatic annotation is commonly used to annotate the multimedia data based on low level features, which obviously lacks the semantic nature of the multimedia data. Yet, we have not come across with any such system which automatically annotate the multimedia data based on the extracted semantics accurately. In this paper, we have performed automatic annotation of the images by extracting their semantics (high level features) with the help of semantic libraries. Semantic libraries use semantic graphs. Each graph consists of related concepts along with their relationships. We have also demonstrated with the help of a case study that our proposed approach ensures an improvement in the semantic based retrieval of multimedia data.

Key words: automatic annotation, content-based image retrieval, multimedia database architecture, image abstract extraction, semantic libraries.

1. Introduction

Semantic extraction has always been a challenging problem in multimedia databases. Various architectures of multimedia databases have been developed in past but still there is a need to refine them to get the desired results of users' interest. Multimedia databases should able to store and extract semantics from images in a way, the user perceives them. Generic Multimedia Architecture (Rehman *et al.*, 2005) has introduced the new concept of "Semantic Libraries" to extract the semantics from the images so that the user can get successful query results based on semantics of images, e.g., "night scene in a village", "picture in a room" or "human playing with animal".

The retrieval of multimedia objects is facing a common problem known as semantic gap (Dunckley, 2003; Dorai *et al.*, 2002). The process of querying the multimedia objects is complex depending on not only what and how the information can be retrieved, but also on how this information can be merged logically to reduce the semantic gap. The three levels of complexity in retrieving the information defined in (Bartolini *et al.*, 2001) are; Level 1, 2 & 3 on the basis of Primitive, Logical and Abstract features respectively. But there exists a semantic gap among these three levels. Work can be found on reducing the semantic gap between Level 1 and Level 2 and still a lot more is to be done. But as far as Level 2 and Level 3 are concerned, very little has been done.

Our proposed work is an effort towards understanding the media rather than how media look like. Understanding media (image) is a complex phenomenon. It depends on a number of components. These components are objects and relationship amongst them, events taking place, scene etc. Each object exhibits its own environment, e.g., it is rare to find tigers and lions (wild animals) walking in the streets. We can find them in zoo or in jungle. So in the actual environment there is a strong relationship of tigers, ponds, trees, deep grassy area, bushes and weeds. The relationship amongst the objects and environment not only helps us in finding the abstract of media but sometimes if the objects are not clear, then by understanding the scene, we can depict the objects in it (Torralba and Sinha, 2001) or vice versa.

In our proposed semantic libraries, we are basically analyzing the media (image for the time being) through its components. For example if in an image we find tiger, deer and one of the components like tree or grassy area then it means that image is representing the abstract "tiger chasing deer in a jungle".

Simple semantics and abstract combines to form more complex semantics. For example an abstract of "picnic" has many underlying semantics like playing, running, horse riding etc. These sub-semantics further require different combinations of specific objects in different context. So when constructing libraries that contain this kind of complex semantic, care should be taken by making sure that information of all objects and their context is present in the libraries.

The main emphasis of our proposed architecture is to work like how human perceives the abstract of an image. The system pulls out the semantics of image along with low-level features like shape, color and texture etc. Semantics are extracted from the images by identifying the dominant classes, their combination, their context, and user's interest along with low-level features like colors, texture & shapes.

2. Related Work

In past many researchers developed multimedia database architectures. However, their aims were different. The aim of early multimedia database systems was to integrate the different databases (Tork Roth *et al.*, 1996). Some cover the specific domain (Dimitrova and Golshani, 1994; Olivia *et al.*, 2000; Wang *et al.*, 2005) while some put their efforts to make good architecture by improving query-processing technique (de Vries, 1998), feedback techniques (Iqbal and Aggarwal, 2003; Heesch *et al.*, 2003) and indexing (Kosch *et al.*, 2000; Kosch *et al.*, 2001) etc. There are some other projects that enhanced the already existing architecture of database management systems like Oracle9i by introducing plug-in components of Intermedia (Böszörményi *et al.*, 2001). Intermedia supports new and complex data types like ORDImage, ORDVideo, ORDAudio and ORDDoc. Instances of these object types include metadata, media data and methods of image, video, audio and document. We can assign weights to handle low-level features (colors, texture shape and location etc.) in multimedia data but nothing beyond that can be done. In this section we shall briefly discuss different multimedia architectures.

Popular general purpose image retrieval systems (Carson *et al.*, 2002; Mehrotra *et al.*, 1997; Faloutsos *et al.*, 1994) in the past decade views the query image and stored image in the multimedia databases as a collection of low-level features like color, texture and shape and then ranks the images in databases in proportion to these low-level features but the meanings (semantics) of the image were not considered by these systems. In the past, some research papers proposed the enhanced working of Ontology (Wiesman *et al.*, 2003; Dasiopoulou *et al.*, 2004; Hammiche *et al.*, 2004) but by handling ontology in the system means attaching semantics (keywords) with images at the time of storage manually and then mapping the user query on the same ontology for better results. It is not possible to assign keyword manually for large data repository.

The second popular technique to cater semantics is statistical classification method. In this method images are grouped into semantically meaningful categories using low-level visual features and categorizing the images into clusters by putting them in specific feature space and thus narrowing down the search, describing those images in the same clusters that have same semantics. This method has limitations in explaining the semantics among the different clusters (Sheikholeslami *et al.*, 2002; Vailaya *et al.*, 2001; Wang *et al.*, 2001; Chen *et al.*, 2003).

Another method is to rely on users for semantics by taking users' relevance-feedback (Cox *et al.*, 2000; Rui *et al.*, 1998). These systems have shown good results in specific domains but they may have added burden to users especially when he/she is interested in more complex semantics and especially in the situation where users' do not know how to provide feedback to the system. Though we can partially rely but it is not adequate to totally bank on relevance-feedback of users for extracting semantics.

Handling the semantics of the images by automatic classifying regions cannot extract the complex semantics between different objects and their context (Aghbari and Makinouchi, 2003). The design of semantic retrieval framework deals in handling the semantics on higher abstraction level using XML and MPEG-7 standards and for semantic abstraction they are relying on manual annotation. They proposed that in order to understand the "semantics", interfaces should support the user by "understanding" him or her. In our proposed system user understanding is being handled by users' profiles. The work proposed by Ying Liu, Dengsheng Zhang, Guojun Lu and Wei-Ying Ma (Liu *et al.*, 2005) in reducing the semantic gap between numerical image features and semantics in user mind by segmenting the image into color-texture homogeneous regions. Herald Kosch and Mario Döller analyzed the different popular multimedia architectures (Kosch *et al.*, 2005). They also analyzed the projects that mainly address needs of applications for richer semantic contexts that include MARS & Multimedia Data cartridge.

The main areas covered by MARS (Huang *et al.*, 1996) project were (1) multimedia content representation which includes extraction of low-level visual contents and textual annotation (2) Multimedia information retrieval which includes content-based multimedia retrieval techniques (3) multimedia feature indexing for efficiently retrieval based on similarity (4) multimedia database management for effectively and efficiently incorporate multimedia information into structured database processing. This project again depends on low-level features and manual annotation for semantics.

Multimedia Data Cartridge (MDC) (Döller and Kosch, 2003; Kosch *et al.*, 2005) is extension of Oracle 9i with added features of multimedia query language, query processing and optimization, indexing. This system is relying on MPEG-7 standard content-based retrieval but lacks in extracting the comprehensive semantics.

Multimedia database management system developed at the university of Twente, Netherlands (MIRROR system) (de Vries, 1998) improved the gap between user's high level information need and the search techniques in the databases by proposing the iterative query process using relevance feed-back. The basic aim of this project is to improve the search due to huge amount of data. The main emphasis is handling and investigating the problem of information overload. Efficiency and retrieval issue is based on query processing not semantics of image contents.

The Garlic Project (Tork Roth *et al.*, 1996) integrated the data that was resided on different databases. No mechanism was described on semantics extraction of multimedia data. Since it was amongst the earlier multimedia databases systems, so the main emphasis at that time was the integration of multimedia data.

Matt McKenzie (1996) analyzed the two common content-based systems of Virage engine and BLOBWORLD. Virage engine works by identifying the four main features like texture, color-type and location, shapes and structures. User can even assign weights to these features according to the importance of the selected option in their query. BLOBWORLD divides the image in regions (BLOBs), which are homogenous with respect to color and texture. User selects any of the blobs from the user interface to give his/her preference to the specific blobs. Then those images are retrieved from the database, which have best similarity with the blobs in the query image. Also like Virage engine user can choose the option to give importance to the some features like color, texture and location. Another similar system Windsurf (Ardizzoni *et al.*, 1999) applies the wavelet transform to extract the color and texture features and then divide the image into regions. Local features describe each region. So it can pose query that refer to local properties of image like find all blue regions enclosed by red regions. As far as semantic retrieval is concerned, all the above-mentioned systems work on the idea of how image look like, not what actually image is.

There are some other systems that are handling semantics by automatic annotation of images. These systems assign keywords automatically using different approaches. Some of the latest approaches of automatic image annotations are translation models (Duygulu *et al.*, 2002; Kang *et al.*, 2004), relevance language models (Jeon *et al.*, 2003; Lavrenko *et al.*, 2003) and classification approaches (Chang *et al.*, 2003; Li and Wang, 2003; Cusano *et al.*, 2004). In translation models words are annotated to image regions instead of whole image unlike classification approaches (Chang *et al.*, 2003; Li and Wang, 2003; Cusano *et al.*, 2004). The machine translation model (Duygulu *et al.*, 2002) translates the image to textual words. It first segments the image into regions. These regions are then grouped into clusters. Lastly similarity between the clusters and annotated words is learned. As described by (Kosch *et al.*, 2005), relevance language models take the test images and find the similar images in the training set and at the end combine the annotated words of similar training set image with those of test images. Results presented in (Kang *et al.*, 2004) have shown that performance of relevance models (Jeon *et al.*, 2003; Lavrenko *et al.*, 2003) are better than machine translation model but the two approaches proposed in (Kosch *et al.*, 2005) have shown better results than (Kuhn, 2003). All of the automatic image annotations cannot extract the full abstract or number of abstracts attached to single image although automatic image annotation is better than extracting the low-level features of image.

3. Proposed Multimedia Generic Architecture

The proposed framework of generic multimedia architecture with semantic libraries is shown in Fig. 1. The proposed architecture is flexible, so modules can be plugged-in or unplugged as required. The unified description of other components is as follows.

Feature extractors work on image by extracting the color and texture of shapes from the image at generic level and detailed object identification and image context is done at the domain specified level.

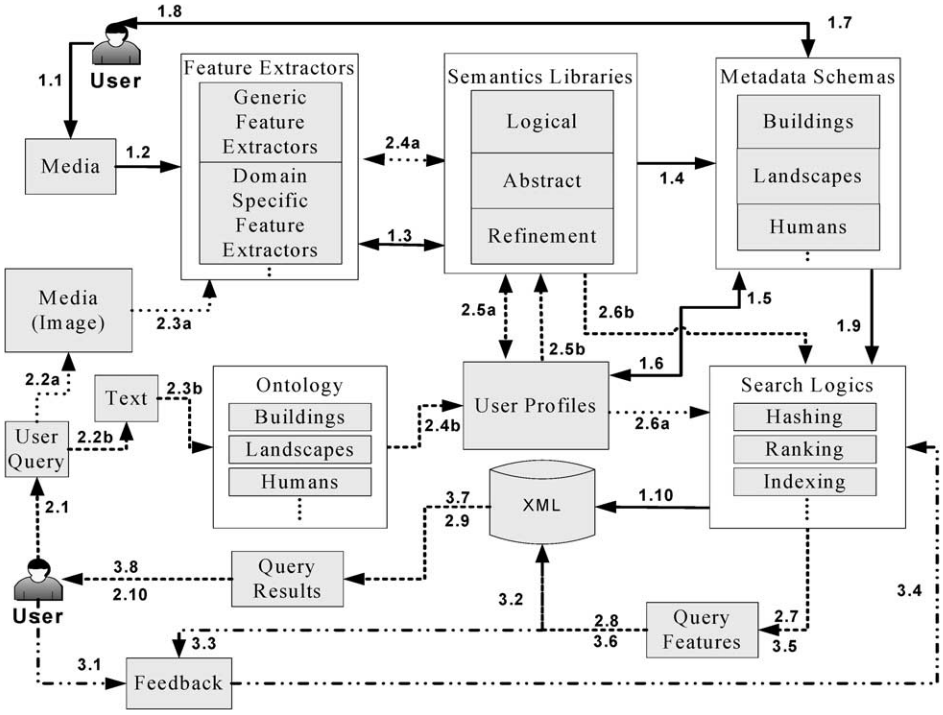


Fig. 1. Proposed generic multimedia architecture.

Semantic libraries work in accordance with the user perception. To add the semantic to the image the following points are considered:

At first, objects are detected in the given image. Since objects in one image collectively show some semantics and they have that semantic relationship amongst them.

Secondly, objects have strong relationship with their environment (Torralba and Sinha, 2001; Morgenstern and Heisele, 2003), so considering objects in the image and their environment to figure out the semantic. In this paper we are extracting not only context of the image for checking the environment of the image but also identify those objects that could not be identified and can be helpful in judging the semantics of image. For example the detected objects in an image are human, vehicle and building. The undetectable objects are near human and possess the shape like human then these unidentified objects are human. Since these objects are near object vehicle then possible semantic can be "pedestrians on road". So context of the image can give more accurate semantics of image.

Lastly, Users perception and interest in the image have great impact on the semantics of the image as different users see the same image with different angles so user's profile is helping us to figure out the semantics in which he/she interests. Our proposed semantics libraries work on the same three points mentioned above.

Search logic is a kind of search engine with all the possible components that are required for efficient search and retrieval of media file. Ontology contains a description of learning concepts that will add meaning to it. XML based database contains XML files that have three different sections. The first section contains the media file reference, second has metadata extracted using profiles and given by user according to standards applied and the last section has content metadata extracted automatically, using Feature Extractor.

User profiles play a key role in the collection of metadata and by making a query more related to what is required. The importance of user profile is as follows, i.e., making query more related to user, narrowing down the search by making semantic libraries to figure out the most probable semantics of image out of possible semantics in user context. We can find some good works related to the context-awareness for the improvement in media retrieval. User's thinking, searching and working with information explain his/her working context, which can increase information system retrieval performance (Rode and Hiemstra, 2005). User profiles will contain this user-context to help search logics in retrieval and ranking. User profiles can be user dependent modules and user-independent modules both modules have advantages and disadvantages (Rode and Hiemstra, 2005). Discussing different approaches and deciding which one is best, is outside the context of this paper. (Iftikhar *et al.*, 2005) proposed the multimedia retrieval architecture using the context-awareness. Adopting the presented modules like groups identifier and user dynamic context/history/future manager can be used to expand and modify the structure of user profiles as these profiles will contain user domain of interest, user search history and user web browser history. Feedback module will refine the results presented to the user.

Flow of the System

In this section we are first discussing **Media Submission for storage** purpose which has been shown in the Fig. 1.

User Input (1.1): User logon and submits any kind of supported media file to the system.

Feature Extraction (1.2): The Media, in this step will be handed over to the Generic Feature Extractors, where generic features of the media will be extracted, so that it can help Semantic Libraries in identifying the possible related classes of the input media.

Semantics Libraries (1.3): The generic extracted features of the media file are sent to the semantic libraries for related objects classes' identification. Semantic libraries are based on Semantic Graphs, so the related concept will be extracted from the semantic graph. On the bases of the output of Semantic Graph, the Semantic Libraries will request Domain Specific Feature Extractors for further specific feature extraction to confirm the higher-level concept/abstract. This is an iterative process and the Semantic Libraries may repeatedly request Domain Specific Feature Extractors for more feature extraction unless the ambiguity (if any) is removed. The graph may return more then one output. In such a case the weights of the output are important, and they are stored with the output to mention the relevance of the media with the concept/abstract. After processing the input, the Logical Library module sends a list of related objects & their classes back to the Feature Extractor Module for Domain specific feature Extraction.

Metadata Schemas (1.4): The extracted features, related object IDs and other information are sent to the Metadata Schemas Module for Metadata collection. This Module fills in the basic information (e.g., filename, Media type, etc), according to Schema.

Input to User Profiles (1.5): The Metadata Schema Module asks User Profile Module for possible input to metadata and sends the related Metadata Schema to it.

Output of User Profile (1.6): The User Profile Module fills in the information available in the user's profile and sends back to the Metadata Schemas Module.

Metadata to User (1.7): The Metadata Schemas Module sends all collected information according to the schemas to the user.

User Input to Metadata (1.8): The user fills in missing fields and modifies already filled by system (if required) and sends back to the Metadata Schemas Modules.

Input to Search Logics (1.9): The extracted features (Contents Metadata), Metadata and related object IDs are sent to the Search Logics Module for further processing, required for search and retrieval.

XML Storage (1.10): After applying all available techniques (e.g., Hashing, Ranking, Indexing) on the input, this module stores the output to the XML based database (an XML file).The system provides two types of **media search**. One is through the text query whereas second type is Query by Example, where the user provides a media as an example in order to find the appropriate results. Next we have explained how the system incorporates both queries.

User Query (2.1): User posts the query according to certain requirements.

Query Type (2.2): There can be of two types as shown below.

Media (2.2.a): Audio, Video or Image file

Text (2.2.b): Key words

Feature Extractors (2.3a): Generic features extractor will extract generic features from a media example for its classification.

Ontology (2.3b): Using the knowledge structure, Ontology expands user query.

Semantic Libraries (2.4a): On the basis of Generic Features, the Semantics Libraries identify classes of the features/Objects. After getting the possible classes of the features, the feature extractor performs Domain Specific Feature extraction, which is specific to the classes identified by the Semantics Libraries. This is an iterative process and works similar to Step 1.3.

User Profiles (2.4b): Expanded query is quantified on the basis of user's search profiles.

Refinement (2.5a): The expanded query is quantified on the basis of user's Search Profiles. The refinement module will consult the user profiles to find out his interests and try to remove ambiguity (if any) in the extracted features.

Classes Identification (2.5b): Domains are specified according to the expanded query's knowledge structure.

Search Logics (2.6): The refined query is qualified by search logics.

Query Features (2.7): Now the query is complete for searching the database.

XML Based Database (2.8): Query is posted to database for execution.

Query Results (2.9): Results are calculated and stored here.

Results Display (2.10): Results are displayed to the user in a proper format.

Feedback Module (3.1): After looking at the results user may give some kind of input to the system as a feedback. The user may select one or more output results as "related" and/or "not related".

XML Storage (3.2): The feed back module will ask for the related and not related properties/ features from the XML storage to perform the necessary operation.

Feed Back Module (3.3): The feed back module will perform Intersection operation on the features of all related images with the original query image and perform Minus operation on features of not related images. The output of this will be the some refined features ready to be searched again.

Search Logics (3.4): The refined features are qualified by search logics.

After this Step 3.5 to 3.8 will be same as Step 2.7 to 2.10.

4. Detailed Working of Components

4.1. *Feature Extractors*

Generic feature extractors will extract the dominant shapes and their colors and texture from the image (see Fig. 2 and Fig. 3) and pass these values to logic libraries.

At this level we require image segmentation methods that split the image into meaningful regions. Then each region will be a separate sub-image of the original image. There

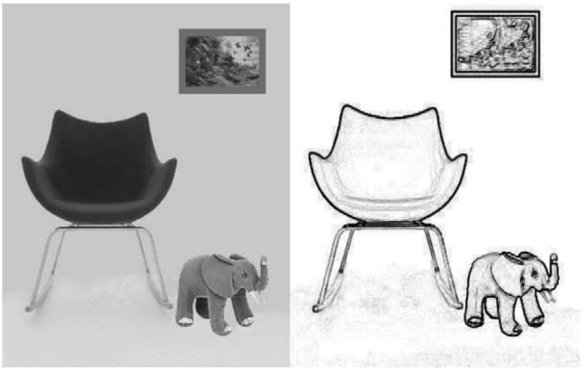


Fig. 2. Original image (on the left) and image segmentation (on the right).

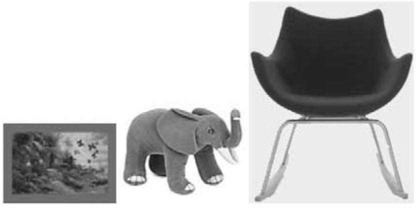


Fig. 3. Separated images.

are various image segmentation methods available to us at this stage. Image segmentation methods use brightness, color, texture and edge detection and some other features to segment the image into regions. Commonly known image segmentation methods are edge detection, histogram analysis and region growing (Pal and Pal, 1993). Histogram analysis is simple to implement and is suitable for simple images (Pal and Pal, 1993). Histogram is created using common features of texture, color or brightness etc. By analyzing the histogram threshold value is detected and regions are extracted based on this threshold value (Delon *et al.*, 2005). Edge-detection method involves finding edges of objects in images. However finding true edges of objects is a difficult task since noise in image, and different color and texture variation of the same object may greatly affect the results of edge detection methods.

To cater this problem and to refine the results some common techniques of edge image threshold, edge relaxation and border tracing are commonly known (Sonka *et al.*, 1998). Another commonly used method of image segmentation in many current CBIR systems is region growing with k-mean algorithm (Ardizzoni *et al.*, 1999; Chen and Wang, 2002). Yuan Jiang and Zhi-Hua Zhou (2004) have used a well-known clustering method SOM (Self Organization maps) Neural Networks for image segmentation. They have shown that the results of their proposed method is better than k-means based method (Ardizzoni *et al.*, 1999; Chen and Wang, 2002; Palus and Bogdanski, 2003) and method proposed in (Jiang *et al.*, 2003). Baris Sumengen performed robust image segmentation based on variation techniques (Sumengen, 2004). He evaluated the variation techniques on natural images. He claimed that amongst curve evaluation techniques, Edge flow has shown highly successful results on natural images. Any of the mentioned techniques that give better results in our scenario could be used.

Domain Specific Feature Extractors are specialized feature extractors, which will be used for the extraction of objects and their context. They are used iteratively for the extraction and refinement of semantics in the semantic library (see Fig. 1), i.e., to extract the context of an image. Suppose in an image, which is showing crowd on road, identified objects were human and car, foreground is road, then if the specialized feature extractors of car and human are used, it is the possibility that these feature extractors identify some more cars or humans that were not identified in the logical library. These specialized features extractors can work in various ways like Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio have shown that object can be detected by their components (Mohan *et al.*, 2001). So when using specialized feature extractor of human, this can extract other humans in the image by identifying only their parts. This can help out semantic library to extract the accurate results like “pedestrians crossing the road” or “road accident”.

4.2. *Semantic Libraries*

Semantics can be described as combination of two things; concepts and relationships among them. This information (concepts & relationship) can be viewed as a graph. Each base level vertex is a concept and an edge between these two vertexes represents their relationship. Many concept vertexes lead to the one higher-level vertex, this higher-level vertex represents a semantic for these concept vertexes and their relationship. Now this semantic may serve as a concept vertex for a level above this and so on. Such a definition of semantics and concepts can be viewed in (Kuhn, 2003; Giunchiglia *et al.*, 2003; Gomez-Perez and Corcho, 2002; Maedche and Staab, 2001). Semantic Libraries pull out the semantics of images by using the combination of semantics libraries and feature extractors. To demonstrate the example, we have assumed that a home contains drawing room, dining room, bedroom, kitchen and TV lounge (see Table 1). Each concept has other concepts within it as shown above in Table 1. We have defined a concept graph with contexts for different real life scenarios. Assume a concept of drawing room. In a drawing room, there are some concepts (scenarios of objects with names) that must be there and others are optional. It means we need to define compulsory containment and optional containment. The containments may be given some weights in order to see that how closely one concept relates in a bigger concept or within an abstract. Based upon those weights, a threshold weight can be defined in order to classify the collection of objects with its description.

For example, a drawing room should give a view of an enclosed area. With the view of enclosed area, if the scene has sofa or easy chairs, then it may be named as a drawing room.

The notations we have introduced in order to describe a concept contains many subconcepts, as follows:

Table 1

Containment of objects

Home	Drawing Room	Sofa Easy Chairs Enclosed Place Fancy lights Decoration piece Fire place Paintings Minor cabinet Plants Flowers
	Bed Room	Bed Fancy lights Decoration piece Fire place Paintings Minor cabinet Plants Sofa Flowers Easy Chair Enclosed Place
	Dinning Room	Fancy lights Decoration piece Fire place Chair Table Enclosed Place

1) **Containment Operator:** $\exists$, Usage: SemanticsName $\exists$ SemanticNames Left hand side is the master concept (its semantics description) and the right hand side represents one or more concepts / semantics descriptions contain in the master concept.
2) **Concept Qualifiers: compulsory concept [ ] , optional concept { }** Usage: [SemanticName] the concept enclosed in square brackets (semantic name) must be there and carry weight more than 50 out of 100 for the expression outside these brackets. {SemanticName} the concept enclosed in curly brackets (semantic name) should be there and carry weight less than 50 out of 100 for the expression outside these brackets. For example, a drawing room must contain a sofa. We will specify the containment operator, as **DrawingRoom** $\exists$[enclosed place]. A drawing room may contain a table, which can be written as **DrawingRoom** $\exists${table} .
3) **Concept relationship operators: AND Relation** $\wedge$, **OR relation** $\vee$, Usages: these are binary operators, which will show the relationship between concepts upon which these operators are applied. AND relation will be TRUE only when both the concepts are TRUE with reference to their weights. Otherwise this relation will result FALSE. OR relation will be TRUE if any one of the concepts is TRUE with reference to the given weight. With reference to the given weights if all the concepts are false, then OR operator will result FALSE.

We need to relate these concepts in one expression, which can be done by using the following relationship operators. These operators are optional because in some situations, the weights may be sufficient.

Now the drawing room concept can be written as **DrawingRoom** $\exists$[enclosed place] $\vee$$\{$table$\}$.

However, the concepts containment is not binary in real life. It may have some weights, which qualify the concept for its parent concept. For that, we have introduced the weight operator. Based upon the weights of the concepts enclosed in a concept, we will be able to attach the weights in order to qualify the chances of giving an abstract/semantics description.

In this paragraph we will explain how weight qualifier will work. Its description is as follows:

4) **Weight Qualifier: ( )** Usages: expression enclosed in parenthesis represents the weight of the concept immediately after this expression / left hand side of this expression.

If we assume that the 10% drawing rooms contain table and 70% drawing rooms have a visible enclosed place and sofa set. This can be represented **DrawingRoom** $\exists(70)$[enclosed place] $\vee(10)\{$table$\}$.

We may write this without OR operator, since there are weights with each sub-concept and the structure is not complex. The sub-concepts will be separated by comma in case we do not use AND or OR operators, like **DrawingRoom** $\exists(70)$[enclosed place], (10)$\{$table$\}$.

In the light of notations introduced and the concepts of a room, drawing room, bedroom and a home can be described as follows:

1) **DrawingRoom** $\exists$(x) [enclosed place], (x) [Sofa], (x) [Easy Chairs], (x) $\{$paintings$\}$, ...
2) **BedRoom** $\exists$(x) [(x)[enclosed place] $\wedge$ (x) [bed]] $\vee$ (x) $\{$table$\}$$\vee$ (x) $\{$sofa$\}$ $\vee$ (x) $\{$chairs$\}$...$\}$
3) **DiningRoom** $\exists$(x) [(x) [enclosed place] $\wedge$ (x) [table] $\wedge$ (x) [chairs]] $\vee$ (x) $\{$cutlery$\}$ ...$\{\ldots\}$. A home can be defined, by using the concept already defined, as **Home** $\exists$ (x) [enclosed place] $\vee$ (x) $\{$BedRoom$\}$ $\vee$ (x) $\{$DrawingRoom$\}$ $\vee$ (x) $\{$DiningRoom$\}$ $\vee\{\ldots\}$....

The above relations between different concepts can be shown as semantics graphs which are basically the semantic libraries as shown in the Fig. 4, where solid line shows the mandatory containments and broken line shows the optional containments with the weights shown on the lines. Broken line will carry weight less then 50, Solid lines will carry weight more than 50, and maximum weight, for each containment, is 100. The graphs can be traversed and the concepts can be extracted with the weights of the concept.

The concept graph of Fig. 4 can be used to define higher-level concepts as shown in Fig. 5.

Deduction Process to Extract Semantics from the Objects Contained in an Image

Assume the graph in Fig. 4, as the semantics library. The objects identified by the logical libraries in the first phase (by using generic feature extractors) are paintings, flowers,

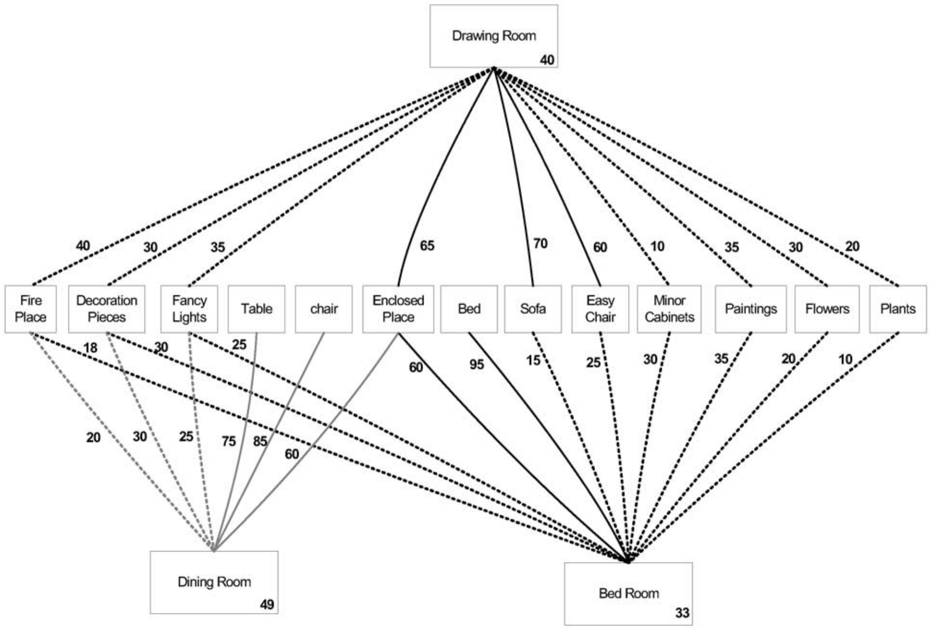


Fig. 4. Lower level concept.

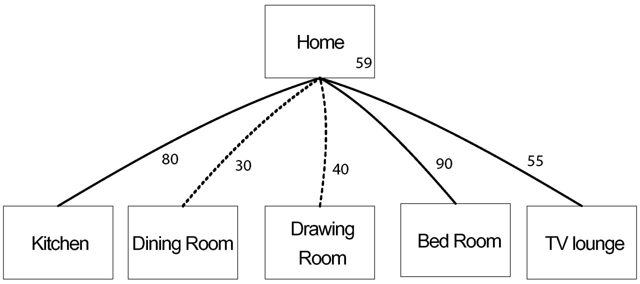


Fig. 5. Higher-level concept.

table, and chair. Now one needs to find these features in the semantic graph, a hash table can be used for this, which will generate pointers to the corresponding nodes of these objects in the semantic libraries. By traversing the semantics graph, the system finds out that these concepts are contained in the three higher-level concepts, i.e., drawing room, dining room, and bedroom as shown in the Fig. 4. The system invokes the specific feature extractors to extract the missing concepts of each of these higher-level concepts, and then based upon the extracted concepts, the graph will be traversed again as we have explained before, it is an alterative process and may repeat many times. The specific feature extractors may give weights to the extracted features, which will then be compared with the weights in the semantics graphs. Similarly, by traversing the semantic graphs, system will reduce towards the higher-level concept.

4.2.1. *Logical Libraries*

Logical Libraries will extract the class for each set of texture; shape and color from the above image and objects are named accordingly. For example from the Fig. 3 the shapes, colors and texture of elephant, chair and picture are closely related to class animal, class furniture and class picture in the Logical Library. Some of the other shapes may not match with the classes present in the logical library (see Fig. 6).

Logical libraries will contain the shape description method that will generate the numeric shape descriptor vector. Shape descriptor is associated with each stored shape in the logical library that uniquely identifies the shape in the library. This shape descriptor vector remains constant for translation; rotation and scale of image (see Fig. 6). Then the shape descriptor vector is generated for each input sub-image and compared with already stored shape descriptor vector of known objects (Loncaric, 1998). Different viewpoints, color values and texture values of the objects should also be catered for accurate matching. At the end of this module the recognized shapes are given names of objects with which shape descriptor vector was matched. Lastly, original image is automatically annotated with the objects recognized by shape description method.

4.2.2. *Abstract Libraries*

Abstract Library contains the semantic values against each set of context values and combination of class values (see Fig. 9). For example consider the Fig. 7, the extracted classes are class human and class dog background and foreground green (grassy texture) then the semantic value will be "human playing with dog". So if the semantic value extracted at this stage then this value along with image will be stored in database. Now consider the image shown in Fig. 8. There are no semantic values against the combination class tiger and class sofa then the stored laws in the abstract libraries will be considered. Laws are "No wild animal can be presented in room-like environment" so the possible semantics can be:

1) Tigers' picture in a room.
2) Tigers' model or skin in a room.
3) Tigers' view though window.

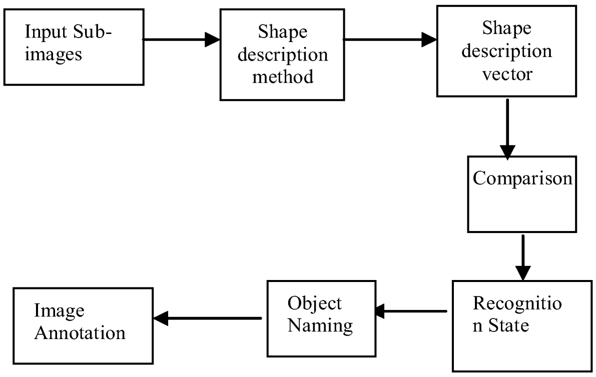


Fig. 6. Logic Library Functionality.

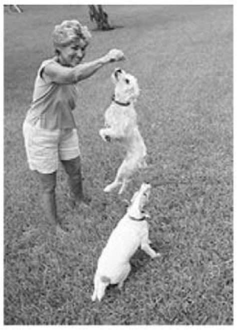


Fig. 7. Input image to abstract libraries.

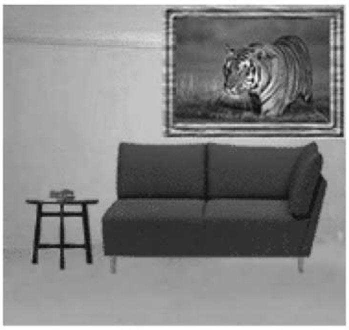


Fig. 8. Input image for abstract library laws.

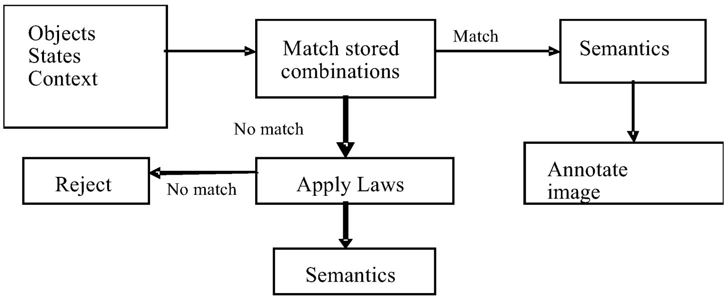


Fig. 9. Working of Abstract Library.

4.2.3. *Refinement Process*

Refinement Process part of the semantic library will be used in the retrieval process to extract the best possible semantics among the various possible semantics by consulting the user profiles. Possible semantics and information from user profiles will come to Refinement Process. This process will find the similarity between the possible semantics from abstract libraries and user profiles and extract the nearest possible semantic. Consider Fig. 10, from user profile, refinement process will come to know that one of the

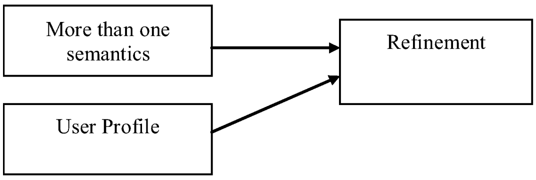


Fig. 10. Refinement process.

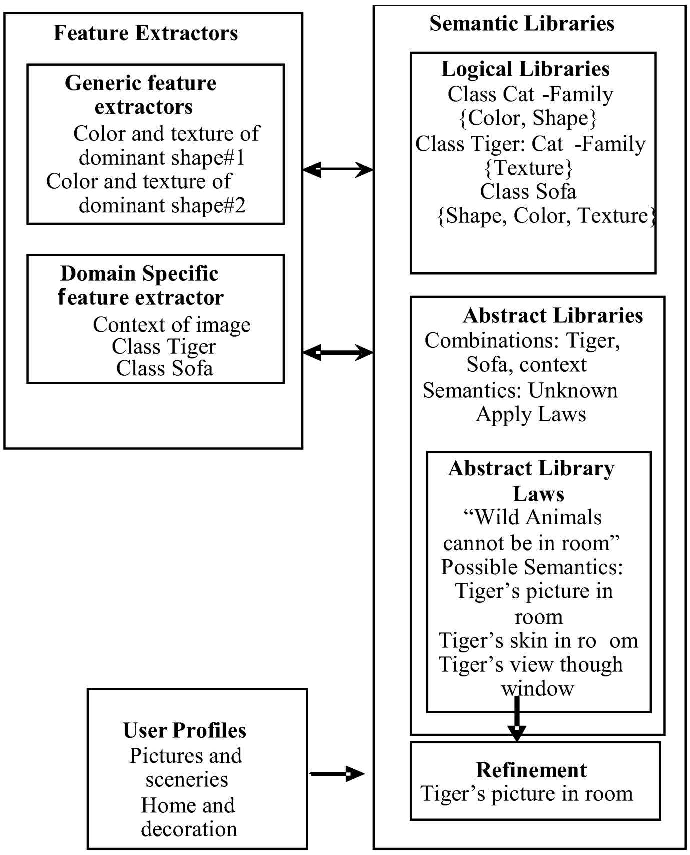


Fig. 11. Structures of feature extractors, semantic libraries and user profiles.

user domains of interest is “pictures and art” and therefore the possible semantics will be “tigers’ picture in room”. Sometimes it may happen that for specific combination of objects and their context do not match with the values in abstract libraries then user profiles will help much in this regard. Fig. 11 shows the relationship of feature extractors, semantic libraries and user profiles.

4.3. *Search Logics*

Search Logics is a kind of a search engine with all the possible components that are required for efficient search and retrieval of a media file. These components include indexing, hashing and ranking etc. As all the storage is XML based, a lot of work has already been done in indexing XML files. Some of these techniques build indexes on XML Documents and store these indexes in an RDBMS. These schemes use different techniques like Inverted Index, B+-tree etc. Some of these even store the XML Documents in the RDBMS. Few of them even do not use RDBMS at all. But still a great deal of research needs to be done for extracting information efficiently from XML documents. Several query languages have been suggested as tools to accomplish this job (Chamberlin *et al.*, 2001; Clark and deRose, 1999; Deutsch *et al.*, 1998; Robie *et al.*, 1998). The biggest factor, while measuring performance of many XML operations, is the performance of the XML parser (Nicola and John, 2003). These operations include; XML conversion into a relational format (Zhang *et al.*, 2001), evaluating XPath expressions (Gottlob *et al.*, 2003) or XSLT processing (Groppe and Böttcher, 2003). Matthias Nicola and Jasmi John (2003) state that XML parsing persists as a major bottleneck and is proven to be a major performance concern seriously threatening the overall success of their project. The issue mentioned is not specific to their project; it is more of a general problem related to all XML applications. So, it is clear that there is a need to develop efficient parsers and a lot of research work is in progress.

Among the present indexing techniques, very few preserve the relationship/semantics as mentioned in (Cohen *et al.*, 2003). In the context of searching multimedia data, the semantics are of great importance. So the indexing technique must preserve the semantics while indexing and the search process must incorporate these semantics.

We do not restrict our architecture to any particular technique. It is an implementer's choice according to the requirements of the system being implemented.

4.4. *Metadata*

The future applications should be based on the Meta data, they should able to locate the related entities, translate them from different Meta data standard to the canonical (single) Meta data standard, and then integrate the results, and show the results to the user, based on the user context.

There are number of different Media Metadata Standards available but any one standard cannot be applied to all types of media. For example IEEE LOM Standards are used for multimedia files that are for Education purposes, where as MPEG 7 are used for Motion Pictures such as videos. Therefore to make the system as generic as possible, there can be more than one metadata standards.

This module contains such Metadata Standards that are used to collect the metadata of a media. The process of gathering metadata is semi-automatic, where the user only needs to specify as few as possible i.e., with the help of "User Profiles".

4.5. *User Profiles*

The user needs to define profiles that can interact with the system. The definition of a profile is a context in order to specify a domain and some basic data about the user. A user can specify more than one profile according to his need. These profiles play key roles in the collection of metadata and by making a query more related to what is required.

Iftikhar *et al.* (2005) stated that this component as a whole has several sub components such as User Profile, Group Profile, Profile generator, Profile Updater, and Profile Identifier. The User Profile contains static and dynamic user profile, preferences and interests. Static user profile would include his/her name, profession, gender, qualification, etc. Dynamic profile includes future events defined by the user, event name, day, and time of the events to occur.

Group Profile has subcomponents like identifier, which helps in identifying the group to which a user belongs at a particular time of the day. Static Group profile component includes group preferences, which uniquely identifies the groups. Each group maintains history including refined query, relevant results, and cluster information, object's information in case of query by image, date, and time. Date and time in the history are used to discard results after some time period. This would avoid over use of memory space.

4.6. *Ontology*

Ontology contains a description of the learning concepts that will add meaning to it. It is a kind of a dictionary that contains knowledge representations that are very similar to "Libraries" in structure. Thus it consists of different classes that are used to expand the query keyword within a specific domain or class. Module is adaptable to new classes as well.

In general the ontology can be defined as a specification of conceptualization (Gómez-Pérez, 2001). Domain specific ontologies are designed for knowledge sharing and reusability (Gruber, 1993). So, we can use ontologies to represent the concept and relationships about a particular domain. To expand the search terms of user query ontologies are useful. The main subject of any ontology is to describe the categories that exist or may exist within the targeted domain. Specific languages are used to construct ontologies (Corcho and Pérez, 2000; Su and Ilebrekke, 2002), and domain specific concepts, relationship, predicates and word senses are represented through that language.

4.7. *Feedback Calculation*

Feedback has now become a core module in CBIR architectures (Iqbal and Aggarwal, 2003; Heesch *et al.*, 2003; Bartolini *et al.*, 2001). In relevance feedback similarity function is used and weights are updated over several iterations of user-system interaction. Both positive and negative relevance feedbacks are used to optimize the query for image. The summarized steps (Bartolini *et al.*, 2001) of feedback mechanism involve query formulation in which original query is submitted by the user, query processing in which query image is compared with database images using distance function and results are presented to the user which are similar to the query image using similarity function, in feedback loop user evaluates the presented results by choosing the options of "relevant" or "non-relevant". At the end by using results from the feedback loop new query is formulated and new distance from the query image is calculated. It is also discussed in (Bartolini *et al.*, 2001) that the relevance feedback mechanism suffers the two drawbacks. First, since relevance feedback is iterative process, convergence could be slow. Second, when the feedback loop is terminated, information about the particular query is not retained for further reuse so for successive queries feedback is started as a new process. Feedback Bypass (Bartolini *et al.*, 2001) is working by properly storing the past query results from feedback loops and make decision whether to bypass the feedback loop completely or predict near-optimal parameters to formulate the new query.

As a result the overall iterations can be reduced giving faster and optimal results. Feedback module is important in our purposed framework since an image can contain many semantics and complex semantic can contain many simple semantics. Feedback module can be helpful in predicting the semantics in which user is interested in. Here the similarity function will find the similarity of semantics of the displayed images to the user and then recalculate the results based on similar semantics.

4.8. *XML Based Database*

This database contains XML files that have three different sections. The first section contains the media file reference, second has the metadata extracted using profiles and given by the user according to standards applied and the last section has the content metadata extracted automatically using Feature Extractor. A sample XML file is shown in Fig. 12.

5. Case Study

In this section, we provide a case study to facilitate a better understanding of our architecture. Consider the image shown in Fig. 13 (a), which is supplied by the user as "Query-by-Example".

In the first step, image is segmented into regions using any of the edge detection methods (see Fig. 13 (b)).The sub-images will take the form as shown in the Fig. 14. At this stage the process of generic feature extraction is completed. Now these sub-images will be fed to logical library for further processing. Each sub-image is passed through shape description method and a shape descriptor vector is generated for each of the sub-images. These vectors are then compared with stored vectors. At the end the objects are named according to the matched classes of objects. So we get named objects as the result from logical library. These named objects will be the part of query features in search logics. Fig. 15 shows named objects of query image.

The next step is to send this main image to the feature extractors along with identified objects (that are bed, chair, lamp, picture, and stool) for confirmation of these objects if

```
<mediafile id="MMDF000120050311">
      <media>
            <url>.../filename.jpg</url>
      <media>
       <contentmetadata>
                 <shapes>
                    <rectangle>10,20,56,98</rectangle>
                    <rectangle>21,10,100,12</rectangle>
                    <rectangle>152,200,220,250</rectangle>
                    <circle>40,50,20</circle>
                    <circle>17,23,6</circle>
                    ...
                 <shapes>
                   <colors>
                    ...
                   <colors>
                  ...
       <contentmetadata>
        <metadata>
            <filename>filename</filename>
            <filetype>jpg</filetype>
            <filesize>500kb</filesize>
            <filewidth>600px</filewidth>
            <fileheight>600px</fileheight>
            <fileowner>abc</fileowner>
                ...
        <metadata>
      <mediafile>
```

Fig. 12. A sample XML file.

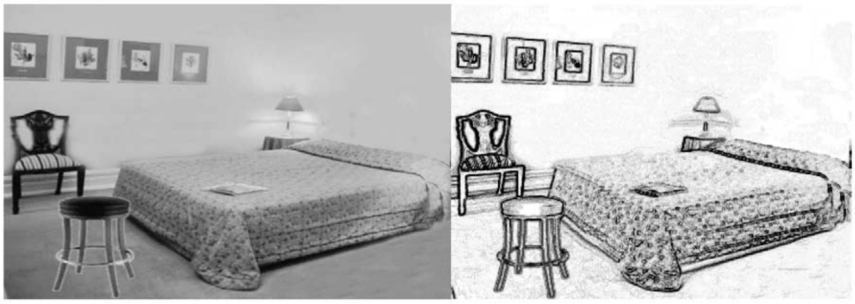


Fig. 13. (a) User Query Image; (b) Edge detection of User Query Image.

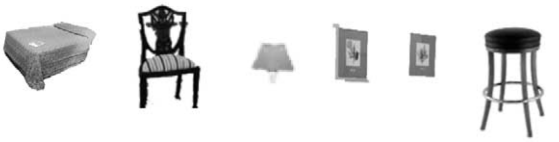


Fig. 14. Some meaningful sub-images of User Query Image.

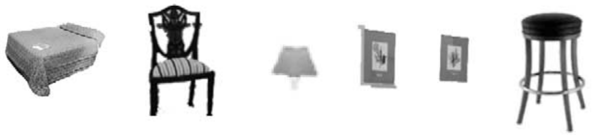
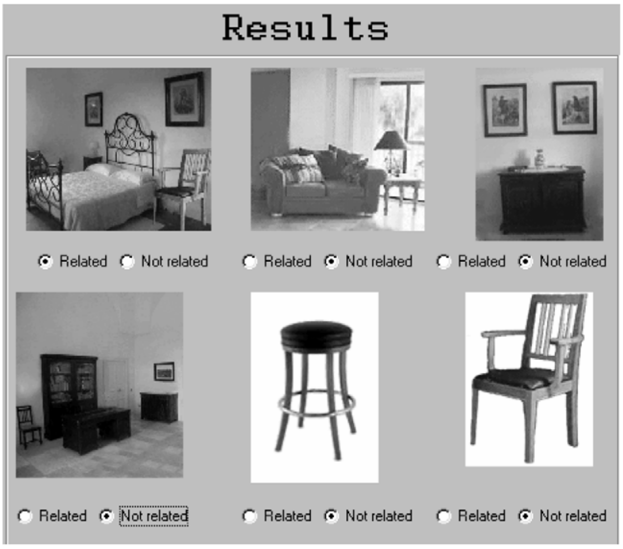


Fig. 15. Name of sub-images User Query Image.



Results

● Related ○ Not related ○ Related ● Not related ○ Related ● Not related

○ Related ● Not related ○ Related ● Not related ○ Related ● Not related

Fig. 16. Results and feedback of the user.

they are really in the image and also if there exist some more of the detected objects in image that were not detected in logical libraries. In our case no such objects are there.

Now names of all objects are passed to abstract library and combination of these object are considered for extraction of semantics. Consider Fig. 13, bed, chair, paintings make the concept of bedroom. Another broad semantic could be room. There is also the possibility that user might be interested in only objects in theimage. So each objects and possible semantics will be considered by the search logics.

Now in search logics, addresses of those images in the database will be located through hash table whose semantics and keywords are similar to the query image and then images are ranked accordingly. Then these query features will retrieve the image from database and results are displayed to the user (Fig. 16 shows the proposed screen layout).

Since the extracted features from query image are chair, picture, lamp, stool, bed, bedroom and room, related images by considering the above extracted features, are displayed as shown in Fig. 16. From user feedback it is calculated that user is interested only in bedroom images so the results are shown in Fig. 17.

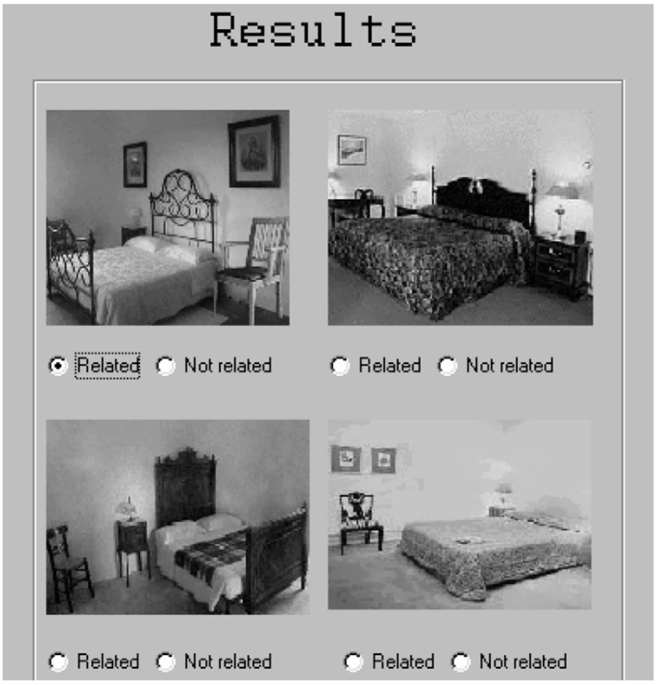


Fig. 17. Results after user feedback.

6. Conclusion and Future Work

In this paper, we have presented a framework for generic multimedia database, which adds new dimension of semantic libraries. Initially the integration of different data storage models was called multimedia database. Most of the research has been done in specific areas of multimedia database (de Vries, 1998; Iqbal and Aggarwal, 2003; Böszörményi *et al.*, 2001; Heesch *et al.*, 2003), e.g., query processing technique, feedback etc. Although there exists many domain specific multimedia database systems (Dimitrova and Golshani, 1994; Olivia *et al.*, 2000; Wang *et al.*, 2005), but unlike other multimedia architectures, our framework concentrates on different important aspects of databases, i.e., semantic library, feature extraction, user profile, ontologies, metadata etc. Some popular image retrieval systems are capable to extract low level feature (Wiesman *et al.*, 2003; Dasiopoulou *et al.*, 2004; Carson *et al.*, 2002; Mehrotra *et al.*, 1997; Faloutsos *et al.*, 1994; Hammiche *et al.*, 2004), i.e., color, texture, shape, location but those systems are not able to extract or conclude semantics.

The framework of generic multimedia database will exploit the strengths of semantic library through semantic graph. We believe that using these semantic libraries in our framework will increase degree of precision for feature extractors. In earlier domain specific multimedia database systems (Dimitrova and Golshani, 1994; Olivia *et al.*, 2000; Wang *et al.*, 2005), no mechanism has been proposed to handle semantics and user profile. In our approach two levels of feature extractions, i.e., generic and domain specific along with semantic libraries will result better and will automatically preserve all the relationships in the global schema. Our framework tries to automate this process of relationship preservation as well.

The development of a primitive prototype to demonstrate the working of semantic libraries, feature extractors and user profiles in Generic Multimedia Database Architecture is in progress.

References

Aghbari, A., and A. Makinouchi (2003). Semantic approach to image database classification and retrieval. *NII Journal*, **7**.

Ardizzoni, S., I. Bartolini and M. Patella (1999). WindSurf: Region-based image retrieval using wavlets. In *Proceedings of 10th International Workshop on Databaseand Extert Systems Applications*.

Bartolini, I., P. Ciaccia and F. Waas (2001). *FeedbackBypass*: *A New Approach to Interactive Similarity Query Processing*. Technical Report CSITE-09-01, CSITE–CNR, 2001. Available at URL `http://www-db.deis.unibo.it/MMDBGroup/TRs.html`.

Bartolini, I., P. Ciaccia and F. Waas (2001). FeedbackBypass: A new approach to interactive similarity query processing. In *Proceedings of the 27th International Conference on Very Large Data Bases* (*VLDB'01*). Rome, Italy. pp. 201–210.

Böszörményi, L., C. Stary, H. Kosch and C. Becker (2001). Quality of service in distributed object systems and distributed multimedia object/component systems. In *Proceedings of the Workshops on Object-Oriented Technology*, vol. 2323. pp. 7–29.

Carson, C., S. Belongie, H. Greenspan and J. Malik (2002). Blobworld: image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal, Machine Intel.*, **24**(8), 1026–1038.

Chamberlin, D., D. Florescu, J. Robie, J. Simeon and M. Stefanescu (2001). *XQuery*: *A Query Language for XML*. W3C Working Draft, Technical report.

Chang, E., K. Goh, G. Sychay and G. Wu (2003). CBSA: content-based soft annotation for multimodel image retrieval using Bayes point machine. *CirSys Video*, **13**(1), 26–31.

Chen, Y., and J.Z. Wang (2002). A region-based fuzzy-matching approach to content-based image retrieval system. *IEEE Trans. On Pattern Recognition Analysis and Machine Intelligence*, **24**(9).

Chen,Y., J.Z. Wang and R. Krovertz (2003). Content-based image retrieval by clustering. *MIR'03*.

Clark, J., and S. DeRose (1999). *XML Path Language* (*XPath*). Version 1.0. W3C Recommendation, Technical report.

Cohen, S., J. Mamou, Y. Kanza and Y. Sagiv (2003). XSEarch: a semantic search engine for XML. *VLDB'03*.

Corcho, O., and A.G. Pérez (2000). A roadmap to ontology specification languages. In *Proceedings of Knowledge Engineering and Knowledge Management. Methods, Models, and Tools*: *12th International Conference*, *EKAW 2000*, vol. 1937. Juan-les-Pins, France.

Cox, I.J., M.L. Millar, T.P. Minka, T.V. Papathomas and P.N. Yianilos (2000). The Bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE Trans. Image Processing*, **9**(1), 20–37.

Cusano, C., G. Ciocca and R.Schettini (2004). Image annotation using SVM. In *Proceedings of Internet Imaging IV*, vol. SPIE 5304.

Dasiopoulou, S., V.K. Papastathis, V. Mezaris, I. Kompatsiaris and M.G. Strintzis (2004). An ontology framework for knowledge-assisted semantic video analysis and annotation. In *Proc. 4th International Workshop on Knowledge Markup and Semantic Annotation* (*SemAnnot 2004*) at the *3rd International Semantic Web Conference* (*ISWC 2004*).

De Vries, A.P. (1998). Mirror: multimedia query processing in extensible databases. In *14th Twente Workshop on Language Technology. Language Technology in Multimedia Information Retrieval*, Enschede, The Netherlands.

Delon, J., A. Desolneux, J. Lisani and A. Petro (2005). *Histogram Analysis and Segmentation by a Contrario Methods*. Univ. Illes Balears, Spain.

Deutsch, A., M. Fernandez, D. Florescu, A.Y. Levy and D. Suciu (1998). *XMLQL*: *A Query Language for XML*. W3C, Technical report.

Dimitrova, N., and F. Golshani (1994). Rx for semantic video database retrieval. In *Proceedings of ACM Multimedia'94*. San Francisco. pp. 219–226.

Döller, M., and H. Kosch (2003). An MPEG-7 multimedia cartridge. In *Proc. Of the SPIE Conference on Multimedia Computing and Networking 2003* (*MMCN 2003*). Santa Clara, California, USA. pp. 126–137.

Dorai, C., A. Mauthe, F. Nack, L. Rutledge, T. Sikora and H. Zettl (2002). Media semantics: who needs it and why? In *Multimedia'02*, December 1–6 , Juan-les-Pins, France.

Dunckley, L. (2003). Multimedia databases – an object-relational approach. ISBN # 0 201 78899 3, pub 2003, Chapter 5, page 117–122.

Duygulu, P.K., N. Barnard, D. Freitas and D.A. Forsyth (2002). Object recognition as machine translation: learning a lexicon for a fixed image image vocabular. In *Proceedings of 7th European Conference on Computer Vision*.

Faloutsos, C., R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz (1994). Efficient and effective querying by image content. *J. Intell. Inform. Syst.*, **3**(3–4), 231–262.

Giunchiglia, F., P. Shvaiko and M. Yatskevich (2003). Semantic matching. in *The Knowledge Engineering Review*.

Gómez-Pérez, A. (2001). Evaluation of ontologies. *International Journal of Intelligent Systems*, **16**(3).

Gomez-Perez, A., and O. Corcho (2002). Ontology languages for the semantic web. In *IEEE Intelligent Systems & Their Applications*.

Gottlob, G., C. Koch and R. Pichler (2003). The complexity of XPath query evaluation. In *PODS 2003*, San Diego, CA.

Groppe, S., and S. Böttcher (2003). XPath query transformation based on XSLT stylesheets. In *WIDM'03*, New Orleans, Louisiana, USA.

Gruber, T.R. (1993). Toward principles for the design of ontologies used for knowledge sharing. Knowledge Systems Laboratory, Stanford University, Technical Report KSL-93-4. *Communications of the ACM*, **37**(7), 48–53147.

Guo, G., A.K. Jain, W. Ma and H. Zhang (2002). Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Trans. On Neural Networks*, **13**(4), 811–820.

Hammiche, S., S. Benbernou, M. Hacid and A. Vakali (2004). Semantic retrieval of multimedia data. In *MMDB'04*, November 13, Washington, DC, USA.

Heesch, D., A. Vanlinsky and S. Ruger (2003). Performance comparison of different similarity models for CBIR with relevance feedback. *Image and Video Retrieval Second International Conference*, *CIVR 2003*. Urbana-Champaign, IL, USA, July 24–25. pp. 456–466.

Huang, T., S. Mehrotra and K. Ramchandran (1996). Multimedia analysis and retrieval system. In *Proc. of 33rd Annual Clinic on Library Application of Data Processing Digital Image Access and Retrieval.*

Iftikhar, N., Z. Zafar and S. Ali (2005). Context-aware querying in multimedia databases – a futuristic approach. In *Proceedings of The 3rd World Enformatika Conference* (*WEC' 05*), vol. 5. April 27–29, Istanbul, Turkey. pp. 140–143.

Iqbal, Q., and J.K. Aggarwal (2003). Feature integration, multi-image queries and relevance feedback in image retrieval. In *6th International Conference on Visual Information System* (*Visual 2003*). Miami, Florida, Sep 24–26. pp. 467–474.

Jeon, J., V. Lavrenko and R. Manmatha (2003). Automatic image annotationand retrieval using cross-media relevance models. In *Proceedings of 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jiang, Y., and Z. Zhou (2004). SOM ensemble-based image segmentation. *Neural Processing Letters*, **20**(3), 171–178.

Jiang, Y., K.-J. Chen and Z.-H. Zhou (2003). SOM based image segmentation. In: G. Wang, Q. Liu, Y. Yao and A. Skowron (Eds.), *Lecture Notes in Artificial Intelligence*, vol. 2639. Springer, Berlin. pp. 640–643.

Kang, F., R. Jin and J.Y. Chai (2004). Regularizing translation models for better automatic image annotation. *CIKM'04*.

Kosch, H., and M. Dollar (2005). *Multimedia Database Systems*: *Where Are We Now*? IASTED DBA Konferenz, Innsbruck, Österreich. Available at:
`www.itec.uni-klu.ac.at/~harald/MMDBoverview.pdf`.

Kosch, H., L. Böszörményi, A. Bachlechner, B. Dörflinger, C. Hanin, C. Hofbauer, M. Lang, C. Riedler and

R. Tusch (2001). SMOOTH – A distributed multimedia database system. In *VLDB'2001*. Rome, Italy. pp. 713–714.

Kosch, H., L. Böszörményi, R. Tusch, A. Bachlechner, B. Dörflinger, C. Hofbauer and C. Riedler (2000). The SMOOTH video db – demonstration of an integrated generic indexing approach. In *Proceedings of the 8th ACM Multimedia Conference*. Los Angeles, USA. ACM Press. pp. 495–496.

Kosch, H., L. Boszormenyi, M. Doller, M. Libsie, P. Schojer and A. Kofler (2005). The life cycle of multimedia metadata. *IEEE MultiMedia*, **12**(1), 80–86.

Kuhn, W. (2003). Semantic reference systems. *International Journal of Geographical Information Science*.

Lavrenko, V., R. Manmatha and J. Jeon (2003). A model for learning the semantics of pictures. In *Proceedings of Advance in Neutral Information Processing*.

Li, J., and J.Z. Wang (2003). Automatic linguistic indexing of pictures by a statistical model approach. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, **25**(19), 1075–1088.

Li, J., and J.Z. Wang (2004). Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Xplore*, **13**(3), 340–353.

Liu,Y., D. Zhang, G. Lu and W. Ma (2005). Region-based image retrieval with high-level semantic color names. In *11th International Multimedia Modeling Conference* (*MMM05*). pp. 180–187.

Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, **31**(8), 983–1001.

Lux, M., and J. Becker (2002). XML and MPEG-7 for interactive annotation and retrieval using semantic metadata. *Journal of Universal Computer Science* (*JUCS*), Hg., 965–984.

McKenzie, M. (1996). *Virage Versus the BLOB – Visual Information Retrieval Technology Drives a Search Engine*. The Seybold Report on Internet Publishing.

Maedche, A., and S. Staab (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems*.

Mehrotra, S., Y. Rui, M. Ortega-Binderberger and T.S. Huang (1997). Supporting content-based queries over Images in MARS. In *Proc. IEEE Int'l Conf. On Multimedia Computing and System*. pp. 632–633.

Mohan, A., C. Papageorgiou and T. Poggio (2001). Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(4), 349–361.

Morgenstern, C., and B. Heisele (2003). Componenet-based recognition of objects in an office environment. In *AI Memo 2003-024*.

Nicola, N., and J. John (2003). XML parsing: A threat to database performance. In *CIKM'03*, Louisiana, USA.

Olivia, F.C., B. Taylor, A. Noakes, S. Markel, D. Torres and K.M. Drabenstott (2000). Browse and search patterns in a digital image database. *Journal Information Retrieval*, **4**, 287–313.

Pal, N.R., and S. Pal (1993). A review on image segmentation techniques. *Pattern Recognition*, **26**, 1277–1294.

Palus, H., and M. Bogdanski (2003). Clustering techniques in colour image segmentation. In *Proceedings of the 4th Symposium on Methods of Artificial Intelligence*. Gliwice, Poland. pp. 223–226.

Rehman, M.U., I. Ihsan, M.U. Ahmed, N. Iftikhar and A. Qadir (2005). Generic multimedia database architecture. In *Proceedings of The 3rd World Enformatika Conference* (*WEC' 05*), vol. 5. April 27–29, Istanbul, Turkey. pp. 128–131.

Robie, J., J. Lapp and D. Schach (1998). *XML Query Language* (*XQL*). Available at `http://www.w3.org/TandS/QL/QL98/pp/xql.html`.

Rode, H., and D. Hiemstra (2005). Conceptual language models for context-aware text retrieval. In *Proceedings of the 13th Text Retrieval Conference* (*TREC*). NIST Special Publications.

Rui, Y., T.S. Huang, M. Ortega and S. Mehrotra (1998). Relevance feedback: a power tool for interactive content-based retrieval. *IEEE Trans. Circuits and video Technology*, **8**(5), 644–655.

Sheikholeslami, G., W. Change and A. Zhang (2002). SemQuery: Semantic clustering and quering on hetrogenous features for visual data. *IEEE Trans. Knowledge and Data Engineering*, **14**(5), 988–1002.

Smith, J.R., and S.F. Chang (1997). Visually searching the web for content. *IEEE Multimedia*, **4**(3), 12–20.

Sonka, M., Hlavac and R.V. Boyle (1998). *Image Processing, Analysis, and Machine Vision*, 2nd Ed. PWS, Pacific Grove, CA.

Su, X., and L. llebrekke (2002). A comparative study of ontology languages and tools. In *Proceedings of Advanced Information Systems Engineering*: *14th International Conference*, *CAiSE*, Toronto, Canada.

Sumengen, B. (2004). *Variational Image Segmentation and Curve Evolution on Natural Images*. Publication#199, Vision Research Lab, Ph.D. Thesis, University of California, Santa Barbara.

Tork Roth, M., M. Arya, L. Haas, M. Carey, W. Cody, R. Fagin, P. Schwarz, J. Thomas and E. Wimmers (1996). The Garlic Project. In *Proceeding of 1996 ACM SIGMOD International Conference on Management of Data*. pp. 557.

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision*, **53**(2), 169–191.

Torralba, A., and P. Sinha (2001). Statistical context priming for object detection. In *International Conference on Computer Vision* (*ICCV'01*), vol. 1. pp. 763–771.

Vailaya, A., M.A.T. Figueredo, A.K. Jain and H.-J. Zhang (2001). Image classification for content-based Indexing. *IEEE Trans. Image Processing*, **10**(1), 117–130.

Wang, F., Y. Ma, H. Zhang and J.T. li (2005). A generic framework for semantic sports video analysis using dynamic Bayesian networks. In *11th International Multimedia Modelling Conference* (*MMM'05*). pp. 115–122.

Wang, J.Z., J. Li and G. Wiederhold (2001). SIMPLIcity: Semanitcs-sensitive integrated matching for picture libraries. *IEEE Trans. Pattern Anal. Machine Intell.*, **23**(9), 947–963.

Wiesman, F., S. Bocconi and B. Arsenijevic (2003). Intelligent information retrieval and presentation with multimedia databases. In *DIR 2003*, *4th Dutch-Belgian Information Retrieval Workshop*.

Zhang, C., J. Naughton, D. DeWitt, Q. Luo and G. Lohman (2001). On supporting containment queries in relational database management systems. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*.

O. Abdul Hamid is a MS student at Mohammad Ali Jinnah University, Islamabad, Pakistan. Omara received her MSc degree in computer science from Fatima Jinnah Women University, Islamabad, Pakistan. Her research activities are in the areas of semantic retrieval in multimedia databases.

M. Abdul Qadir is a professor in Faculty of Engineering & Sciences at Mohammad Ali Jinnah University, Islamabad, Pakistan. He is the head of the CDSC (Center for Distributed & Semantic Computing) research group (http://www.cdsc.jinnah.edu.pk). Dr. M.A. Qadir received the MSc degree in electronics from Quaid-i-Azam University Islamabad, Pakistan, the PhD degree from University of Surrey, UK in parallel processing/distributed computing. His research activities are in the areas of grid computing, parallel processing, distributed computing and context aware computing. Dr. M.A. Qadir is a member of IEEE.

N. Iftikhar is a course/project coordinator in Faculty of Engineering & Sciences at Mohammad Ali Jinnah University, Islamabad, Pakistan. Nadeem Iftikhar received the BS degree with double major in applied mathematics and computer science from Eastern Mediterranean University, Turkey, the MS degree in information science from University of New South Wales, Sydney, Australia. His research activities are in the areas of context aware computing, schema Integration, real-time data warehousing, and information retrieval in multimedia databases.

M. Ur Rehman received his MS degree in multimedia communication from Mohammad Ali Jinnah University, Islamabad, Pakistan. His research activities are in the areas of semantic based information retrieval for xml based distributed multimedia databases and is an active member of CDSC (Center for Distributed & Semantic Computing) research group (http://www.cdsc.jinnah.edu.pk).

M. Uddin Ahmed received his MS degree in multimedia communication from Mohammad Ali Jinnah University, Islamabad, Pakistan. His research activities are in the areas of semantic integration of multimedia objects using synchronized multimedia integration language SMIL and is an active member of CDSC (Center for Distributed & Semantic Computing) research group (http://www.cdsc.jinnah.edu.pk).

I. Ihsan is a lecturer in Faculty of Engineering & Sciences at Mohammad Ali Jinnah University, Islamabad, Pakistan. Imran Ihsan received the MS degree in multimedia communication from Mohammad Ali Jinnah University, Islamabad, Pakistan, MCS degree in computer science from Hamdard University, Islamabad, Pakistan, and MCS degree in physics from University of the Punjab, Lahore, Pakistan. His research activities are in the field of e-learning and multimedia databases. He is active member of the CDSC (Center for Distributed & Semantic Computing) research group (http://www.cdsc.jinnah.edu.pk).

Bendroji multimedijos duomenų bazių architektūra, grįsta semantinėmis bibliotekomis

Omara ABDUL HAMID, Muhammad ABDUL QADIR, Nadeem IFTIKHAR,
Mohib UR REHMAN, Mobin UDDIN AHMED, Imran IHSAN

Semantika grįstam multimedijos duomenų saugojimui ir suradimui reikia labai kruopščios duomenų anotacijos, kuri gali būti daroma rankiniu arba automatiniu būdu. Paieška vyksta geriau pagal rankiniu būdu paruoštą anotaciją negu automatiniu. Tačiau pirmasis būdas pasižymi didesnėmis darbo ir laiko sąnaudomis ir yra sunku ją taikyti didelės apimties multimedijos duomenims. Iš kitos pusės, automatinis anotavimas dažnai naudojamas multimedijos duomenų žemesnio lygio savybėms aprašyti, bet tai neapima semantinių multimedijos duomenų aspektų. Autoriams dar neteko matyti sistemos, kuri automatiškai anotuotų multimedijos duomenis pagal išgautus semantinius požymius. Šiame straipsnyje automatiškai anotuojami paveikslai, išgaunant jų semantiką (aukštesnio lygio bruožus) semantinių bibliotekų dėka. Semantinės bibliotekos naudoja semantinius grafus. Kiekvienas grafas sudarytas iš susijusių koncepcijų, pateikiamų kartu su jų ryšiais. Atskiru atveju pademonstruota, kad siūlomas būdas pagerina semantika grįstą multimedijos duomenų suradimą.