MULTIMEDIA OBJECT MODELLING AND CONTENT-BASED QUERYING

Anastasia Analyti
Stavros Christodoulakis

Multimedia Systems Institute of Crete (MUSIC)
Technical University of Crete
Chania 73100, Greece
{analyti,stavros}@ced.tuc.gr

Abstract

Multimedia Database Systems (MMDS) support rich data types, such as text, images, video, and sound. Queries in MMDSs may refer to the content of the stored multimedia objects. This is called *content-based querying*. However, manual entry of content descriptions is very difficult and subjective. A better approach is to provide automatic content-based retrieval through image, text, and sound interpretation. To support queries by content in a MMDS, multimedia data should be analyzed so that description of their content can be extracted and stored in the database together with the original data. These descriptions are then used to search the MMDS and determine which multimedia objects satisfy the query selection criteria. Because content-based queries tend to be imprecise, database search should be approximate and multimedia objects up to a prespecified degree of similarity with the query specification should be retrieved. This implies the definition of some distance measure between the query and the stored multimedia objects which captures what humans perceive as similarity between the objects. The contents of the multimedia objects may be queried from different aspects, depending on the type of the multimedia objects. For example, subject queries apply to all multimedia types whereas spatial queries apply only to images and video, and temporal queries apply only to video. This paper proposes an object-oriented multimedia representation model and overviews content-based searching in text, image, and video database systems.

1 Introduction

A *Multimedia Database System* (MMDS) deals with the storage, manipulation, and retrieval of all types of digitally representable information objects such as text, still images, video, and sound [Gros94, ChKo95]. Providing mechanisms that allow the user to retrieve desired multimedia information is an important issue in MMDSs. Information about the content of the multimedia objects is contained within the multimedia objects and is usually not encoded into attributes provided by the database schema. Because content equality is not well-defined, special techniques are needed for the retrieval of multimedia objects with content similar to that specified in the user's query. In text databases, information-retrieval techniques allow one to retrieve a document if the document's keywords are close to these specified in the query [Rijs79, CoRi86, Salt89, FrBY92, SAB94]. In image databases, one can retrieve an image if the image's features, such as, shape and spatial position of contained objects, are similar to these specified in the query [ChHs92]. In video databases, one can retrieve a video scene based on the (temporally-extended) actions of the conceptual objects appearing in the scene [SmZh94, JaHa94, DiGo94, DDIK95]. A multimedia document is a structured collection of attributes, text, image, video, and audio data. Multimedia document retrieval should be possible through the structure, attributes, and media content of the multimedia document [CTHP86, Than90, MRT91].

In general, a multimedia object can be viewed as a collection of long, unstructured sequences of bytes, called BLOBs (binary large objects). Because of the large size of BLOBs, database

systems offer special support for reading, inserting, deleting, and modifying BLOB data. Though MMDSs should provide for efficient storage of BLOBs, this is not enough for multimedia application support. Querying long uninterpreted sequence of bytes is limited to pattern matching and reconstruction of a multimedia object from its BLOB may be impossible because of lost structural information. Even if it was possible to extract information of the multimedia object in real time, e.g., using pattern recognition techniques, this would had been completely impractical. Therefore, a MMDS should maintain an internal logical structure of BLOBs and pose semantics on its logical components. Breaking a multimedia object into its component parts allows portions of the BLOB to be indexed and retrieved based on logical structure and semantics.

A logically structured multimedia object is mapped into a hierarchical structure of syntactic components, such as chapters and sections in text, shots and scenes in video. This *logical structure* determines how syntactic components are related to multimedia BLOB contents. In addition to the logical structure, the conceptual structure of a multimedia object should be defined. The *conceptual structure* provides semantic information about the content of the multimedia BLOB. Given a collection of multimedia BLOBs, appropriate representations of their content should be derived and stored in the database for later information retrieval. This involves the detection and identification of the important conceptual objects in the document, image, and video objects stored in the database.

The user should be able to query and easily navigate through the structure of the multimedia object. Multimedia object components are usually identified by pathnames. However, exact knowledge of the structure of the multimedia object is not a realistic assumption and the query language should allow data querying without exact knowledge of the schema. This can be achieved by the partial specification of paths and by querying the data definition (schema) and the actual data in a uniform way [MRT91, CACS94].

Retrieving multimedia objects based on their semantic content can be achieved through manually entered content values and textual descriptions and/or automatic semantic analysis using domain knowledge. The user should be able to query the content of multimedia objects by specifying:

- *values of semantic attributes of the multimedia object.*
  For example, if *beak_shape* is an attribute of the *bird_image* class, the user may request images of birds with “acute” beak. This is the simplest form of content-based retrieval and is usually based on manually entered values, e.g., “acute.” However, because the user may not know all potential attribute values, this type of query can be facilitated by using thesaurus mechanisms that include pictures or diagrams to allow the user to select a value.
- *words or phrases contained in semantic textual descriptions of the multimedia object.*
  For example, the user may request a movie title by describing the movie story. Answering this query requires a similarity measure on text content and mapping of text to the appropriate metric space.
- *global features of the multimedia object.*
  In image and video database systems, this type of query is usually submitted in pictorial form or through a graphical user interface. For example, the user can submit a sample image and request the retrieval of similar images. Retrieved images should have similar global features, such as, colour distribution and texture, as the sample image. The user may select colours from a colour editor and request the retrieval of images having the selected colours in certain percentages. Global features of video objects can be temporally extended in a sequence of frames. For example, shot lighting and shot distance are temporally

extended features of shot-video objects. Answering this type of query requires a similarity measure on global features and global feature extraction from the multimedia objects.

- *visual properties and spatial interrelationships of the conceptual objects appearing in the multimedia object.*
  These queries may be submitted in words, through a query language, or in pictorial form. For example, the user can submit a sample image or a sketch with a number of conceptual objects and request the retrieval of similar images. Similar images present similar conceptual objects with similar spatial interrelationships. Answering this type of query requires application-specific image analysis and understanding for the extraction of primitive objects and the identification of (complex) conceptual objects contained in the image. It also requires a similarity measure on conceptual objects and their visual spatial interrelationships.
- *actual properties and interrelationships of the conceptual objects appearing in the multimedia object.*
  Actual properties and interrelationships of conceptual objects may be different from their visual properties and interrelationships in the multimedia object. For example, the visual properties and interrelationships, lower-upper, large-small in an image may correspond to near-far, in reality.
- *temporal behaviour of conceptual objects contained in the multimedia object.*
  For example, the user may specify one or more conceptual objects, their activities, and temporal interrelationships and request the retrieval of video scenes or shots containing similarly behaved objects. Answering this query requires in addition to still image analysis, the extraction of object trajectories from the video frame sequence and motion analysis for determining object behaviour. It also requires a similarity measure on object motion in a motion picture.

Queries addressing both the contents and the structure of the multimedia objects should be possible. An example query of this type is: Retrieve the documents discussing about *healthy diet* that contain in their cover a picture showing a fruit bowl filled with red and green apples in the center of the image.

MINOS [CTHP86] and MULTOS [Than90, MRT91] are two multimedia document information systems. MINOS introduced an object-oriented approach to model the content and presentation aspects of multimedia documents. A multimedia document in MINOS is a multimedia object, i.e., it has an object identifier. Each multimedia object is composed of attributes, a collection of text segments, a collection of voice segments, a collection of images, and information about other related multimedia objects. Each multimedia object also has information describing how its various parts are interrelated. A text segment may be logically subdivided into title, abstract, chapters, and references. Each chapter is subdivided into sections, paragraphs, sentences, and words. Extraction of components of complex documents is allowed through the query mechanism.

The MULTOS document model incorporates the standard Office Document Architecture (ODA) model for the specification of the logical and layout structure of the documents. In addition to the ODA logical and layout structures, MULTOS requires a document to contain a conceptual structure. Retrieval of documents, based on logical structure, text and image content, is provided. For example, the user may request the retrieval of documents by author *X* that have an abstract containing the word *Y* and contain a graph that looks like graph *Z*. Because the interpretation of the multimedia content may not be certain, MULTOS supports probabilistic queries. For example, the user may request the retrieval of documents containing an facial image with certainty > 70%.

In this paper, we present an object-oriented multimedia model supporting both logical and conceptual structuring of the multimedia objects. Based on this multimedia model, we give a generic model for measuring multimedia object similarity in content-based queries. We overview similarity content-based retrieval and similarity distance measures for text, image, and video databases. Many concepts developed for querying one multimedia type can be applied to the other. This is because, documents can contain images, videos can be viewed as sequences of images (frames), the textual description of a picture or video can be treated as a document, and the content of a document can be viewed as a video script. Similarity content-based retrieval requires appropriate modelling of multimedia objects. Video modelling should support not only querying on text and static images (frames) but also on temporal information associated with a sequence of frames.

The remainder of the paper is organized as follows: Section 2 presents an object-oriented multimedia representation model and defines multimedia object similarity. Section 3 overviews work on similarity searching in text databases. Section 4 overviews work on content-based retrieval in image databases. Video structuring, modelling and querying is presented in Section 5. Finally, Section 6 contains the concluding remarks.

2 Multimedia Object Modelling and Similarity-Based Querying

Each multimedia object has a value, a logical structure of the value, and an interpretation of the value content. The multimedia value may correspond to a number of (single) media values, stored in a number of BLOBs. The logical structure and the requirements for the presentation and storage of the multimedia value are expressed through the multimedia description model. The *multimedia description model* describes the logical composition of the multimedia object, synchronization and timing constraints among its components, media value placement, and parameters needed for the display of the media value. The *multimedia interpretation model* interprets the multimedia value by describing the real-world conceptual objects represented in the multimedia value and their interrelationships.

2.1 Multimedia Description Model

Each multimedia object is composed of a number of (single) media objects and/or already existing multimedia objects. The media objects in a multimedia composition can belong to the same or different media types. The composition can be sequential or parallel. In a sequential composition, the multimedia components are presented in a sequential fashion, i.e., one after the other. For example, a section in an article is the sequential composition of a title, a number of paragraphs or figures, and a number of subsections. A figure is the sequential composition of an image and a caption. In a parallel composition, the multimedia objects correspond to different streams presented in parallel. Overlay of multimedia objects is a special case of parallel composition. For example, in a movie shot, the video and the audio part are played concurrently. A slide-presentation object is the sequential composition of a set of slide-voice objects where each slide-voice object involves the parallel presentation of an image and its verbal annotation. In the case of time-dependent composite multimedia objects, the time dependency between the components of the object is either explicitly stored or derived from other information about the object. For example, in a slide-presentation object, timing information may indicate the start time and duration of the presentation of each slide-voice object.

An object-oriented multimedia description model is presented in Figure 1. The *multimedia_object*, *media_value*, and *multimedia _interpretation* classes are subclasses of the generic class *object*. Each *composite* multimedia object is composed of a list of simpler multimedia objects which can be either composite or simple. A simple multimedia object is considered the unit of the multimedia object and corresponds to one medium, such as, text, image, video, audio. For example, a movie shot is a parallel composite multimedia object which is composed of two media units, a video shot and its corresponding audio. A video shot is the media unit of a movie because it corresponds to an unbroken sequence of frames from one camera. A composite multimedia object can be viewed as the root of a tree whose non-leaf nodes correspond to sequential or parallel multimedia objects indicating the sequential or parallel composition of its children. The leafs of the tree are media_unit objects. Figure 2 depicts the hierarchical composition of a movie.

Each media_unit object has a media value. A media_value object encapsulates an uninterpreted sequence of bytes (a BLOB or part of a BLOB) and a media value descriptor. The media value descriptor contains information required for the appropriate presentation of the BLOB data. For example, a video value descriptor should contain the frame rate, frame width, height, and depth, colour model, and decompression information for the video BLOB. Each multimedia object contains a number of attributes with technical information related to it as a whole. For example, a movie shot may contain information about the lens type, opening and closing shot information, such as, fade and dissolve. A multimedia document is a sequential composite multimedia object with extra bibliographic information. For example, the bibliographic information of a movie includes title, category, abstract, director, cast, and release date.

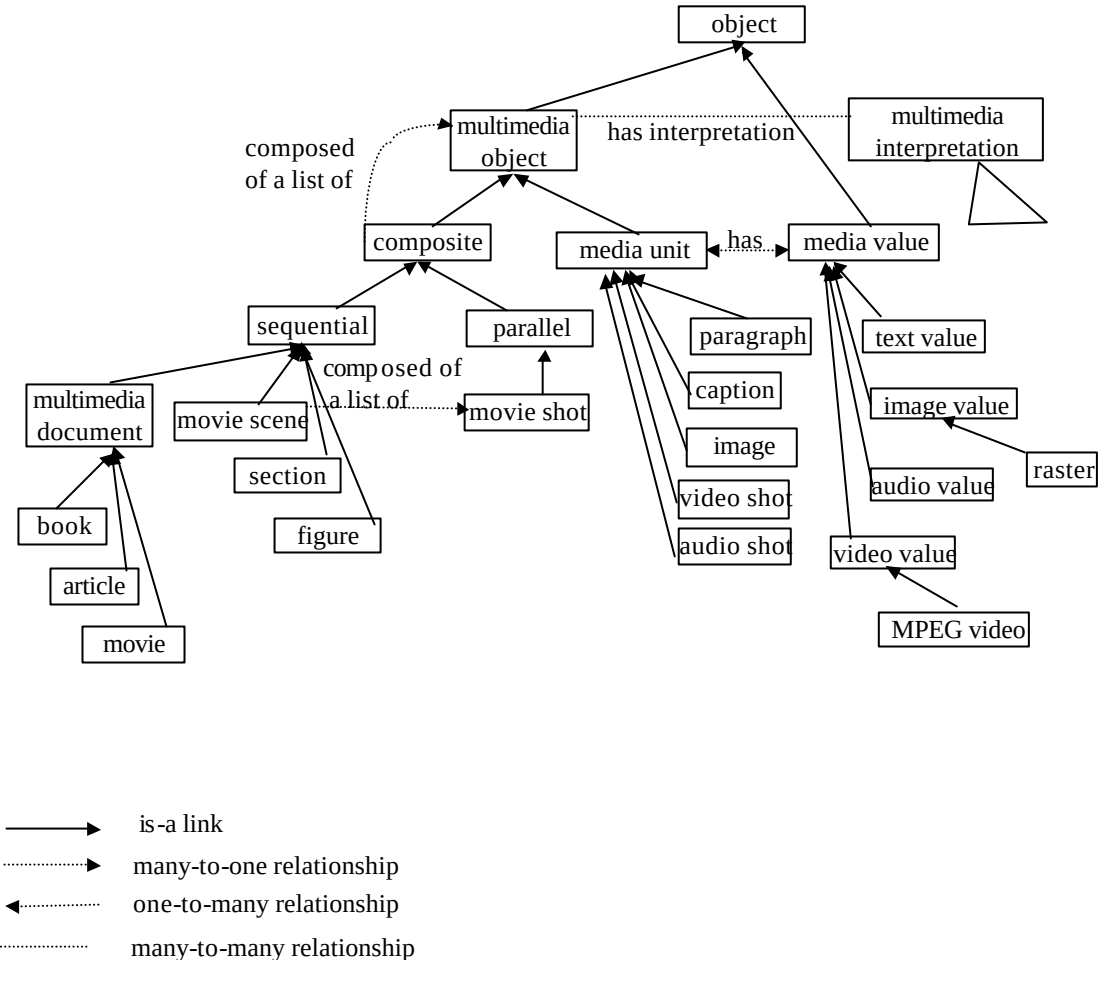


Figure 1. Part of the *is-a* and *aggregation* class hierarchy of the multimedia description model

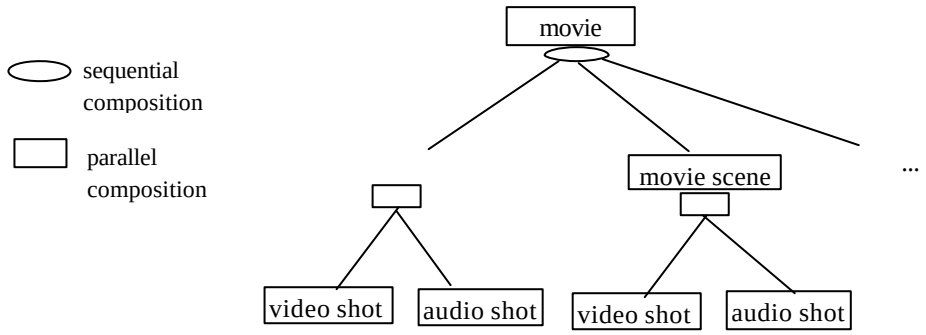


Figure 2. The hierarchical logical composition of a movie

2.2 Multimedia Interpretation Model

The *multimedia interpretation model* interprets the multimedia value by describing the conceptual objects contained in the multimedia value and their relationships. Part of the *is-*a and aggregation class hierarchy of the multimedia interpretation model is depicted in Figure 3. The interpretation of the content of a multimedia object is stored in a *multimedia_ interpretation* object which can be a complex object built from simpler ones. A complex multimedia_interpretation object is linked with its subobjects through the *has_conceptual_subobjects* relationship. During knowledge acquisition, multimedia_interpretation objects are individualized

- *In a composite multimedia object*: by a set of conceptually-related subcomponents. For example, a paragraph and a figure in a magazine section may describe a product. In this case, the multimedia_interpretation object represents the described product.
- *In text or audio*: by the text or audio fragment they are mentioned. For example, one or more words in a text or speech may correspond to a company name.
- *In an image*: by the surface they cover which is usually defined by its enclosing polygon. For example, a tree depicted in an image is individualized by enclosing it in a polygon.
- *In video*: by the surface they cover in a sequence of frames. For example, a person walking in a video shot is individualized by the surface that he/she covers every d^{th} frame of the shot, where *d* is the semantic analysis rate with respect to frames.

The recognition of the conceptual subobjects is not always certain. For example, it is possible that a picture is not clear enough to allow a definite classification of its subobjects, some picture subobjects are partially hidden by other subobjects, or some voice is not recognized with certainty. Object recognition is usually associated with a certainty degree. Because text can be searched efficiently, the raw text value may substitute for the recognition of its conceptual subobjects.

Each multimedia_interpretation object may contain a number of classified objects and a number of unclassified ones. For example, the image of a face will contain classified objects, such as eyes, nose, and mouth, but it may also contain unclassified objects, such as a peculiar

mark. *Classified conceptual objects* are important for the application objects and exist independently of the multimedia content interpretation. Classified conceptual objects are classified into an *is-a* hierarchy of conceptual object classes according to the object-oriented (OO) model. The attributes of these classes correspond to the *static properties* of the conceptual objects. The conceptual object properties that are dependent on the multimedia content (*dynamic properties*) are stored with the multimedia content interpretation. Classified conceptual objects may be complex objects consisting of simpler conceptual objects. In this case, the *has_parts* association between two conceptual objects is represented in the OO schema with a *has_parts* edge between their corresponding classes. Let a multimedia_interpretation object O contain a number of conceptual subobjects O_{sub} through the *has_conceptual_subobjects* relationship). If O represents a (real-world, classified) conceptual object O_c then O_c will probably have subobjects represented (through the *represents* relationship) by the objects in O_{sub}.

Unclassified objects do not belong to conceptual object classes of the database schema. This is because either they are not part of the reality as it is modelled or their class has not be identified. The existence of unclassified objects depends on the multimedia object in which they appear. Thus, unclassified objects are identified only by their properties in the multimedia object and may be given an artificial name.

The interpretation of a media_unit multimedia object belongs to the media_unit_interpretation class and its conceptual subobjects correspond to the conceptual objects appearing in the media_unit value. For example, the conceptual subobjects of an image interpretation are the conceptual subobjects appearing in the image. The interpretation of a composite multimedia object O_{comp} is also composite. The conceptual (first-level) subcomponents of the interpretation of O_{comp} correspond to (possibly nested) logical subcomponents of O_{comp} or subobjects of the media_unit subcomponents of O_{comp}. For example, the interpretation of a movie_shot object (movie_shot_interpretation in Figure 3) has as conceptual subobjects, the interpretation of the video_shot and audio_shot objects subcomponents (video_shot_conceptual_object and script in Figure 3) of the movie_shot object.

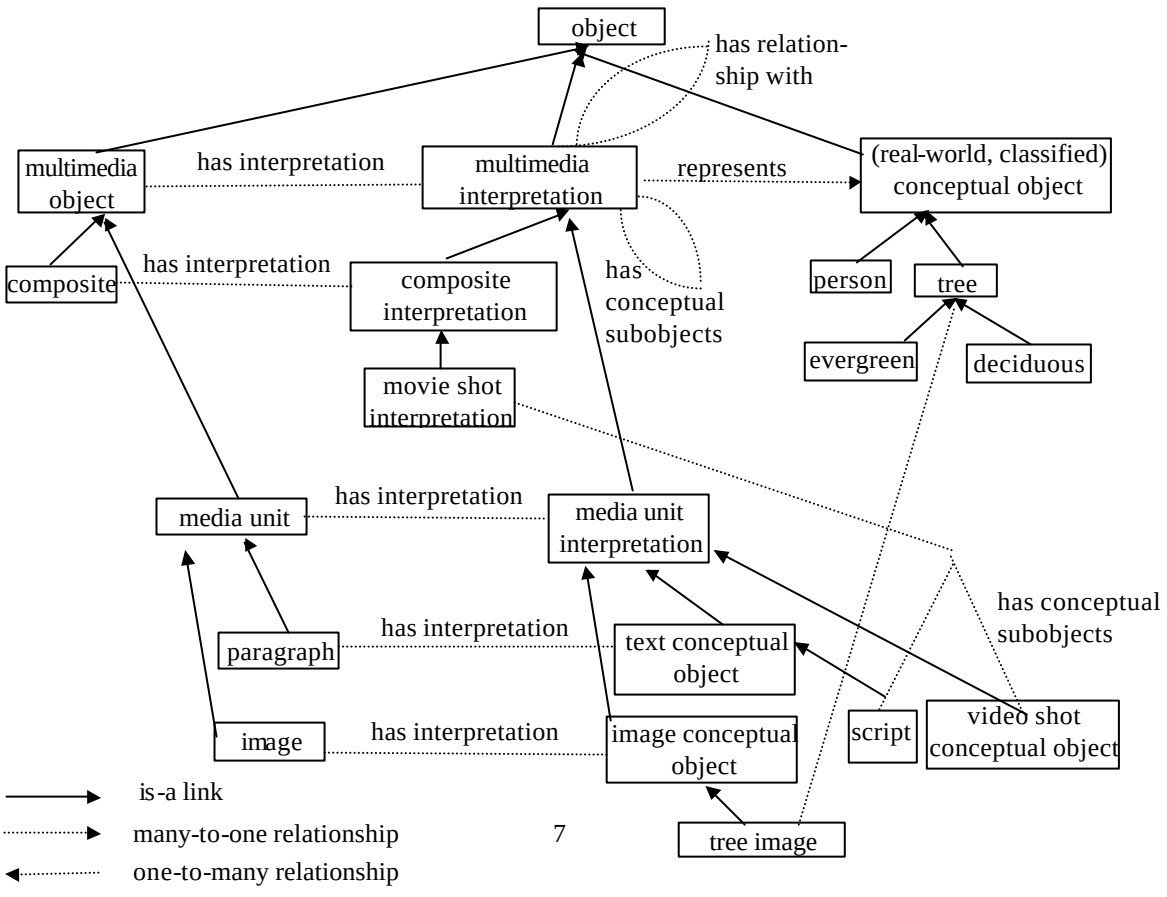


Figure 3. Part of the class hierarchy of the multimedia interpretation model

There are two types of *dynamic* properties and interrelationships among the conceptual subobjects of a multimedia_interpretation object. The first type refers to properties and interrelationships according to the *visual* presentation of the conceptual subobjects in the multimedia object, e.g., left, right, top, down. The second type refers to the *actual* properties and interrelationships among conceptual subobjects of the multimedia object. For example, a small object in an image (visual property) may represent a distant object (actual property) in reality.

The attributes of a multimedia_interpretation object *O*, which is an instance of class *C*, include:

- An inverse pointer to the corresponding multimedia_object, if there is one. Otherwise, a pointer to the corresponding multimedia BLOB or part of the multimedia BLOB.
- A pointer to the classified conceptual object represented by *O*.
- Global properties of *O* requiring exact match, e.g., segmentation technique.
- Global properties of *O* requiring similarity match, such as, content description and colour histogram in images.
- For each conceptual object appearing in *O*,
  a pointer to the corresponding multimedia_interpretation subobject, if there is one, otherwise,
  (i) a name for the conceptual object (if the object is unclassified, the name is artificial)
  (ii) a list of dynamic properties of the conceptual object.

  If the conceptual object appears in all objects of class *C* then the properties in (ii) are usually encoded in corresponding attributes of *C*. Otherwise, the name of the property together with the value appear in the property list of the conceptual object. For example, if *C* is the *facial_image* class then *C* may have an attribute *face_shape* for the shape of the faces depicted in the class images. If *C* is the generic *image* class and an image object depicts a face then the name of the property, i.e., *shape*, together with the value will appear in the property list for the conceptual object *face*.
- Interrelationships among conceptual objects appearing in or represented by *O*.
  If the conceptual objects are common to all objects of class *C* then their interrelationships are usually encoded in attributes of *C*. Otherwise, the name of the interrelationship together with the value appear in the interrelationship list.
- An inverse pointer to the corresponding multimedia superobjects (if there is any).

This is a *semi-structured* approach because each object has a set of standard properties belonging to the database schema and a set of properties that do not belong to the class definition. For example, the property "smiles" for a face in a facial-image may not be a standard attribute of the *facial-image* class. Using the schema in Figure 3, we give an example conceptual structuring for the *image_conceptual_object O* corresponding to the image in Figure 4. The attributes of object *O* include:

- A reverse pointer to the corresponding *image* multimedia object.

- Global properties (requiring inexact match), e.g., *colour_histogram*, *content_description*.
- For the conceptual subobject *house*: a name and a list of dynamic house properties, e.g., (*colour*: white). This conceptual subobject is unclassified because there is no class *house* in the database schema, i.e., *house* is not part of the modelled reality.
- For the conceptual object *tree*: a pointer to the subobject O_{tree} corresponding to the segmented subimage of the tree.
- A list of *house* - *tree* interrelationship properties, e.g., (*visual_spatial_relationship*: house *left_of* tree).

The image conceptual subobject O_{tree} is member of the *tree_image* class and has the following attributes:

- A pointer to the segmented region in the original image BLOB.
- Visual properties (requiring exact match), e.g., *height.*
- Visual properties (requiring inexact match), e.g., *shape*.
- A pointer to the superobject *O*.

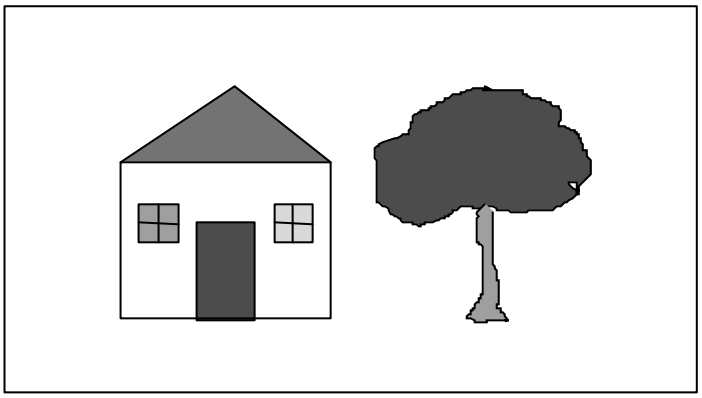


Figure 4. The presentation of an image BLOB

To compare multimedia objects based on their content, we need a measure of content similarity. Let Q be a query object with property values q_1, ..., q_n and conceptual subobjects c_1, ...,c_k. If P is a multimedia object in the database with corresponding property values p_1, ...,p_n then the distance between Q and P is defined as

$$Dist(Q,P) = \min_F \left(\sum_{i=1}^{n} dist(q_i, p_i)^d + \sum_{i=1}^{k} Dist(c_i, F(c_i))^d + \sum_{i,j=0,\ldots,k} dist(c_i, c_j, F(c_i), F(c_j))^d \right)^{1/d}$$

where

c_0 is the conceptual object represented by Q,

F is a mapping from the subobjects in Q to the subobjects in P,

$dist(q_i, p_i)$ denotes the distance between the property values q_i and p_i,

$dist(c_i, c_j, F(c_i), F(c_j))$ denotes the distance between the interrelationship of c_i, c_j in Q and the interrelationship of $F(c_i)$, $F(c_j)$ in P, and

d is the order of the metric (d=1 for the city-block distance and d=2 for the Euclidean distance)

The above definition of $Dist(Q, P)$ is recursive because the distance between objects Q and P depends on the distance of their subobjects, c_i and $F(c_i)$, for $i = 1, ...,k$. If q_i and p_i require exact match and $q_i \neq p_i$ then $dist(q_i, p_i)=\propto$. The same is true for $dist(c_i, c_j, F(c_i), F(c_j))$. This implies that if a conceptual subobject c of Q is of class C_q and $F(c)$ is of class C_p and neither C_q *is-a* C_p nor C_p *is-a* C_q then $Dist(c,F(c))=\propto$ and thus, $Dist(Q, P)= \propto$. In the case of similarity match, computing $dist(q_i, p_i)$ and $dist(c_i, c_j, F(c_i), F(c_j))$ requires the mapping of object properties and relationships to points in a metric space where the city-block or Euclidean distance can be used to express similarity.

3 Content-Based Retrieval in Text Document Database Systems

A document database organizes a collection of documents to facilitate searching. In a text retrieval system, the user should be able to retrieve documents that satisfy his/her query. This implies matching the representation of the query with the representation of all the documents in the database. An exact matching approach will retrieve only the documents that contain exactly the features in the query. In contrast, a similarity-retrieval technique will retrieve a document if the document's concepts are close to the ones specified in the user's query. Retrieved documents are ranked in relevance order by taking into account the importance of concepts in the query and the documents. The importance of a concept within a document is usually derived automatically by considering the relative frequencies of the words in the document. The effectiveness of an approach is measured by its ability to retrieve all and only relevant documents. Specifically, effectiveness is often measured in terms of *recall* and *precision*, where:

$$recall = relevant\ retrieved\ /\ all\ relevant$$

and

$$precision = relevant\ retrieved\ /\ all\ retrieved$$

Document retrieval can be keyword-based or based on full-text. *Keyword-based retrieval systems* support document retrieval based on a manually or automatically generated set of keywords characterizing the document. Document retrieval based on keyword lists is very simple and is achieved by looking up the keywords in an index. However, this approach has several problems, including: (i) choosing keywords which describe the content of a document is not easy, (ii) many document 'misses' will occur in case the keyword list does not reflect adequately the subject content. In *full-text retrieval systems*, documents are considered to have contents which can be queried. Unstructured text documents are represented as a sequence of words. Each word is indexed, excluding commonly used words, such as, 'a', 'the.' Document retrieval is also achieved by looking up the query terms in an index. Queries can be formulated by using keywords interconnected with boolean operators. Automatic indexing techniques are described in [Salt89].

3.1 Similarity Measures for Text Retrieval

The *vector processing model* [Salt91] is a model for determining the similarity between a query and a document. In the vector processing model, both query and documents are represented by weighted term vectors. Similarity measures defined on these vectors are used to reflect the degree of similarity between two documents and between a query and a document.

A weighting system assigns weights to the terms in a document based on their frequency in the document and their discrimination power in the collection of documents stored in the

database. For example, in the *tf×idf* (*term frequency* times *inverse document frequency*) weighting system, the weight w_{ik} of term T_k in document D_i is defined as follows

$$w_{ik} = \frac{tf_{ik} * \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 * \left(\log(N / n_k)\right)^2}}$$

where
tf_{ik} is the frequency of occurrence of term T_k in D_i,
N is the number of documents in the collection,
n_k is the number of documents in the collection containing term T_k.

The reasoning behind the *tf×idf* weighting system is that terms frequent in one document but relatively infrequent throughout the collection are good descriptors of the document's main subject. The denominator in the expression computing w_{ik} is used for document length normalization. This way documents in the collection are given equal chance of being retrieved, independent of their size.

Let $Q = (w_{q1}, ..., w_{qt})$ be a query vector and $D_i = (w_{i1}, ..., w_{it})$ be the vector for document D_i then the similarity of Q and D_i can be defined as follows:

$$Sim_{text}(Q, D_i) = \sum_{k=1}^{t} w_{qk} * w_{ik}$$

Globally similar documents usually have similar logical structure. For this reason, the precision of the retrieval can be improved if pairs of documents with sufficient similarity according to Sim_{text} are further compared for similarity of their logical structures, such as sections and paragraphs [SaBu91, SAB94].

In the above retrieval method, the weight of a term is the same, no matter where it occurs in text. For this reason, the method is better suited for accessing abstracts and short documents than longer documents. However, it is possible that a document contains a dense discussion about a subtopic. The user should be able to query about a subtopic and that subtopic to be specified with respect to the document's main topic. However, this is not possible if querying is against the entire content of a document. This is because the frequency of a subtopic term in a document is usually small and there is no way to differentiate between a subtopic term and a term having a few passing references throughout the document. In [HePl93], the *Texttiling* method is proposed that partitions full-text documents into coherent multi-paragraph units. Texttiling finds coherent subdiscussions by using quantitative lexical analysis. First all pairs of adjacent blocks of text are compared and assigned a similarity value. A low similarity value indicates a potential subtopic boundary. The block size for a document is equal to the average size of the paragraphs in the document. The weight $w_{b,k}$ of a term T_k in a block b of a document D is computed as the *tf×idf* weight of a term with the difference that documents are replaced by the blocks of document D. If n is the number of terms in a document D and $b1$, $b2$ are two blocks in D then the similarity between $b1$, $b2$ is given by

$$Sim_{block}(b1, b2) = \frac{\sum_{k=1}^{n} w_{b1,k} * w_{b2,k}}{\sqrt{\sum_{k=1}^{n} (w_{b1,k})^2 * \sum_{k=1}^{n} (w_{b2,k})^2}}$$

The idea is that if two adjacent blocks share many heavily weighted terms then the two blocks are coherent. Thus, if $Sim_{block}(b1,b2)$, for adjacent blocks $b1,b2$, is larger than a threshold the two blocks $b1,b2$ belong to the same subtopic segment. Based on this automatic segmentation, the user is able to search for a subtopic with respect to a main topic. Thus, in addition to the main topic index, a set of subtopic indexes is built, each local to a document.

Texttiling is essentially trying to identify conceptual objects. However, it considers only adjacent blocks whereas conceptual objects may not appear consecutively in text and may have images associated with them. Even with a manual subtopic segmentation, it is not clear how important the subtopic is and what is the weight of the terms that characterize the subtopic. Yet, detecting clustering of a certain word within a few paragraphs, indicates that the term is more important than the standard word-frequency measures predict. Thus, the importance of a term should take into account the number of different paragraphs that the term occurs in a document.

A connection between the document structure and its content can be obtained by breaking the document into fragments. A fragment is a block of text representing a logical unit, such as, abstract and paragraph. When term references occur within a fragment, there is a better indication for term discussion than if the references were distributed throughout the document. Additional advantages of breaking a document into fragments include: (i) term weights can be different depending on the fragment that the term occurs, and (ii) fragment retrieval is considerably cheaper than whole document retrieval. The retrieval of documents based on fragment content is investigated in [Wilk94]. The similarity of a fragment S with a query Q, denoted by $Sim_{frag}(Q,S)$, is computed similarly to $Sim_{text}(Q,S)$, where the collection of documents has been replaced by a collection of fragments. Fragments are given different weights depending on their type, e.g., a fragment of type *abstract* is given higher weight than another of type *appendix*. Several different ways are considered for computing the similarity, $Sim_{doc}(Q,D)$, of a document D with a query Q based on the similarity of the document's fragments with the query, including:

- Maximum weighted fragment similarity.

$$Sim_{doc}(Q,D) = \max_{S \in D}(weight(S) * Sim_{frag}(Q,S))$$

where $weight(S)$ is the weight of the type of fragment S.

- Sum of weighted fragment similarities.

$$Sim_{doc}(Q,D) = \sum_{S \in D}(weight(S) * Sim_{frag}(Q,S))$$

- Sum of weighted fragment similarities divided by the number of fragments in D.

3.2 Query Expansion Using Concept Relationships

Recall is the portion of relevant documents retrieved from the database. Recall can be improved by augmenting the set of query terms through concept relationships. In [CaDa93], a knowledge-based structure is defined, called *cognitive network* (CN). CN is a directed graph whose nodes represent concepts and its edges represent relationships among these concepts. The strength of a relationship is indicated by a number. The degree of relevance $Drel(c,c')$

between two concepts c, c' is computed based on the strength of the edges in the paths connecting the two concepts. Given a query term t, the user can indicate that he/she is also interested in terms t' in cognitive distance r from t, i.e., $Drel(t,t') < r$. If t' is such a term then t' is added to the query with weight $Drel(t,t')$ (the query vector is expanded). The similarity between a query Q and a document D is computed using the expanded query vector Q_{exp}, that is, $Sim_{exp}(Q, D)= Sim_{text}(Q_{exp}, D)$.

Query expansion over different types of concept relationships is considered in [Voor94]. The query expansion is based on a directed graph whose nodes contain a set of synonym terms, called *synset*, and edges represent different types of relations between synsets. Synset relationships include the *is-a*, *part-of* relationships, and their reverse ones. Given a query Q, the user can indicate that he/she is also interested in terms related to these in Q with a concept relation of type i. The user can also indicate the maximum length of the expansion chain for each type. An expanded query vector is a set of query subvectors Q_0, Q_1,..., Q_n, where Q_0 is the initial query and Q_i, i=1,...,n, is the expansion of the original query over concept relation of type i. The similarity between the expanded query vector Q_{exp} and a document D is computed as the weighted sum of the similarity of the query subvectors and the document.

$$Sim_{\exp}(Q,D) = \sum_{i=1}^{n} w_i * Sim_{text}(Q_i, D)$$

where w_i is a weight reflecting the importance of the expansion of type i.

In [QiFr93], instead of using a general thesaurus, a term-term similarity matrix, called *similarity thesaurus*, is constructed from a given document collection. The similarity between two terms is defined as the summation over all documents in the collection of the degree that a document represents the meaning of the two terms. In this way, the similarity thesaurus reflects the domain knowledge of the collection from which it was constructed. Specifically, if D_k, k=1, ..., N is the set of documents in the collection, the similarity of two terms T_i, T_j is defined as

$$Sim_{term}(T_i, T_j) = \sum_{k=1}^{N} w(T_i, D_k) * w(T_j, D_k)$$

where $w(T_i,D_k)$ denotes the degree that document D_k expresses the meaning of the term T_i (a formula for its computation is given in [QiFr93]).

Let Q =(w_{q1}, ...,w_{qt}) be a query vector. Then, the similarity between Q and a term T is defined as

$$Sim_{q\_t}(Q,T) = \frac{\sum_{i=1}^{t} w_{qi} * Sim_{term}(T_i, T)}{\sum_{i=1}^{t} w_{qi}}$$

The initial query vector Q is expanded by adding to it the vector (w_{e1}, ...,w_{en}), where $w_{ei}=Sim_{q\_t}(Q,T_i)$ iff T_i belongs to the top r ranked terms for $Sim_{q\_t}(Q,T)$, otherwise, w_{ei}=0. In the case that $n>t$, new terms have been added to the original query. With this approach, the query is expanded by adding terms that are most similar to the whole concept of the query rather than to individual query terms. The new query is the expanded query vector Q_{exp}, that is,

$$Sim_{exp}(Q, D)= Sim_{text}(Q_{exp}, D).$$

The concepts of document, term, and concept similarity are used in [AMC95] for the automatic construction of hypermedia from a collection of documents. Each document is an object of the *document* class. Automatic authoring produces a hypertext in which every document object is connected to other relevant document objects by means of links. First, from each document, a set of terms describing its informative content is extracted. Document-to-document links are established on the basis of measures of similarity. Each index term is an object of the *index_term* class and is associated with its frequency of occurrence within the collection. It is also associated with the set of documents from which it was extracted. Term-to-term and document-to-term links are placed based on statistical information of term occurrence within the documents. Concepts are instances of the *concept* class. A relationship between two concepts is represented by an instance of the *relationship class*. Subclasses of the relationship class include: scope, hierarchy, synonymity, and association. Semantic relationships between concepts are set manually by domain experts. Index terms can be concatenated to construct a multiword. If the multiword is a concept then a term-to-concept link is constructed, automatically.

4 Content-Based Querying in Image Database Systems

In a content-based image retrieval system, the primary objective is to retrieve images in the database having similar content as the image described in the query. There are several ways in which queries can be expressed:

- *Directly using a query language*. This implies that the user knows how the system structures image objects, the database schema, and the used vocabulary.
- *By a sketch*. The user sketches the conceptual objects of the image. Both the class of the conceptual objects and their relative position are taken into account for the retrieval of images that contain similar and similarly placed objects.
- *By example*. The user shows an image and requests the retrieval of images with similar content.
- *By association*. An image is associated with some text (caption or related paragraph) and the user specifies associated text conditions in his/her query.

In case images in the database are indexed by a set of keywords or free-text descriptions, image retrieval can be achieved similarly to document retrieval. However, image retrieval using the text-based approach only is very limited in power. Not all image properties can be described, but even if this was possible, obtaining detailed image descriptions requires a great deal of manual effort. The image retrieval capability of a system is greatly improved through the additional use of visual features from the image as retrieval indices. Visual features include (i) properties of the image as a whole, such as colour and texture, (ii) visual properties of the subobjects contained in the image and (iii) visual spatial interrelationships among such subobjects. Chosen visual features should be immune to small distortions, discriminating, and facilitate matching.

The conceptual objects in an image can be obtained through manual or semi-automatic segmentation techniques. In the manual segmentation, the user draws closed contours corresponding to dominant objects or regions in the image, and annotates them with relevant information. In the semi-automatic segmentation, the system initially segments an image by extracting edges and regions of different colour and texture. Then, the resulted segmentation is edited by the user who may delete unnecessary objects, aggregate or split others, or modify their shape.

Segmented subimages are linked with the image from which they were derived. Visual properties of segmented subobjects include shape, position of their mass center, area

perimeter, enclosing rectangle, orientation, colour, texture. Visual spatial interrelationships among segmented subobjects include relative orientation, minimum distance, relative position. As we mentioned in Section 2, image conceptual objects are classified into an *is-a* hierarchy of classes with root the *image_conceptual_object* class (Figure 3). If *C_image* is a subclass of the *image_conceptual_object* class then the class contains the interpretation of images representing conceptual objects of class *C*. For example, the objects in class *tree_image* are conceptual interpretations of images representing trees. Global image interpretation properties and relationships among segmented conceptual subobjects are stored with the image_conceptual_object. If a segmented subobject corresponds to a real-world conceptual object of class *C* then the segmented conceptual subimage and its properties are stored in the *C_image* class (which is a subclass of the image_conceptual_object class). For example, the interpretation of the tree subimage of Figure 4 is an object of the tree_image class (Figure 3).

After selecting the visual features, the objective becomes to develop a similarity measure for obtaining the level of similarity between two images. Content-based queries are usually inexact and the similarity measure should capture what humans consider as similar. To support efficient search of images, the image database system should provide for automatic feature extraction and image index building based on the extracted features.

4.1 Image Similarity Based on Global Image Features

A simple way to provide content-based querying is by using the bitmap representation of the images. Then, queries of the form "Find all images in the database that contain subimage *X*" can be answered by exact or inexact pattern matching. Exact pattern matching will retrieve the images in the database that contain a subimage identical to *X*. Inexact pattern matching will retrieve the images in the database that contain a subimage similar to *X*. A simple similarity measure between images is the fraction of matching black pixels. In [ChTo91], the *Fully Inverted Quadtree* (*FI*-quadtree) structure is proposed which uses the region quadtree representation of image bitmaps and permits the implementation of both exact and inexact image pattern searching within an image set. If images in the database have at most 2^n*2^n pixels, class-*n* region quadtrees are used. In the quadtree representation, the image bitmap is split into four blocks, each block into four subblocks, and so on. Each node of the *FI*-quadtree corresponds to an image block and has four children corresponding to the four subblocks of the node block. Each node holds one bit for each image in the database. When this bit takes the value 1, the corresponding block of the image is black. To perform pattern searching, the query subimage is translated to all possible positions in the 2^n*2^n grid. For each of these positions, the query subimage is encoded to a set of quadtree black-node prefixes. Exact searching will retrieve all these images in the database whose black-node prefixes include the prefixes of the query subimage. Similarity between an image *p* in the database and a query image *q* is measured by the fraction of matched black-node prefixes and the fraction of matched black pixels. Specificaly, the similarity between *q* and *p* is defined as

$$Sim_{pix}(q,p) = \frac{1}{2}\left(\frac{N_{pref}}{Z_{pref}} + \frac{N_{pix}}{Z_{pix}}\right)$$

where N_{pref} is the number of matched prefixes, Z_{pref} is the number of prefixes of query image *q*, N_{pix} is the number of matched black pixels represented by the matched prefixes, and Z_{pix} is the number of black pixels represented by the prefixes of query image *q*. The first fraction expresses similarity in the hierarchical decomposition of the two images whereas the second fraction expresses similarity in the black areas of the two images. This similarity measure

considers matching only at the same quadtree level. Yet, similarity matching using bitmaps should also consider the case that a black block of a database image covers or is covered by a black block of the query subimage.

The *colour similarity measure* corresponds to colour differences between the two compared images. In each image, the overall intensity of each colour is computed by counting the number of pixels with that colour. This will give the colour histogram of the picture. To make the comparison of the colour histograms of two images meaningful, each histogram entry is divided by the total number of pixels in the image. If N is the number of colour intensities in the query image q then the similarity between images q and p is given by the expression

$$Sim_{colour}(q,p) = \frac{1}{N}\sum_{j=1}^{N}\left(1 - \frac{|NH_q(c_j) - NH_p(c_j)|}{\max(NH_q(c_j), NH_p(c_j))}\right)$$

where $NH_X(c)$ denotes the normalized histogram entry for colour intensity c in picture X. The query image q is considered similar to image p stored in the database if $Sim_{colour}(q,p)$ is smaller than a threshold.
Because colour histograms represent the colours of the entire image, positional information of the colour is lost. As a result, the above technique can lead to "false" retrievals. In [LOT93], the use of a multi-level colour histogram is proposed for improving the discrimination power of the colour histogram technique. The root of the multi-level histogram contains the colour histogram of the whole image. The i^{th} level of the multi-level histogram contains 4^{i-1} histograms. The image is split into 4^{i-1} regular regions and the colour histogram of each region is included in the i^{th} level of the multi-level histogram. During query processing, the query and target images are compared using their top-level histograms. If the query and target images are considered similar at the top-level, the next level is searched, and so on. Only when the query and target image are considered similar at the leaf-level, the target image is retrieved. Increasing the number of levels provides a better composition of the image and thus, the accuracy of the method is increased.

Similarly to colour, the *coarseness* and *contrast similarity measures* between two images is defined [CCPL94]. Texture is modelled by coarseness and contrast. The *texture similarity* between two images is obtained by averaging their coarseness and contrast similarities.

To model adjacency relationships between regions and objects in the image, the *colour-pair image* retrieval technique is proposed in [CLP94]. The technique divides each picture into a number of cells. For each pair of adjacent cells, a set of colour-pairs is extracted (a colour-pair is formed by selecting one colour from each cell). The more different are the colours in a colour-pair, the more characteristic the colour-pair is considered to be. If S is the set of the N most characteristic colour-pairs in the query image q (N is a parameter) then the similarity between images q and p is given by the expression

$$Sim_{col\_pair}(q,p) = \frac{1}{|S|}\sum_{i \in S}\left(1 - \frac{|NP_q(i) - NP_p(i)|}{\max(NP_q(i), NP_p(i))}\right)$$

where $NP_X(i)$ denotes the number of occurrences of colour-pair i in picture X divided by the total number of occurrences of all colour-pairs found in X.

One of the major advantages of the above techniques is that they can be carried out without image segmentation and analysis which usually requires human intervention. However, they

are applied to overall image contents without taking into account the characteristics of the individual objects in the image. For this reason, the above techniques are particularly useful when the picture is abstract or does not contain recognizable objects. On the other hand, they are ineffective when the images contain specific objects and the background dominates the characteristics of the whole image. To solve this problem, [CLP94] proposes segmenting the objects within the images so that similarity matching can be performed within the object boundaries.

Colours are usually specified in the CIELUV colour space which provides device-independent colour description and uniform representation of colour differences. A numerical description of colour appearance (lightness, chroma, hue) can be used to obtain colour similarity judgements. The difference between two colours is defined as the Euclidean distance between the two points representing the colours. This difference approximates colour differences as perceived by humans [CaCa83].

An image retrieval system dealing with colour images is described in [BGS94]. There, numerical measurement of features are mapped into qualitative linguistic labels to deal with the uncertainty that characterizes feature descriptions. A similarity condition is expressed as: X is l, where X is a feature and l is a qualitative description of X interpreted as a fuzzy set with a membership function m. The membership function m is elicited by interviewing experts. The values of the membership function are related to the difficulty of attributing label l to the numerical values of X. Similarity is also a fuzzy set because it is impossible to evaluate similarity conditions in a Boolean manner. For example, if v is the distance between two colours then $m_{sim}(v)$ expresses the degree that the two colours are similar.

Selection of an image usually requires the evaluation of a combination of similarity conditions. This implies that a weighted aggregation should be built to account for the importance of the similarity conditions. Thus, a fuzzy set aggregation operator should be defined $h:[0,1]^n \rightarrow [0,1]$ where n is the number of fuzzy sets in the aggregation. An *elementary query* $EQ(x,A_1,A_2,A_3)$ is defined in [BGS92] as the compound fuzzy proposition:

"For selected colour x: Colour is A_1 *and* Coverage is A_2 *and* Distribution is A_3"

where
- A_1 denotes a term in {same, similar, very similar, not very similar},
- A_2 denotes a term in {~10%, ..., ~100%},
- A_3 denotes a term in {compact, spread-out}.

The colour is selected through a pictorial interface in one dimension of lightness, chroma, hue, or in all three. Composition of elementary queries gives rise to complex queries. The degree of similarity between an elementary query and an image pattern characterized by the feature values (a_1, a_2, a_3) is:

$$m_{x,A1,A2,A3}(a_1,a_2,a_3) = h(w_1 * m_{x,A1}(a_1),\ w_2 * m_{x,A2}(a_2),\ w_3 * m_{x,A3}(a_3))$$

where w_i is the weight of the i^{th} similarity condition.
When A_1 equals *same*, "Colour is A_1" represents a boolean condition and $m_{x,same}(a_1)$ takes the values 1 or 0. Thus,

$$m_{x,same,A2,A3}(a_1,a_2,a_3) = m_{x,\text{same}}(a_1) * h(w_2 * m_{x,A2}(a_2),\ w_3 * m_{x,A3}(a_3))$$

A complex query Q is composed by connecting elementary queries of equal importance with the connectives *and, or*. When elementary queries are connected with *and* (*or*), the h

operator is the *min* (*max*) operator applied to $\{m_{x,A1,A2,A3}(a_1,a_2,a_3) | EQ(x,A_1,A_2,A_3)$ is an elementary query of $Q\}$.

An image similarity model based on colour could possibly be defined similarly to the text vector processing model [Salt91], presented in subsection 3.1. An image can form a weighted colour vector, similarly to the weighted term vectors in text. The weight of an image colour should increase with the intensity of the colour in the image and decrease with the proportion of images of the database that have this colour. Image similarity is defined similarly to the text similarity with the difference that colour to colour similarity has to be defined and included in the similarity measure.

Let $Q = (w_{q1}, ..., w_{qN})$ be the query image vector and $I = (w_{i1}, ..., w_{iN})$ be the vector for an image I in the database then the similarity of Q and I can be defined as follows:

$$Sim_{vector}(Q, I) = \sum_{j=1}^{N}\sum_{k=1}^{N} w_{qj} * w_{ik} * sim_{col}(j, k)$$

where N be the number of colours and $sim_{col}(j,k)$ is the similarity of the colours j, k.

4.2 Image Similarity Based on Image Subobject Features

The definitions of image similarity presented in the previous subsection, do not consider the conceptual objects appearing in an image and their interrelationships. A more general similarity measure is used by the content-based retrieval engine CORE, described in [WNML95]. Each multimedia object O has a set of attributes A, a set of feature values F derived from the multimedia BLOB, and a set of conceptual descriptions M for the feature values in F. Features in F include global features of O, features of objects in O, and features of relationships between these objects. Each feature F is either numerically characterized by a measure in the appropriate feature space $F_1 \times ... \times F_p$ or conceptually characterized in M. Conceptual descriptions of features are subjective and usually represented by fuzzy sets or free text given by the user.

Multimedia BLOB, feature measures (F), and feature concepts (M) form a three-layer hierarchy. The multimedia BLOB layer contains the digitized version of a multimedia object, including labelled regions of interest in the multimedia BLOB. The feature-measure layer (resp. feature-concept layer) contains feature measures (resp. feature concepts) of the multimedia BLOB. For example, a facial image BLOB is at the BLOB level, the size of the mouth is at the feature-measure level, and the conceptual interpretation of the size of the mouth, i.e., small, medium, or large, is at the feature-concept level.

In CORE, content-based querying of multimedia objects may be based on the attribute and values of the query object. The result of a content-based query need not be based on exact match. Similarity measures are defined with respect to features and attributes that are important from the user's point of view. For example, in the STAR trademark application [WNML95], developed using CORE, similarity measures for the image and the word in the trademark are defined. Similar objects are presented to the user in ranked order for acceptance. Similarity for words and phonetics is measured by the same-ordered characters (common factors) in the two words normalized by the average number of characters in the two words. Specifically, the similarity measure between two words Q and P is given by

$$Sim_{word}(Q,P) = \sum_i w_i (\frac{l_i}{l_{word}})^{p_i}$$

where l_i is the length of the i-th common factor between the words Q and P, l_{word} is the average length of the two words, w_i is a weight factor, and p_i is a parameter used to increase the effect of longer common factors.

Feature concepts are usually represented by fuzzy sets in fuzzy space with $M_1 \times \ldots \times M_n$. For each fuzzy set M_i, the membership function m_{Mi} represents the certainty that the object feature is described by the concept that the fuzzy set M_i represents. Fuzzy sets overlap which implies that the fuzzy space is not orthogonal. Figure 5 shows fuzzy sets M_1, M_2, M_3 for a fuzzy feature F.

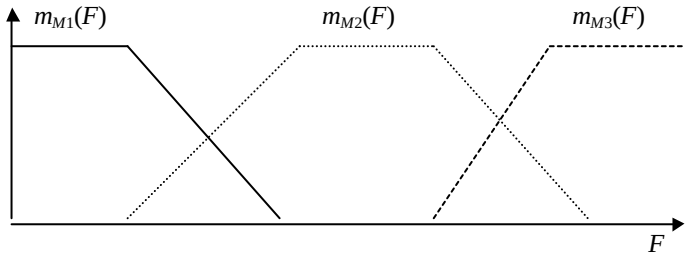


Figure 5. Fuzzy sets M_1, M_2, M_3 for the fuzzy feature F

Let the interpretation for a fuzzy feature F of a query object Q be described by the fuzzy vector Q_i, $i = 1,\ldots,n$, i.e., $Q_i = m_{Mi}(F_Q)$, and the interpretation of the same feature for an object P in the database be described by the fuzzy vector P_j, $j = 1,\ldots,n$, , i.e., $P_j = m_{Mi}(F_P)$, where F_Q, F_P are the values of feature F in the objects Q, P, respectively. The distance between the fuzzy vectors P, Q is defined in [WNML95] as follows

$$dist_{fuzzy} = \sum_{i=1}^{n} \left(|Q_i - P_i| * \sum_{k=1}^{n} cor(M_i, M_k) * |Q_k - P_k| \right)$$

where $cor(M_i, M_k)$ expresses the correlation between the fuzzy sets M_i and M_k.

The overall similarity between a query object and a multimedia object in the database is defined in [WNML95] as the multiplication of the boolean similarity measures for attributes requiring exact match with the weighted sum of the similarities measures for (i) attributes that require inexact match, (ii) features measures, and (iii) feature concepts. A boolean similarity measure between values a, a' equals 1 iff $a = a'$, and equals 0 iff $a \neq a'$. This implies that the attributes requiring exact match A_{exact} should be the same in the query and the retrieved objects. By checking first the equality of attributes in A_{exact} the search space over which similarity matching is carried out, is restricted. In the case that the query object has some attributes undefined, matching is relaxed by considering that the value *undefined* matches any value.

A visual information management system for the retrieval of faces is described in [BPJ93]. Because domain knowledge is confined in the knowledge module, the developed system architecture is not limited to a specific domain. In the facial image retrieval application, a face is considered a conceptual object which contains other subobjects, such as left-eye, right-eye, and nose. Facial image segmentation is the process of determining where in the image the

face subobjects are located. Because of the spatial interrelationships of the face subobjects, an efficient location process is used which focuses on the general location of the subobject. When a facial image is segmented, a certainty value Seg_i is associated with the subobject based on restrictions for the subobject type and spatial restrictions with the other subobjects. For example, the *mouth* subobject should be below the *nose* subobject. The evaluation of a subobject is accepted only when Seg_i is greater than a minimum value. This value is considered in the calculation of image similarity. When a user searches for an image, subobjects are specified in terms meaningful to the user. Then, the system based on statistics for the subobject and domain knowledge will provide a mapping from the user-specified term to an actual value for that feature. For example, the size of a *wide nose* will be calculated based on the variance of the size of the *nose* subobject.

The user can also specify relative values, such as wider, narrower, for refining subobject descriptions. Specifically, let T be the current value for the feature F and s be the standard deviation for F. The *relative feature value change* D is defined as: $D = (s * K * d)/N$, where K is the number of images to be returned to the user, d is the magnitude of change specified by the user, N is the number of occurrences of feature F in the database. The new feature value T' is computed as: $T' = T + D * s$.

The distance between a query facial image Q and a facial image P in the database is defned in [BPJ93] as

$$Dist_{face}(Q,P) = \sum_i \left(W_i * Seg_i * Conf_i * \frac{|Q(i) - P(i)|}{s_i} * C_i \right)$$

where
W_i is the weight of feature f_i,
Seg_i is the segmentation certainty of feature f_i,
$Conf_i$ is the user's confidence in describing f_i,
$Q(i)$ and $P(i)$ are the values of feature f_i in images Q and P,
s_i is the standard deviation of feature f_i, and
C_i is a constant to scale the amount of deviation for different features.

A multimedia information system for electronic book support, called *Hyperbook*, is presented in [TYFN91]. In an application of Hyperbook on birds, the user can retrieve birds by a silhouette of their shape. Each bird-silhouette object has two global features: body axis (6 possible values from *horizontal* up to *vertical*) and size. It also contains six subobjects: *head*, *beak*, *foot*, *tail*, and *body*. Each subobject has its own features, e.g., the features of the *head* subobject are size, height, shape, and crown feather. The features of a bird object are its global features and the features of its subobjects. Because the user may not remember the exact silhouette pattern of the objective bird, silhouette patterns can be entered interactively. From the selected image, the user can specify the parts that he/she feels to be characteristic and also specify a degree of confidence in them. The user need not specify all the parts. The system computes the weighted summation of the distances between the values of the specified features in the query silhouette and the bird images in the database.

Similarity matching between two images requires the comparison of their corresponding features. However, matching corresponding subobjects (subobject features and features of subobject relationships) assumes that all image subobjects are classified. This assumption is dropped in [PeFa96] where each image is assumed to contain a fixed number k of classified

objects and a variable number of unclassified ones. Image descriptions are given in terms of object properties and relationships between the objects. A set of object properties and visual spatial relationships between the objects are proposed for the retrieval of medical images by content. Object properties include *area size*, *roundness*, defined as the ratio of the smallest to the largest second moment, and *orientation*, defined as the angle between the horizontal direction and the axis of elongation. Spatial relationships between the objects include the *distance* of the two objects, defined as the minimum distance between all pairs of line segments of the two objects, and *relative position*, defined as the angle between the horizontal direction and the line connecting the mass centers of the two objects. Object properties and relationships are mapped into points in a multidimensional space. Because the axes should be known in advance and fixed, each image is decomposed into subimages containing one unclassified object and the k classified objects (common to all images). Then, an *R*-tree data structure [Gutt84, SRF87] is used to hold the vectors of all subimages. A query image is also broken down into subimages containing the k classified objects and one unclassified object. Each subquery corresponding to a subimage is mapped to a point in the multidimensional space and treated as range query with radius the maximum acceptable distance between the query and target image.

The case that all objects in an image are unclassified is considered in [PeOr93]. Each image containing n objects is decomposed n times into a group G_k of subimages, for k=1,...,n. Each subimage in G_k contains k objects and is indexed based on the properties of its objects using a multidimentional index structure Ind_k. Because the axes of the multidimensional space should be fixed, the objects in the image are ordered (o_1, ...,o_k) based on the x-coordinate of their mass centers, i.e., $x_{o1} < ... < x_{ok}$, where x_{oi} is the x-coordinate of the mass center of object o_i. Let (o'_1, ...,o'_k) be the ordering of the objects in the image based on the y-coordinate of their mass centers, i.e., $y_{o'1} < ... < y_{o'k}$, where $y_{o'i}$ is the y-coordinate of the mass center of object o'_i. Each object o_i has a fixed number of attributes which are: (i) its index j in (o'_1, ...,o'_k), i.e., $o_i = o'_j$, (ii) the index l of the object o_l in (o_1, ...,o_k) which is closer and surrounds o_i, (iii) the size of o_i, (iv) the roundness of o_i, and (v) the orientation of o_i. Given a query image with m objects, the index Ind_m is searched for the retrieval of images containing m objects with properties matching these of the objects in the query image. In the case that an image in the database contains $n < m$ objects, the query image is broken into subimages containing n objects each.

The spatial relationship among the objects in an image is captured in a 2-D string [CSY87]. A 2-D string (u,v) for an image I is constructed by sorting the objects into two lists, u and v, by their x and y coordinates. Specifically, two object labels o_i, o_{i+1} in u (resp. v) are separating by '<' iff o_i is west of o_{i+1} (resp. o_i is south) of o_{i+1}. The *rank* $r(a)$ of a symbol a in a string u is the number of '<' preceding the symbol a in u. A string u is a *type-i* 1-D subsequence of string u', iff

1. u is a subsequence of a permutation string of u', and
2. if $a_1 w_1 b_1$ is a substring of u, a_1 matches a_2 in u' and b_1 matches b_2 in u' then

(type-0) $r(b_2)-r(a_2) \geq r(b_1)-r(a_1)$ or $r(b_1)-r(a_1)=0$

(type-1) $r(b_2)-r(a_2) \geq r(b_1)-r(a_1) > 0$ or $r(b_2)-r(a_2) = r(b_1)-r(a_1) = 0$

(type-2) $r(b_2)-r(a_2) = r(b_1)-r(a_1)$

If (u,v) and (u',v') are two 2-D strings then (u,v) is a type-i 2-D subsequence of (u',v') iff (i) u is a type-i 2-D subsequence of u', and (ii) v is a type-i 2-D subsequence of v'.

An image Q is a type-i subimage of an image P iff the 2-D string of Q is a type-i subsequence of the 2-D string of P. Thus, the image matching problem becomes a 2-D string matching

problem. However, type-*i* matching between two images *Q* and *P* requires every object in *Q* to be mapped to an object in *P* of the same class and the matched objects in the two images to have exactly the same spatial relationship. However, because images are rarely identical, images with similar spatial arrangement of objects as *Q* should also be retrieved. For this reason, the similarity between two 2-D strings is defined in [LSY89] as the longest subsequence the two strings have in common. It can also be defined as the minimum cost of the transformations required to transform the first string to the second.

In the pictorial database query language *PICQUERY*$^+$, presented in [CITB93], the retrieval of image objects satisfying the predicate clause of a query can be requested. A predicate clause is a sequence of simple predicate clauses "<object> <relational operator> <object value>" connected with a logical operator. The relational operator can be an arithmetic, evolutionary, temporal, spatial, or fuzzy operator. Evolutionary operators include *evolves, splits, fuses*. Temporal operators include *after, before, during, between, in, overlaps, meets, equivalent, adjacent, follows, proceeds*. Spatial operators include *intersects, contains, is collinear with, infiltrates, left of, right of, above, below, in front of, behind*. Fuzzy operators include *similar to*. An <object value> can be a literal, an object id, a query_by_example value, or a fuzzy set. For example, possible queries are:

1. Retrieve the image objects similar to image *X*.
2. Retrieve the hand images of middle-age Caucasian males whose thumb region is similar in shape to that in image *X*.

In [RaSa92], a query language and a query processing technique for image retrieval are presented. Image retrieval is based on access structures representing the content of the images. During the image analysis process, basic objects are recognized first and then more complex objects are derived using domain specific rules. For this to be possible, images must belong to a specific application domain. Because image subobject recognition is not always certain, each subobject is associated with a degree of recognition. Each image is composed of a set of contexts. Each context contains the complex objects which have been recognized in the image with their associated degree of recognition. Each complex object is recursively composed of simpler objects. The similarity between a query and an image depends on the importance of the query clauses, the matching degree between the query clauses and the image, and the uncertainty of the image representation as a result of the image analysis process.

A query *Q* can request the retrieval of *N* images from a domain *D* that best match the query clauses q_i i=1,...,n, each with importance im_i. A query clause q_i indicates that the query image contains objects O_j^i, j=1, ...,n_i , each with a minimum accepted recognition degree r_j^i. Objects O_j^i satisfy constraints c_k^i, k=1, ...,m_i , each with preference degree p_k^i. Each object O_j^i may be further decomposed into simpler objects. Importance and preference degrees are expressed in linguistic terms, each corresponding to a particular value. If a preference degree p_k^i is not specified then p_k^i =1. An example query *Q* is:

Give me 3 images from the domain *kitchen* with

1. (importance degree *HIGH*)
   a table *Table* (recognition degree 0.8) and a chair *Chair* (recognition degree 0.7) such that *Table* and *Chair* are close (preference degree *PREFERRED*) and *Table* is north of *Chair*, and
2. (importance degree *LOW*)
   a table *Sofa* (recognition degree 0.9).

Then,

- for query clause q_1: im_1=*HIGH*=0.95, O_1^1=*Table*, r_1^1=0.8, O_2^1=*Chair*, r_2^1=0.7, p_1^1=*PREFERRED=0.9*, p_2^1=1 (p_2^1 is not specified), and
- for query clause q_2: im_2=*LOW*=0.3, O_1^2=*Sofa*, r_1^2=*0.9*.

The degree of similarity between a query and an image depends on the importance, recognition, and preference degrees in the user query and the recognition degrees of each object in the image. Specifically, if *Q* is a query and *I* is an image then

$$Sim_{query}(Q,I) = \sum_{i=1}^{n}\left(\sum_{k:(c_k^i=true)} p_k^i\right) * \left(\sum_{j\ (C_j^i=true)} R_j^I\right) * im_i$$

where C_j^i is the condition that the object O_j^i is present in *I* with recognition degree R_j^I.

If *I* is an image in the database from the domain *kitchen* and *I* shows a table *Table* with recognition degree 0.9 and a chair *Chair* with recognition degree 0.8 such that *Table* and *Chair* are close then $Sim_{query}(Q,I) = p_1^1 * (R_1^I + R_2^I) * im_1 = 0.9*(0.9+0.8)*0.95$.

5 Searching in Video Database Systems

Similarly to text, a video has a syntactical structure corresponding to a hierarchy of film components such as sequences, scenes, and shots. A shot is an unbroken sequence of frames from one camera and is defined by its beginning and ending frame. The shot is considered in [DSP91] as the fundamental film component. A scene is a collection of adjacent shots focusing on the same objects and describing a complete chain of actions usually in the same place and at the same time. A sequence of frames that alternate between two difference persons as they are discussing, corresponds to multiple shots but to the same scene. A sequence is a group of scenes, not necessarily adjacent, linked together by a common thread of action. For example, the movie heroine may narrate her dream in a number of scenes scattered throughout the movie. The common thread of action in these scenes is the heroine's dream. Identifying the syntactic structure of a video is the first step towards video understanding. Video parsing is the process of identifying the video's syntactic structure and making it explicit.

5.1 Automatic Shot Detection

A lot of research has been done on automatic shot detection [NaTa91, ZKS93]. Most of the developed algorithms detect a shot boundary from the discontinuity between the values of certain video parameters in neighbouring frames. The last and first frames of adjacent shots will be significantly different because a shot boundary corresponds to a camera break (resulting from the application of the camera's stop operation or the trimming of the source video material during video editing). Locating boundaries between camera shots involves establishing suitable metrics that quantify the qualitative interframe discontinuity. A shot boundary is detected when the quantified interframe difference $diff(f_i, f_{i+1})$ exceeds a threshold, where f_i, f_{i+1} are adjacent frames. Parameters used for this purpose include pixel intensities, grey-level and colour intensity histograms, and motion vectors. Though straightforward, these techniques can lead to false detection because video phenomena, such as motion of large or high speed objects and strobe lighting, can cause large interframe

differences in many video parameters. Moreover, they can miss shot boundaries because video effects, such as wipe and dissolve, can cause gradual transition of shots. Reducing the threshold value will reduce this problem but it will make the algorithms more prune to false detection.

The *twin-comparison* algorithm, proposed in [ZKS93] avoids this weakness by establishing two thresholds, t_b and t_s, where $t_s < t_b$. The algorithm is based on the idea that in case of gradual cut, the histogram difference between consequent frames does not exceed t_b but only t_s. Yet, threshold t_b is exceeded by the interframe difference between the first and last frame of the gradual transition period. Whenever $diff(f_i, f_{i+1})$ exceeds t_b, a camera break is reported. When $t_s < diff(f_i, f_{i+1}) < t_b$, the algorithm considers the possibility that frame f_i marks the beginning of a gradual transition period. Subsequently, the differences $diff(f_i, f_j)$ are computed for $j=i+1,...$ until either $diff(f_i, f_j) > t_b$ (a cut is reported), or $diff(f_i, f_j) < t_s$ (hypothesis of gradual cut is dropped). [TATS94] developed an algorithm for shot boundary detection based on the principle that short interval observations are suitable for instant cuts whereas long interval observations are suitable for gradual ones. For this reason, differences in video parameters between frames are calculated and evaluated for small, medium, and long intervals.

5.2 Video Information Modelling and Querying

In retrieving information from a video database, the user should be able to formulate a query related to the contents and the structure of the video. Because, video conceptual information is temporarily extended in a sequence of successive frame, queries may specify that video conceptual objects are in a particular motion (spatio-temporal) relationship. For example, the user may request "the name of the actor *driving* in the opening scene of movie *X*" or "the scenes showing two prime ministers *handshaking*." For this to be possible, video should be structured and information should be extracted from the multimedia BLOB and stored with the appropriate objects according to the multimedia description and interpretation model described in Section 2. Video component classes form an *aggregation* hierarchy which corresponds to the *composed_of* relationship of the video components.

- A *whole_video* object is a *multimedia_document* object with information including (i) bibliographic information, such as, title, category, subject, abstract, year, director, cast, (ii) pointers to video_scene subcomponents, (iii) technical information, such as, coding technique, rate (number of frames recorded per second), quality (width pixels, height pixels, depth of bits per pixel), and (iv) a pointer to the corresponding multimedia_interpretation object which contains content information for the video as a whole, such as, a textual content description.
- A *video_scene* video object contains (i) technical information, (ii) pointers to video_shot subobjects, (iii) a pointer to the corresponding multimedia interpretation object which contains content information for the scene as a whole, such as, dramatic place/time of the scene, scene description, and (iv) derivation information (if applicable), such as, pointers to source video_scene objects from which the video_scene object was derived during video editing, and the corresponding derivation method.
- A *video_shot* object contains (i) technical information, such as, camera position, lens information, real place/time, (ii) derivation information (if applicable), such as, a pointer to the source video_shot object from which the video-shot was derived during video editing and the corresponding derivation method, (iii) a pointer to a *video_value* object containing a pointer to the corresponding fragment in the video BLOB, and (iv) a pointer to the corresponding *video_shot_conceptual* object (the *has_interpretation* relationship).

- A *video_shot_conceptual* object O_c contains (i) content information for the video shot as a whole, such as, description of the general activity of the conceptual objects in the shot, (ii) content information for O_c in every d^{th} frame in the shot, (e.g, motion vector of O_c), where d is a semantic analysis parameter, and (iii) pointers to the video_shot_conceptual subobjects appearing in O_c (the *has_conceptual_subobjects* relationship) and their interrelationships. Note that a video_shot_conceptual object may correspond (i) to the conceptual interpretation of the video shot as a whole, (ii) to conceptual objects appearing in the shot, and (iii) to conceptual subobjects of the complex objects appearing in the shot.

In [DiGo94], the trajectories of rigid objects are extracted through low-level motion analysis. A trajectory is a sequence of motion vectors identifying the displacement of a rigid object. If an object consists of several rigid subobjects then the motion of the object is represented by the trajectories of its rigid subobjects. A possible representation of a trajectory is a sequence of points in frame coordinates corresponding to the positions of the rigid object in a frame sequence. At the highest level of the motion analysis, object motion is associated with a domain-dependent action. Automatic high level analysis requires information about the relative position of object rigid subparts and domain knowledge about rules and constrains governing object actions. In [DiGo94], each video shot is associated with the objects appearing in the shot and their action information (trajectory of rigid subobjects, velocity, activity description). This representation makes possible the request of video scenes containing (i) objects in a particular motion, e.g., a man jumping, (ii) objects in similar motion to that of another object in another video shot, e.g., a man jumping like the child in some other shot, (iii) objects participating in actions in a particular spatio-temporal relationship, e.g., a man waving as a woman walks away.

In [DDIK95], video frames at d distance apart are analyzed and conceptual object are extracted in the form of a bounding volume that describes the spatial projection of the object in the three dimensions. A video is modelled as a sequence of segments, each marked with the appearance of a new conceptual object. In each segment, there is a mapping $l(.)$ from the set of conceptual objects appearing in the segment to their duration in terms of frames. There is also a mapping that maps each conceptual object O to the sequence of motion vectors (Z_1, ...,$Z_{l(o)/d}$), where Z_i is the bounding volume of object O in the $(i*d)^{th}$ frame of the segment. Each *bounding volume* is represented as a tuple (*bounding rectangular*, *centroid*, *depth*, *z*) and each *bounding rectangular* is represented as a tuple (*width*, *height*, *x*, *y*), where (*x*, *y*, *z*) are the coordinates of the lower left front corner of the bounding volume. This modelling allows the expression of spatio-temporal interactions among objects using predicate logic. Moreover, video queries can be expressed using spatial and temporal predicates.

For example, let M be a variable indicating a video object, O and O' be variables indicating two conceptual objects, S be an integer variable indicating the order of a segment in M, and F an integer variable indicating the order of the frame within the segment S. The predicate *on_top*(M, S, F , O, O'), expressing that object O is on top of object O' in some frame (S, F) of movie M, can be defined as

$$\begin{aligned} on\_top(M, S, F, O, O') \leftarrow\ & overlaps_x(M, S, F, O, O'), overlaps_z(M, S, F, O, O'), \\ & bounding\_rectangular(M, S, F, O, Width_O, Height_O, X_O, Y_O), \\ & bounding\_rectangular(M, S, F, O', Width_{O'}, Height_{O'}, X_{O'}, Y_{O'}), \\ & Y_O = Y_{O'}. \end{aligned}$$

where

$overlaps_i(M, S, F, O, O')$ evaluates to true iff the projections of objects O, O' on the i-axis (in frame F of segment S of movie M) overlap, and

$bounding\_rectangular(M, S, F, O, Width_O, Height_O, X_O, Y_O)$ evaluates to true iff ($Width_O$, $Height_O$, X_O, Y_O) is the bounding rectangular of object O in frame F of segment S of movie M.

The answer to the query "Give me the titles of the movies that at some point of time, object *o* appears on top of object *o'* and later, object *o'* appears on top of object *o*" can be obtained as follows:

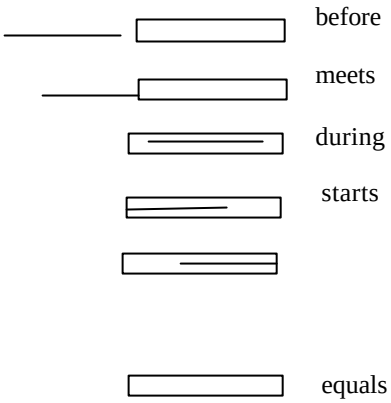
$$requested\_title(Title, O, O') \leftarrow movie\_title(M, Title), on\_top(M, S, F, O, O'), on\_top(M, S, F', O', O), F < F'.$$

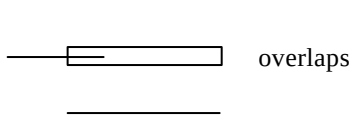
$$requested\_title(Title, O, O') \leftarrow movie\_title(M, Title), on\_top(M, S, F, O, O'), on\_top(M, S', F', O', O), S < S'.$$

The titles of the requested movies are the instantiations of variable *Title* in the evaluation of the predicate $requested\_title(Title, o, o')$.

Conceptual *object actions*, such as, *jumping*, *dancing*, span a number of successive frames. The action predicate $action(Obj, Action, Seg_{st}, Frame_{st}, Seg_{end}, Frame_{end})$ expresses that the conceptual object *Obj* participates in action *Action* from frame (Seg_{st}, $Frame_{st}$) to frame (Seg_{end}, $Frame_{end}$). Action predicates can be defined using logical rules, similarly to the *on_top* predicate in the previous example. The primitive temporal relationships between two object actions *A* and *B* are [Alle83]: $before(A,B)$, $meets(A,B)$, $during(A,B)$, $starts(A,B)$, $finishes(A,B)$, $overlaps(A,B)$, their inverse ones, and $equals(A,B)$. Primitive temporal relationships can be defined in predicate logic using action predicates. Complex temporal relationships between conceptual object actions are defined using the primitive temporal relationships. The primitive temporal relationships between two object actions *A* and *B* are illustrated in Figure 6 (action *A* is indicated by a line segment and action *B* by a rectangular). Inverse relationships can be obtained by inverting *A* and *B*.

A graphical method for expressing queries in video databases, called "Query by Temporal Video Example," is proposed in [DDIK95]. In this method, the user sketches the main objects of interest and their position in a sequence of frames $F_1, \ldots, F_n$, where F_i and F_{i+1} are t_i time units apart, also specified by the user. The matching process should take into account that shapes and positions of sketched objects and temporal information are approximate. Thus, an appropriate similarity measure should be defined.





.

Figure 6. Primitive temporal relationships between two object actions

In [TYFN91], similarity measures for shot retrieval by sketch of the shot content are presented. Each shot with n frames is associated with

- a set of pointers to the (real-world) conceptual objects appearing in the shot.
- a sequence P_1, ..., P_n, where each P_i is a tuple of three matrices (S_x^i, S_x^i, S_x^i). Each matrix S_k^i represents the relative positions between two shot conceptual objects in frame i with respect to the k-axis. In particular, $S_k^i(o, o')$= 1 (resp. 0, -1, *undef*) iff object o is left of (resp. at the same position, right of, undefined) object o' on the k-axis in frame i.
- a mapping from the set of shot conceptual objects to an *is-a* hierarchy of actions, such as, walking, flying.
- a mapping from the set of shot conceptual objects to a sequence of *direction vectors* V_1, ..., V_n. Each vector V =(x,y,z) indicates the direction of the object movement in the x,y,z axis (value -1 denotes negative direction, 0 denotes no direction, and 1 denotes positive direction).

Given a query sketch, the distance between a conceptual object in the query and another in a shot is computed as the distance of their corresponding classes in the *is-a* conceptual object hierarchy. Specifically, if the conceptual objects are instances of the classes Q and P then their distance is computed as

$$dist_{class}(Q, P)=(h_1/H +(1-h_2/H))/2,$$

where
h_1 is the minimum number of nodes in the two paths from the nearest common ancestor M of Q and P to Q and P,
h_2 is the number of nodes from the root to M,
H is the maximum number of nodes on each path from the root to all nodes.

The distance between the relative position of the conceptual objects in the query image and the conceptual objects in a shot frame is defined as the sum of the three distances (corresponding to the three axes) between the relative position matrix of the query and the corresponding one of the shot frame. The distance between two matrices is measured by adding the cosines of the angles of the two vectors corresponding to the same number columns of the two matrices.

The distance between the behaviours of two conceptual objects is computed based on the *is-a* behaviour hierarchy similarly to the distance between two object classes. The distance between the direction of two conceptual objects in a frame is defined as the cosine of the angle of their direction vectors in that frame. The similarity measure between the query sketch and the shot is computed based on the above distances.

6 Conclusions

Multimedia Database Systems (MMDS) support rich data types, such as text, images, video, and sound. In these systems, queries may refer both to the logical structure and the content of the multimedia data. In this paper, we propose an object-oriented multimedia model supporting

both logical and conceptual structuring of the multimedia objects. The proposed model is composed of a *multimedia description model* and a *multimedia interpretation model.* The multimedia description model describes the logical composition of the multimedia object, synchronization and timing constraints among its components, media value placement, and parameters needed for the display of the media value. The multimedia interpretation model interprets the multimedia value by describing the real world conceptual objects represented in the multimedia value and their relationships. Based on this multimedia model, we give a generic model for measuring multimedia object similarity in content-based queries. We overview similarity content-based retrieval and similarity distance measures for text, image, and video databases. We also propose a video structuring model according to the presented multimedia description and interpretation model.

We conclude the paper with several remarks:

- **Real world model**
  Because the multimedia objects represent real-world conceptual objects, a *conceptual_object* class hierarchy should be included in the database schema (Figure 3) to capture the reality of the world. Since it is very difficult and expensive to capture reality, the database real-world model is usually application specific. For example, if we are interested in facial pictures, the real-world model will contain classes, such as, *eyes*, *nose*, and *lips*. In addition to objects, these classes may contain domain knowledge about the approximate shape, and location of the class objects in the face. This will help in the automatic or semi-automatic segmentation of the facial pictures. The model of the real world restricts the user from specifying natural language queries about arbitrary real-world objects to queries in an object-oriented (semantic) language about real-world objects that belong to specific classes.

- **Schema updates**
  The real-world model reflects our knowledge about what is important. However, the importance of things changes and we may have to add new conceptual_object classes in the database schema. Adding a new conceptual_object class implies the identification of the corresponding conceptual objects in the multimedia BLOBs. In addition, conceptual objects belonging to more general classes than the new class may have to be specialized in the new schema.

- **Conceptual objects of unknown class**
  Some conceptual objects in the multimedia data do not belong to classes of the database real-world model (unclassified conceptual objects). These objects are described by their visual attributes, such as, shape, in the multimedia object. Some of these objects may have been automatically identified by some segmentation algorithm but their class is unknown. An unclassified object may become classified in the future when its class is identified.

- **Probabilistic classification**
  When the classification of a conceptual object is not certain, the object belongs to a set of classes with an associated certainty degree. Similarity measures should take this certainty degree into account during the retrieval process.

- **Indexing**
  To support efficient retrieval, several types of indexing should be provided, including:
  1. Indexing of the logical structure of the multimedia object.

2. Indexing of the visual properties and interrelationships (stored in the *multimedia_ interpretation* subclasses) of the conceptual objects.
3. Indexing of the actual (dynamic) properties and interrelationships (stored in the *multimedia_ interpretation* subclasses) of the conceptual objects.
4. Indexing of the static properties (stored in the *conceptual_object* subclasses) of the classified conceptual objects.

- **Querying**
  A MMDS query language should be an object-oriented language that incorporates similarity measures to support both exact and inexact searches on multimedia objects. Queries should refer both to the logical structure of the multimedia objects and the properties and interrelationships (static and dynamic) of the represented conceptual objects. Different weights may be assigned to query search conditions requiring inexact matching. These weights express the importance of the query search conditions in determining the similarity of the query with the database multimedia objects.

BIBLIOGRAPHY

[AMC95] M. Agosti, M. Melucci, F. Crestani, *Automatic Authoring and Construction of Hypermedia for Information Retrieval*, Multimedia Systems, Springer Verlag, 3(1), Feb. 1995, pp. 15-24.

[Alle83] J.F. Allen, *Maintaining Knowledge about Temporal Intervals*, Communications of the ACM, 26(11), 1983, pp. 832-843.

[BPJ93] J. R. Bach, S. Paul, R. Jain, *A Visual Information Management System for the Interactive Retrieval of Faces*, IEEE Transactions on Knowledge and Data Engineering, 5(4), Aug. 1993, pp. 619-628.

[BGS94] E. Binaghi, I. Gagliardi, R. Schettini, *Image Retrieval using Fuzzy Evaluation of Color Similarity*, International Journal of Pattern Recognition and Artificial Intelligence, 8(4), 1994, pp. 945-968.

[CaDa93] D. Cakmakov, D. Davcev, *Experiments in Retrieval of Mineral Information*, Proc. ACM Multimedia 93, 1993, pp. 57-64.

[CITB93] A.F. Cardenas, I.T. Ieong, R.K. Taira, R. Barker, C.M. Breant, *The Knowledge-Based Object-Oriented PICQUERY$^+$ Language*, IEEE Transactions on Knowledge and Data Engineering, 5(4), Aug. 1993, pp. 644-657.

[CaCa83] R.C. Carter, E.C. Carter, *CIELUV Color Difference Equations for Self-luminous Displays*, Color Research and Applications., 8, 1983, pp. 252-553.

[CSY87] S.K. Chang, Q.Y. Shi, C.W. Yan, *Icon Indexing by 2-D Strings*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(3), 1987, pp. 413-428.

[ChHs92] S.K. Chang, A. Hsu, *Image Information Systems: where do we go from here?*, IEEE Transactions on Knowledge and Data Engineering, 4, pp. 431-442.

[ChTo91] J.P. Cheiney, A. Tourir, *FI-quadtree, a New Data Structure for Content-oriented Retrieval and Fuzzy Search*, Proc. of the 2nd Symposium on Spatial Databases (SSD), 1991.

[CTHP86] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, A. Pathria, *Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System*, ACM Transactions on Office Information Systems 4(4), Oct. 1986, pp. 345-383.

[ChKo95] S. Christodoulakis, L. Koveos, *Multimedia Information Systems: Issues and Approaches*, W. Kim (ed.), Modern Database Systems: The Object Model, Interoperability and Beyond, ACM Press, 1995, pp.318-337.

[CACS94] V. Christophides, S. Abiteboul, S. Cluet, M. Scholl, *From Structured Documents to Novel Query Facilities*, Proc. of the ACM SIGMOD International Conference on Management of Data, 1994, pp. 313-324.

[CCPL94] T.S. Chua, S.K. Chan, H.K. Pung, G.J. Lu, *Content-based Image Retrieval System*, Internal Report, Dept. of Information Systems & Computer Science, National University of Singapore, 1994.

[CLP94] T.S. Chua, S.K. Lim, H.K. Pung, *Content-based Retrieval of Segmented Images*, Proc. ACM Multimedia 94, 1994.

[CoRi86] J. Cornelis, J. van Rijsbergen, *A new theoretical framework for imformation retrieval*, Proc. of the 1986 ACM Conference on Research and Development in Information Retrieval, 1986, pp. 194-200.

[DSP91] G. Davenport, T.G.A. Smith, N. Pincever, *Cinematic Primitives for Multimedia*, IEEE Computer Graphics and Applications, July 1991, pp. 67-74.

[DDIK95] Y.F. Day, S. Dagtas, M. Iino, A. Khokhar, A. Ghafoor, *Object-Oriented Conceptual Modeling of Video Data*, Proc. of the 11th International Conference on Data Engineering, Taiwan, 1995, pp. 401-408.

[DiGo94] N. Dimitrova, F. Golshani, *RX for Semantics video Database Retrieval*, Proc. of the ACM Multimedia 94, 1994.

[FrBY92] W.B. Frakes, R. Baeza-Yates (eds.), *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, 1992.

[Gros94] W.I. Grosky, *Multimedia Information Systems*, IEEE Multimedia, 1, pp. 12-24.

[Gutt84] A. Guttman, *R-trees: A Dynamic Index Structure for Spatial Searching*, Proc. of the ACM SIGMOD International Conference on Management of Data, 1984, pp. 47-57.

[HePl93] M.A Hearst, C. Plaunt, *Subtopic Structuring for Full-Length Document Access*, Proc. of the 16th International ACM SIGIR Conference on Research and Development in Informational Retrieval, 1993, pp. 59-68.

[JaHa94] R. Jain, A. Hampapur, *Metadata in Video Databases*, SIGMOD RECORD, 23(4), Dec. 1994, pp. 27-33.

[LSY89] S.Y. Lee, M.K. Shan, W.P. Yang, *Similarity Retrieval of Iconic Image Databases*, Pattern Recognition, 22(6), 1989, pp. 675-682.

[LOT94] H. Lu, B.C. Ooi, K.L. Tan, *Efficient Image Retrieval by Color Contents*, Internal Report, Dept. of Information Systems & Computer Science, National University of Singapore, 1994.

[MRT91] C. Meghini, F. Rabitti, C. Thanos, *Conceptual Modelling of Multimedia Documents*, IEEE Computer, 24(10), Oct. 91, pp.23-30.

[NaTa91] A. Nagasaka, Y. Tanaka, *Automatic Video Indexing and Full Video Search for Object Appearances*, Visual Database Systems ||, IFIP, Elsevier Science Publishers, 1992, pp. 113-127.

[PeOr93] E. G. Petrakis, S. C. Orphanoudakis, *Methodology for the Representation, Indexing and Retrieval of Images by Content*, Image and Vision Computing, 11(8), Oct. 1993, pp. 504-521.

[PeFa96] E. G. Petrakis, C. Faloutsos, *Similarity Searching in Large Image Databases*, IEEE Transactions on Knowledge and Data Engineering, 1996, to be published.

[QiFr93] Y. Qiu, H.P. Frei, *Concept Based Query Expansion*, Proc. of the 16th International ACM SIGIR Conf. on Research and Development in Informational Retrieval, 1993, pp. 160-169.

[RaSa92] F. Rabitti, P. Savino, *An Information Retrieval Approach for Image Datbases*, Proc. of 18th International Conference on Very Large Data Bases (VLDB'92), 1992, pp. 574-584.

[Rijs79] J. van Rijsbergen, *Information Retrieval*, McGraw-Hill, London, 1979.

[Salt89] G. Salton, *Automatic Text Processing - The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, Reading, Mass., 1989.

[SAB94] G. Salton, J. Allan, C. Buckley, *Automatic Structuring and Retrieval of Large Text Files*, Communications of the ACM, 37(2), Feb. 1994, pp. 97-108.

[SaBu91] G. Salton, C. Buckley, *Global Text Matching for Information Retrieval*, Science 253:5023, Aug. 1991, pp. 1012-1015.

[SRF87] T. Sellis, N. Roussopoulos, C. Faloutsos, *The R^+-tree: A Dynamic Index for Multidimensional Objects*, Proc. of 13th International Conference on Very Large Data Bases (VLDB'87), 1987, pp. 507-518.

[SmZh94] S.W. Smoliar, H.J. Zhang, *Content-Based Video Indexing and Retrieval*, IEEE Multimedia, 1(2), 1994, pp. 62-72

[TYFN91] M. Tabuchi, Y. Yagawa, M. Fujisawa, A. Negishi, Y. Muraoka, *Hyperbook: A Multimedia Information System that Permits Incomplete Queries*, Proc. of the International Conference on Multimedia Information Systems '91, 1991, pp.3-16.

[Than90] C. Thanos (ed.), *Multimedia Office Filing and Retrieval: The MULTOS Approach*, North Holland, 1990.

[TATS94] A. Tonomura, A. Akutsu, Y. Taniguchi, G. Suzuki, *Structured Video Computing*, IEEE Multimedia, 1(3) Fall 1994, pp. 34-43.

[Voor94] E.M. Voorhees, *Query Expansion Using Lexical-Semantic Relations*, Proc. of the 17th International ACM SIGIR Conf. on Research and Development in Informational Retrieval, 1994, pp. 61-69.

[Wilk94] R. Wilkinson, *Effective Retrieval of Structured Documents*, Proc. of the 17th International ACM SIGIR Conf. on Research and Development in Informational Retrieval, 1994, pp. 311-317.

[WNML95] J.K. Wu, A.D. Narasimhalu, B.M. Mehtre, C.P. Lam, Y.J. Gao, *CORE: A Content-Based Retrieval Engine for Multimedia Information Systems*, Multimedia Systems, Springer Verlag, 3(1), Feb. 1995, pp. 25-41.

[ZKS93] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, *Automatic Partitioning of Full-Motion Video*, Multimedia Systems, Springer Verlag, 1(1), 1993, pp. 10-28.