World Wide Web (2008) 11:287–311
DOI 10.1007/s11280-008-0045-1

Multimedia Data Modeling Through a Semantic View Mechanism

Qing Li · Jianmin Zhao · Xinzhong Zhu

Received: 04 September 2007 / Revised: 19 February 2008 /
Accepted: 11 March 2008 / Published online: 22 April 2008

Abstract The semantics of multimedia data, which features context-dependency and media-independency, is of vital importance to multimedia applications but inadequately supported by the state-of-the-art database technology. In this paper, we address this problem by proposing *MediaView* as an extended object-oriented view mechanism to bridge the "semantic gap" between conventional databases and semantics-intensive multimedia applications. This mechanism captures the dynamic semantics of multimedia using a modelling construct named *media view*, which formulates a customized context where *heterogeneous* media objects with similar/related semantics are characterized by *additional properties* and user-defined *semantic relationships*. Due to the complex ingredients and dynamic application requirements of multimedia databases, it is however difficult for users to define by themselves individual *media views* in a top–down fashion. To this end, a unique approach of constructing media views logically is devised. In addition, a set of user level operators is defined and implemented to accommodate the specialization and generalization relationships among the views. The usefulness and elegancy of *MediaView* are demonstrated by its application in a multi-modal information retrieval system.

Keywords multimedia data modelling · dynamic semantics · object-oriented view mechanism · context-dependency · media-independency · personalized similarity retrieval · common profile · user profile

Main part of the work by this Qing Li was done when he was on leave from City University of Hong Kong, HKSAR, China.

Q. Li (✉) · J. Zhao · X. Zhu
College of Mathematics, Physics and Information Engineering,
Zhejiang Normal University, Jinhua, China
e-mail: ql@zjnu.cn

J. Zhao
e-mail: zjm@zjnu.cn

X. Zhu
e-mail: zxz@zjnu.cn

1 Introduction

Owning to the expanding Web, recent years have witnessed a phenomenal growth of multimedia information in a variety of types, such as image, video, animation. The vast volume of multimedia data creates the challenge of manipulating them in an organized, efficient, and scalable way, preferably, using a database approach. In the database community, however, although a great number of publications have been devoted to the presentation, indexing, annotation, and querying of multimedia (see, e.g., [3, 8, 15, 22, 26, 31]), relatively little progress has been achieved on the semantic modeling of multimedia, which is of primary importance to various multimedia applications. A typical multimedia application, say, authoring of electronic lecture notes, is more likely to query against the semantic content of data, e.g., "find an illustration of the ANSI/SPARC three-schema database architecture", rather than to query against the primitive data features, e.g., "find all the images in JPEG format with size over 200KB". Therefore, it is critical for a database to model the semantics of multimedia data in order to effectively support the functionality of semantics-intensive multimedia applications. Unfortunately, most existing data models are unable to capture precisely the semantic aspect of multimedia, which features the following two unique properties:

- **Context-sensitive**. Semantics is not a static and inherent property of a media object. (In this paper, a media object refers to an object of any type of modality, such as an image, a video clip, or a textual document.) Rather, the semantic meaning of a media object is influenced by the application that manipulates the object, the role it plays, and other objects that interact with it, which collectively constitute a specific context around this object. As an example, consider the interpretations of van Gogh's famous painting "Sunflower", the leftmost image in Figures 1a, b. When it is placed with the other two images in Figure 1a, which are other paintings of van Gogh, the meaning of "van Gogh's paintings" is suggested. When the same image is interpreted in the context of Figure 1b, however, the meaning of "flower" is manifest. Moreover, a media object may acquire context-specific properties when interpreted in a certain context. For example, as a painting, the "Sunflower" can be described by "artist" and "year", whereas as a flower it can have attributes like "category".
- **User-dependency**. Semantics of media objects can vary from a user's perspective to another's, a phenomenon known as user subjectivity. For example, different users may have and hold different expectations when searching for a romantic picture in terms of features. Currently, the "best" features and the weights of such features are usually fixed in a computer-centered media searching engine, which prohibits the modeling of the difference between the high-level semantic meanings of media objects and users'

Figure 1 (**a**) Context of "van Gogh's paintings". (**b**) The context of "flower".

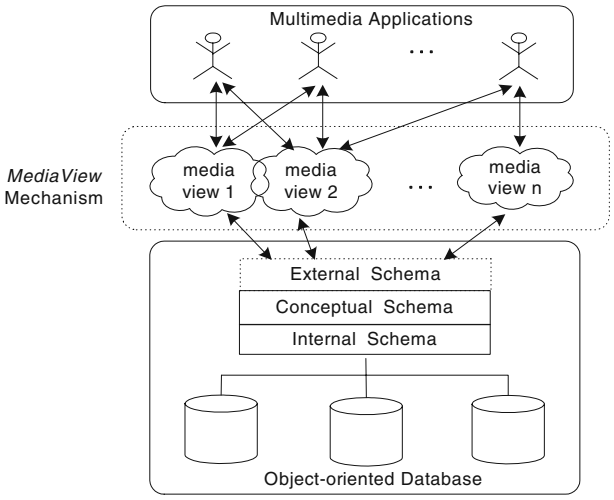


Figure 2 *MediaView* as a "semantic bridge".

subjective perceptions. In addition, assigning weights requires thorough understanding of the low-level features—a task which is a big burden and virtually impossible for an ordinary user.

– **Media-independency**. Media objects of different types of modality (i.e., multi-modal objects) may suggest the related semantic meaning. For instance, the concept of "workflow management system (WfMS)" can be expressed by a textual document, an image illustration, a PowerPoint slide, or a combination of them.

1.1 Objectives and contributions

The dynamic nature of multimedia is fundamentally different from that of the traditional alphanumeric data, whose semantics is explicit, unique, and self-contained. This distinction explains the failing of applying traditional data models to characterize the semantics of multimedia data. For example, in a conventional (strongly typed) object-oriented model, each object statically belongs to exactly one type, which prescribes the attributes and behaviors of the object. This obviously conflicts with the context-dependent nature of a media object, which needs to switch dynamically among various types depending on specific contexts. Moreover, a conventional object model can hardly model the media-independency nature, which requires media objects of different types to have some attributes and methods defined in common.

The incapability of semantic multimedia modeling severely undermines the usefulness of a database to support semantics-intensive multimedia applications. This problem, referred to as the "semantic gap" between databases and multimedia applications, constitutes the major motivation of MediaView as an extended object-oriented view mechanism.

As illustrated in Figure 2, *MediaView* aims to bridge this "semantic gap" by expanding the external view level of the classic three-level database architecture with a set of semantic modelling constructs named *media views*. Each media view, defined as an extended object view, formulates a *customized context* in which the dynamic and elusive semantics of media objects are properly interpreted.

To cope with the dynamic semantics of multimedia, *MediaView* builds the following extensions to the traditional object-oriented view mechanisms (e.g., [1, 12]): (1) A media view can accommodate *heterogeneous* media objects (i.e., objects belonging to different classes) as its members. (2) Objects included as the members of a media view are endowed with *additional properties* that are specific to that media view. (3) Objects in a media view are interconnected by user-defined *semantic relationships*. A media view serves as a container that accommodates semantically related objects and describes these objects by additional properties and semantic relationships.

As MediaView provides a mechanism to link the semantics and media objects, due to the complex ingredients and dynamic application requirements of multimedia databases, it is difficult for users to define by themselves individual media view in a top–down fashion. We therefore also aim to provide a mechanism to systematically generate media views without mass of human effort. So the overall objectives of this paper also include answering the following two key questions:

– How can *MediaView* enhance the performance of multimedia database?
– What is the principle to design the *MediaView* framework?

1.2 Paper organization

The basic facilities of media views are defined in Section 2. In Section 3 we present the design, construction, and evolution issues of *MediaView*; some user-level operators are defined to support customisation of media views. In Section 4, we demonstrate how a real-world application, namely, multi-modal information retrieval, can be elegantly modelled by media views; moreover, we demonstrate how to utilize MediaView mechanism in database navigation, document authoring, and profiled-based retrieval with relevance feedback. Some experimental evaluation results are also included there. Section 5 compares related technologies with *MediaView*. Lastly, the conclusion of the paper is given in Section 6.

2 Fundamentals of *MediaView*

As well known, a multimedia database provides a uniform access point to various types of media, such as text, image, video, music etc. Thus the query performance is the most important characteristic of a multimedia database. To improve the performance, more and more complex media content analysing, modelling, indexing technologies have been employed into multimedia database. However, these are also time-consuming operations. In most cases, even when users issue a keyword query similar to some previous queries, it also needs to perform the query again, which often involves expensive processing. For this reason, *MediaView* mechanism, which links the media objects with semantic contexts, and stores in the database statically, can greatly improve the query performance. When users query for media objects in a certain context, the media objects associated with the media view corresponding to that context could be returned at once as a result.

Intuitively, the semantic link could be generated from historical query results. If a set of media objects is returned as results, we may record it as a *media view*, taking the query as corresponding context. From this point of view, it may be seen as a cache mechanism. With this underpinning strategy, we may ask: what is the cache-hit algorithm? Though we are considering multimedia database with keywords based query interface, a simple keyword

matching will not be enough, for the sake of inflexibility. Instead, a semantic matching algorithm would be more powerful. The latent problems behind this idea are how to deal with the uncertainty of semantics and user behaviours, and how to learn from historical queries. Actually, semantics could be a big problem for the whole multimedia retrieval community. We should balance between digging the depth of semantics and keeping the efficiency of system without too much additional manual work. To some extent, thus, we propose *MediaView* as a general-purpose solution for this goal, and provide the most possible flexibility to cope with complex and customized queries.

2.1 Formalism

MediaView is essentially an extension built on top of a standard object-oriented data model. In an object model, real-world entities are modeled as objects. Each object is identified by a system-assigned identifier, and has a set of attributes and methods that describe the structural and behavioral properties of the corresponding entity. Objects with the same attributes and methods are clustered into classes, as defined below:

Definition 1. *A **class** named as C_i is represented as a tuple of two elements:*

$$C_i =< O_i,\ P_i >$$

1. *O_i is the extent of C_i,which is a set of objects that belong to C_i. Each object $o \varepsilon O_i$ is called an instance of C_i.*
2. *P_i is a set of properties defined by C_i. Each property $p \varepsilon P_i$ is an attribute or a method that can be applied to all the instances of C_i.*

In contrast, a media view as an extended object-oriented view is defined as follows:

Definition 2. *A **media view** named as MV_i is represented as a tuple of four elements:*

$$MV_i =< M_i,\ P_i^v,\ P_i^m,\ R_i, >$$

1. *M_i is a set of objects that are included into MV_i as its members. Each object $o \varepsilon\ M_i$ belongs to a certain source class, and different members of MV_i may belong to different source classes.*
2. *P_i^v is a set of view-level properties (attributes and methods) applied on MV_i itself.*
3. *P_i^m is a set of member-level properties (attributes and methods), which are applied on all the members of MV_i.*
4. *R_i is a set of **relationships**, and each $r \varepsilon\ R_i$ is in the form of $<o_j, o_k, t>$, which denotes a relationship of type t between member o_j and o_k in MV_i.*

The relationship between classes and a media view is exemplified in Figure 3.

As shown in Figure 3a, a set of classes is defined to model media objects of different types, such as *Image*, *VideoClip*, and *Speech*, which are connected into a conceptual schema. From the properties defined in these classes, one can see that they emphasize on the primitive features of media objects, such as the color of images, keywords of text document, which have uniform interpretation irrespective of specific contexts. Although such emphasis is not mandatory, by doing so the conceptual schema is able to provide a context-independent foundation based on which a variety of customized contexts can be formulated. Figure 3b illustrates an example media view called *Workflow*. Each member of

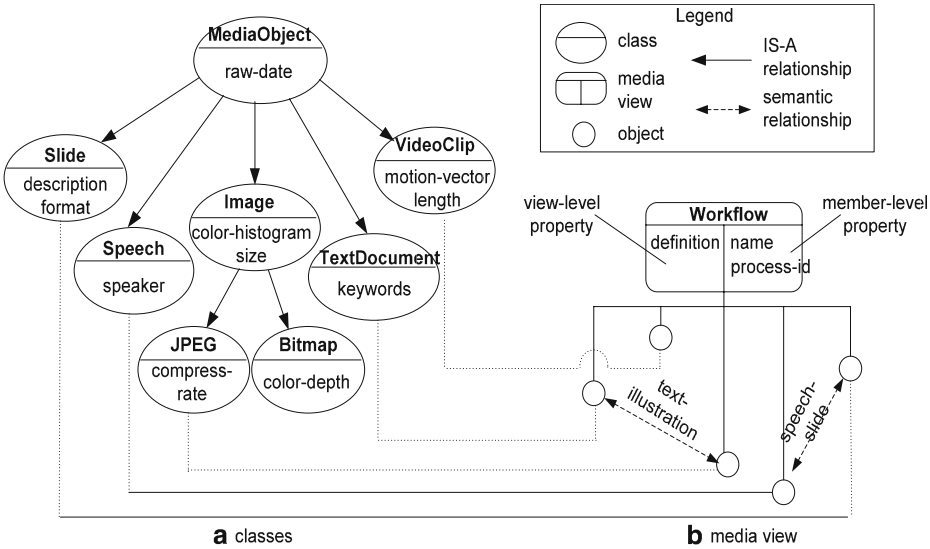


Figure 3 Examples of **a** classes and **b** a media view.

this media view is a media object that is about a specific workflow product, such as a JPEG image illustrating a workflow, a slide as the demonstration of the workflow, etc. Note that all these objects are not created by this MediaView, but are selected from heterogeneous source classes in Figure 3a. However, these objects obtain a set of new (member-level) properties when they become the members of *Workflow*, such as the *name* and *process-id* inside the workflow product. Different from the properties defined in their source classes, their properties in the media view focus on the semantic aspects of media objects. Moreover, a view-level property, *definition*, is used to describe the global property of the media view itself (i.e., the definition of a workflow). Different types of semantic relationships exist between the view members. For example, the "speech-slide" relationship between the *Speech* object and the *Slide* object denotes that the speech accompanies the slide.

In order to support *MediaView* manipulations, a set of basic view operators has been devised, covering the essential manipulation of a media view (e.g., create, delete and so on). These view operators may be classified into two categories according to the types of operands: *type-level* operators that manipulate media views (types) as operands, and *instance-level* operators with view instances (object) as operands. The detailed definitions of these basic operators are given in the Appendix, upon which more sophisticated operations can be implemented as a combination of the basic ones.

3 *MediaView* customisation, derivation and evolution

As we have discussed, *MediaView* provides a contextual mechanism to link the semantics and media objects. Due to the complex ingredients and dynamic application requirements of multimedia databases, however, it is difficult for users to define by themselves individual media views in a top-down fashion. A mechanism is therefore provided to systematically

generate media views without mass of human effort. In addition, necessary facilities are provided to accommodate media views evolution, customisation, and derivation, as detailed in this section.

3.1 MediaView construction

Comparing to those expensive media processing procedures, *MediaView* provides a way to boost the performance of multimedia database query by accumulating previously performed queries. Thus, we synthesize existing information processing technologies to construct the links between media views and concrete media data. Due to the multi-modality of media objects in a multimedia database, we use a multi-system approach in our framework. More specifically, we append a *MediaView Engine* on various keywords based CBIR systems to acquire the knowledge of semantic links between media contents and contexts (queries) from these well designed IR technologies, as Figure 4 shows.

From the queries performed by users, the system is able to learn more about which media objects are semantically similar to each other in a certain context, which are to be recorded and stored in the database for later use. However, different queries may vary greatly with the user liberty of choosing query keywords. To tackle this problem, WordNet [18]—an electronic thesaurus that models the lexical knowledge of English language is adopted, by representing each media view (context) as a concept and organizing the concepts as a hierarchical multi-dimension semantic space following WordNet hierarchies.

In WordNet, a variety of semantic relationships are defined between word meanings, represented as pointers between *synsets*. It is divided into five categories: noun, verb, adjective, adverb, and function word. Hyponymy relationship organizes the meanings of nouns into a hierarchical structure. In WordNet, approximately 57,000 nouns are organized

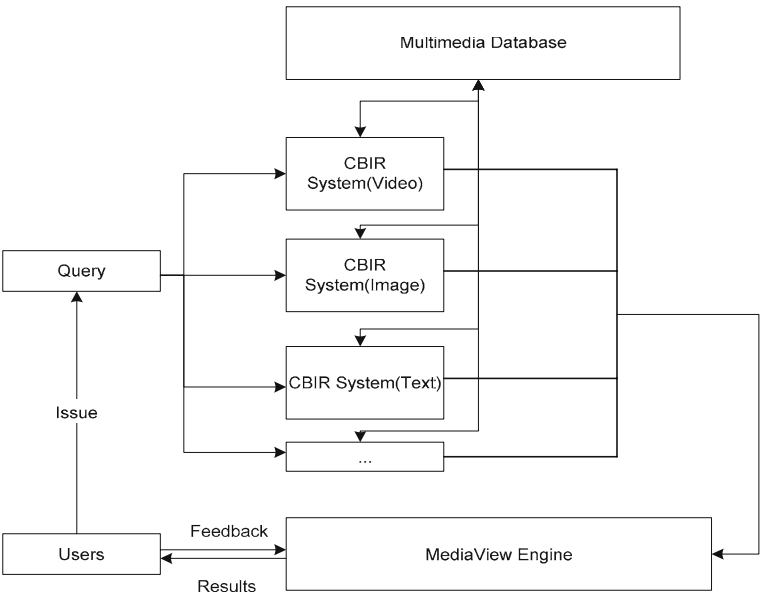


Figure 4 *MediaView* construction processes.

into some 48,800 *synsets*, and the latter are organized as a tree. A synset in WordNet represents a (real world) concept.

Actually, a context could be represented by a concept, e.g., "flower", or a combination of concepts, e.g., "Van Gogh's painting". We call *simple context* as the context which could be represented by a concept. The collection of media views corresponding to all *simple contexts*, therefore organized as the hierarchical structure of WordNet, constitutes the basic architecture of MediaView framework. These media views are called *common* media views.

3.1.1 Hierarchical multi-dimension semantic space

First, we introduce the hierarchical multi-dimension semantic space.

Definition 3. *A **multi-dimensional semantic space** exists under a concept (denoted as "super concept" in* Figure 5*), if there are several sub-concepts related to that concept.*

For example, the concept "Season" has a 4-dimensional semantic space ["spring", "summer", "autumn", "winter"]. Specifically, a sub-concept has an "IS-A" relationship with its super-concept. As a consequence, if some media object is known to be relevant to a super concept, it has a chance to be relevant to one or more of the sub concepts.

– Encoding a Media Object with Probabilistic Tree

By utilizing the concept of *Multi-dimensional Semantic Space*, we could have the knowledge accumulated from previous queries encoded into a *Probabilistic Tree*, as illustrated in Figure 6.

A *Probabilistic Tree* specifies the probability of one media object semantically matching a certain concept in thesaurus. It is encoded as several arrays: {[*point-to-concept, P1, P2 ...*]}. Each array [*point-to-concept, P1, P2 ...*] could be interpreted as: *if a media object is considered as a match of the concept pointed by point-to-concept, then it has the probability Pi to be a match of the i-th sub-concept*. Thus, we also have $\sum_i P_i = 1$ for each array. If, to a specific concept, the corresponding array is missing, it means that we do not have prior knowledge about this semantic space; thus, the *average probability* is used as default. We define a function to indicate the node value of the probabilistic tree: *PT*(*super, sub*) represents the probability of *sub* to be a match, if *super*, the super-concept of *sub*, is a

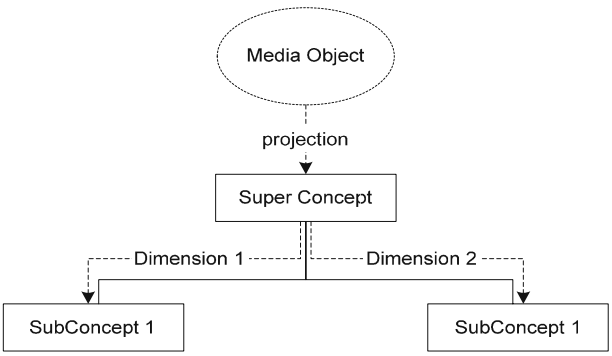


Figure 5 Projection in a sub-semantic space.

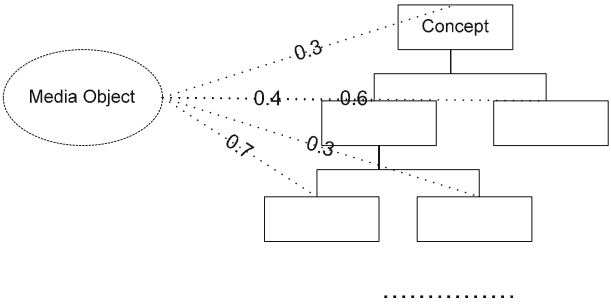


Figure 6 Probabilistic tree of a media object.

match of a media object. With the *Probabilistic Tree*, we can easily deduce the probability of a media object matching a certain concept. The analytical algorithm is presented below:

Procedure:

Step i. Following the thesaurus, trace from the target concept C_1 to the root concept *Root* in thesaurus. Assume the path is: $<C_1, C_2..., Root = C_n>$. Start from $CC = C_n$ and initially set P=1.

Step ii. Suppose $CC = C_i$, and the next concept C_{i-1} is one of the k sub-concepts of C_i. If CC is encoded in the *Probabilistic Tree* of this media object, then let $P=P*PT(C_i, C_{i-1})$. If not, we let $P = P * 1/k$.

Step iii. If CC has not reached C_1, repeat Step ii. Or, P is the probability of the media object matching concept C_1.

An implication behind this algorithm is, therefore, that the deeper a concept residing in the thesaurus, the less probability it is of to become a match of the media content. It would be arguable, however, that if a media object is returned as a result in both queries of concepts C_1 and C_2, it should be intuitively true that this media object has equal probability to match these two concepts. However, when we consider the construction of a media view, we only care of which media objects are most probably relevant to a specific concept. This implies, therefore, that the algorithm is reasonable in comparing the probability of different media objects matching a certain concept. The inference of *PT*(*super, sub*) will be presented later in Section 3.2.2.

3.2 MediaView customization

In the case of retrieving multimedia data, the semantics of a media object are relative to the user's goal and knowledge background [16, 33, 34]. The scenario depicted in Figure 7 shows how a media object is interpreted and classified to a concept. The fact that human cognition and social experience highly affects the understanding of media data suggests us that two key points should be considered in the design of *MediaView*, as follows.

– **Personalization**

Due to the different knowledge background and interest focus of users, a certain concept, especially abstract concept, may be considered as relevant to different groups of media objects by different users. So in *MediaView*, the user has to be supported with a strategy to reflect his specific interests, or namely, domain knowledge.

– **Generalization**

In another way, the personal knowledge background may also cause users to make wrong decision. In particular, a user may not choose the painting "sunflower" as relevant to "Van Gogh", if he doesn't have enough knowledge of that great artist. From this point of

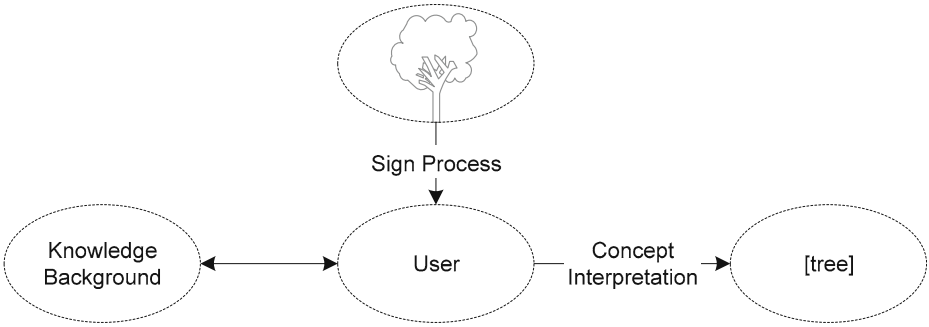


Figure 7 The process of media interpretation.

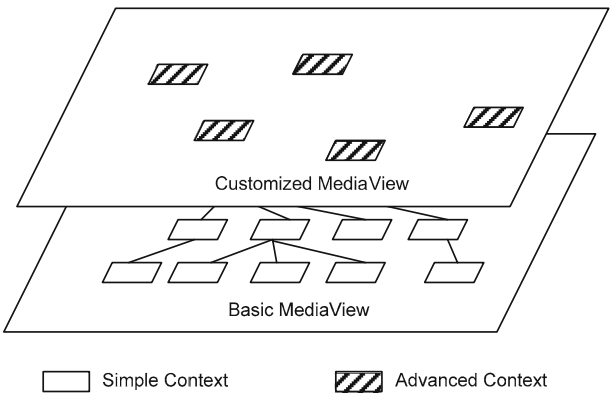
view, MediaView should be constructed properly to reflect the common knowledge of most users.

For the above consideration, we provide a two-level *MediaView* framework to tackle these issues (Figure 8). The first level is composed by common media views, which are permanent to the system and accumulated from the common knowledge of all users, due to our discussion before. The second level is thus for customized media views. User could generate customized media views on the base of common views to accommodate to specific tasks or applications of multimedia database.

3.2.1 Defining an advanced context

The advantage of *MediaView*, obviously, exists in that it avoids the invocation of the expensive media processing algorithm each time a query is processed, instead of which, it accumulates and learns the semantic knowledge among different queries, and provides quick responses to later queries on the multimedia database. However, this also results in a fact that the media views accumulated in database can not cover all of the queries a user could perform. In many cases, complex context may be given by users for preferred retrieval, for example, "the Great Hall in City University". We have indicated that media views associated with concepts should be accumulated and stored in the database for reuse. In this regard, our framework should provide a mechanism to users for constructing those

Figure 8 Two-level *MediaView* framework.



complex-context based media views, based on existed common media views dynamically. Therefore, several user-level operators are devised to support more complex context, besides the basic operators mentioned in the Appendix, as follows.

1. *INHERIT_MV(N: mv-name, NS: set-of-mv-refs, VP: set-of-property-ref, MP: set-of-property-ref): mv-ref*. This operator creates a *media view* named as *N*, which inherits the *media view set* indicated by *NS*. When executed successfully, it returns the reference to the created *media view*, which has all the members and relationships inherited from its super views.
2. *UNION_MV(N: mv-name, NS: set-of-mv-refs): mv-ref*. This operator creates a *media view* named as *N*, which unites the media data in the *media view set* indicated by *NS*. When executed successfully, it returns the reference to the created *media view*, which has all the media contents from the original views. From the point of context, it acts as an OR logic.
3. *INTERSECTION_MV(N: mv-name, NS: set-of-mv-refs): mv-ref*. This operator creates a *media view* named as *N*, which covers the common media data in the *media view set* indicated by *NS*. When executed successfully, it returns the reference to the created *media view*, which has the common media contents from all the original views. From the point of context, it acts as an AND logic.
4. *DIFFERENCE_MV(N1: mv-ref, N2: mv-ref): mv-ref*. This operator creates a *media view* named as *N*, which is the difference set of *N1* and *N2*. It covers the set of media objects as $\{m|m \in N1 \wedge m \notin N2\}$.

It can at best, however, provide a limited flexibility to define advanced context using existed simple contexts. Queries such as "the greatest artist" could not be deduced only from previous query results such as "artist"; in contrary, more high-level semantics of the media data in database should be modelled and provided for that query. As we have discussed earlier, to improve the performance of multimedia database with least additional manual work, and to act as a general-purpose mechanism, *MediaView* will not cover this kind of ability. Moreover, natural language processing (NLP) technology may be used to help model the query. However, it is also out of the scope of this paper.

3.2.2 Fuzzy logic based evolution

The *MediaView* evolution mechanism we propose is based on a progressive approach, which means the media views stored in database are accumulated along with the processes of user interaction. In particular, we have two kind of feedback could be utilized in MediaView evolution: *system-feedback* and *user-feedback*, as shown in Figure 9.

– *System-feedback*

As the main sources of the knowledge acquired by *MediaView*, the multi-retrieval systems become a feedback source to evolve the *MediaView* engine, with the retrieval results of each query. By analyzing each query performed by users, we know more about the semantics of retrieved media objects.

– *User-feedback*

Having been widely applied in IR technology, user-feedback shows some distinctive advantages: efficiency and correctness. By indicating the relevant and irrelevant results to the query, it gives the CBIR systems a chance to improve their performance of retrieval. However, in practice, users usually don't have enough patience to feedback all results, but only the first few of them.

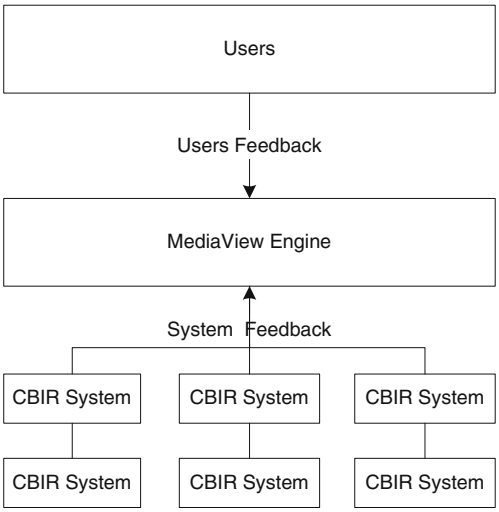


Figure 9 Two feedback sources of *MediaView*.

Hence, in a complete interaction session, the *MediaView* engine receives two phases of feedbacks, from the system side and the user side. That is, as users issue queries from the interface provided by MediaView engine, the engine firstly records the retrieved results from underlying CBIR systems; and then, it records the feedback from users, if available. More confidence exists in the feedback of users, for the reason that current CBIR systems are far from perfect. This raises the need to give different weight to the two kind of feedback. Initially, we set the confidence of each system as ζ_i=1 for weighing the accuracy of *i*-th CBIR system, and set the confidence of user feedback as υ=1. An adaptive algorithm, for adjusting the confidence of each system feedback gradually is suggested as below:

Procedure	
Step i.	Record each feedback performed by users.
Step ii.	For each CBIR system *i* involved, calculate its accuracy rate of retrieval. That is, simply divide the total number of retrieved results by the number of correct results according to user feedback.
Step iii.	Reset the value of ζ_i to its accuracy rate, respectively.
Step iv	Wait for next session of user feedback.

Due to the uncertainty of the semantic of media objects, we can't make an absolute assertion that a media object is relevant or irrelevant to a context. Because a media object in database may be retrieved as relevant result many times to a context, with the belief that more times a media object is considered relevant, the more confidence it has to be relevant to the context, we provide a mechanism to accumulate this effect. Consequently, we use fuzzy logic to describe the assertion of "relevant" or "irrelevant".

For a media object *e*, and a context *c*, $R_e(c)$ stands for the accumulation of historical feedback information, both of system and user feedback. Initially, we let $R_e(c) = 0$. Then, $\Delta Re(c)$ represents the adjustment of $R_e(c)$ after each feedback session, which is defined as:

$$\Delta R_e(c) = \begin{cases} \upsilon & , \text{if feedback}_{\text{u}} = \textit{''relevant''} \\ -\upsilon & , \text{if feedback}_{u} = \textit{''irrelevant''} \\ \sum_i \zeta_i & , \text{if feedback}_{\text{s,i}} = \textit{''relevant''} \\ -\sum_i \zeta_i & , \text{if feedback}_{\text{s,i}} = \textit{''irrelevant''} \end{cases}.$$

Hence, the confidence of e is relevant to c is defined as: $Confidence_e(c) = \frac{1}{(1+R_e^{-1}(c))} \in [0,1]$. To make sure $Confidence_e$ $(c)\varepsilon[0,1]$, we should keep $R_e(c)\geq 0$.

Consequently, we could now give the definition of probabilistic tree function $PT(super, sub)$ in Section 3.1.1 as follows: $PT(super, sub) = \frac{Confidence(sub)}{\sum_i Confidence(i)}$, where i is any sub-concept of *super*.

The up-down fashion, as described in Section 3.1.1, of calculating the probability of a media object matching a certain concept, though intuitively, has the drawback that the lower concept can not affect the upper concept. That is, if we calculate the probability of a media object matching an upper concept, say, "season", we can not leverage the historical information that the media object was a match of some sub-concept, say, "spring", due to the up–down order for calculating. Hence, there is a need, to propagate the confidence of a media object relevant to a concept along the hierarchical structure from bottom up, based on the fact that if a media object is selected to be a match of a sub-concept, it is certainly a match to all of the super-concepts. For example, if a feedback shows the media object is relevant to "spring", then it will give more confidence to be relevant to "season", which is the super concept of "spring". The inverse propagation algorithm is given as follows.

Procedure:

Step i. Wait for a feedback session.

Step ii. For each positive feedback, namely, stating a concept C is relevant to a media object. Following the thesaurus, trace from C to the root concept *Root* in thesaurus. Assume the path is: $<C_1, C_2..., Root = C_n>$.

Step iii. Append Ci as also positive feedback to that media object, where $i= 1$ *to* n.

4 *MediaView* utilization

To show the usefulness and elegancy of *MediaView*, we introduce a real-world application in which media views are found to be a natural and suitable modeling construct. The application comes from our on-going research project on a multi-modal information retrieval system, *Octopus* [32]. In this section, we describe several specific media views created as the data model of *Octopus*. To cater for the requirements of different problem domains, we discuss three main cases of problem for utilizing MediaView mechanism: *multimedia database navigation, document authoring* and *personalized retrieval with relevance feedback*.

4.1 Data model

Octopus is proposed to provide search functionality in multimedia repositories ranging from web to digital libraries, where data are typically of multiple types of modality. The basic search paradigm supported by *Octopus* is query-by-example, that is, a user forms a query by designating a media object as the sample object and the system retrieves all the media objects relevant to it. For example, using the poster (an image) of the movie "Harry Potter" as the sample, we expect to receive media objects such as a textual introduction of the movie, a "highlight" video clip, and the music of the movie. Essential to such a multi-modal retrieval system is the relevance between any two media objects, which is evaluated from the following three perspectives:

1. *User perceptions*. Two media objects are regarded as relevant if users have the same/similar interpretation of them, e.g., annotating them with the same keywords.

2. *Contextual relationship*. Media objects that are spatially adjacent or connected by hyperlinks are usually relevant to each other.
3. *Low-level features*. Low-level features (e.g., color of images) can be extracted from media objects to describe their visual/aural characteristics. Intuitively, media objects are considered relevant if they possess highly similar low-level features.

As shown in Figure 10, a media view called *KB* is created to model the relevance between any two media objects in the database of *Octopus*. The members of *KB* are media objects such as images, videos, audios, which are modelled as instances of heterogeneous source classes (cf. Figure 3). Three types of relationships (*perceptual*, *contextual*, and *feature*) are defined to represent the inter-object relevance from the aforementioned three perspectives. A weight can be associated with each relationship as its property to indicate the strength of the relevance.

KB provides an integrated knowledge base on the relevance among media objects, based on which user queries can be processed by analysing the various relationships contained in it. For each query, a media view named *Result(n)* is created to accommodate the results of the query, where *n* is the serial number. As shown in Figure 10, the global aspect of the query is described by its view-level properties, such as the sample object used, while member-level properties are assigned on each object to describe its characteristics as a query result, such as its relevance score, and users' feedback opinion towards it (relevant, neutral, or irrelevant).

4.2 Navigating the database via MediaView

With a well designed MediaView engine, it turns to be very easy for navigating the multimedia database. Since the media views accumulated in database correspond to the concepts in WordNet, the six semantic relationships mentioned in Table 1, such as *Meronymy*, *Troponomy*, *Entailment*, could be utilized to browse from one media view to another related view, as depicted by Figure 11.

Users could posit queries, for example, by selecting an existing media view from the semantic tree, or by building their own views to reflect specific intentions. Whether the personalized media view is permanent (thus could be shared with other users) or transitory may be decided by users themselves.

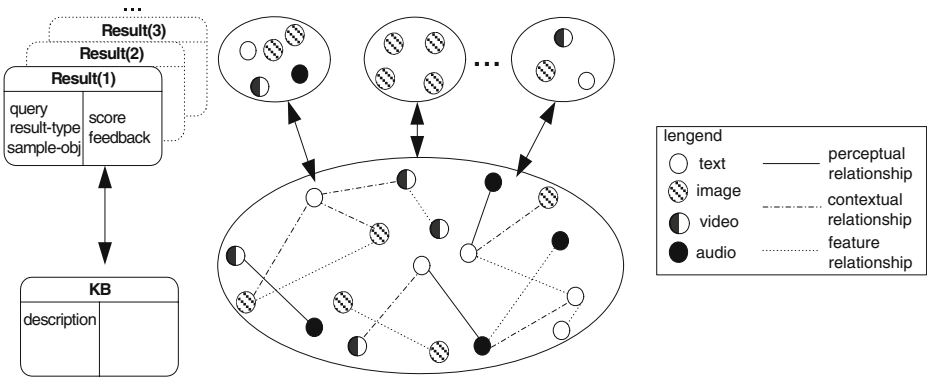


Figure 10 Media views created for *Octopus*.

Table 1 Semantic relationships in WordNet.

Semantic relationship	Examples
Synonymy (similar)	Pipe, tube
Antonymy (opposite)	Fast, slow
Hyponymy (subordinate)	Tree, plant
Meronymy (part)	Chimney, house
Troponomy (manner)	March, walk
Entailment	Drive, ride

4.2.1 Run-time derivation

Let us describe a scenario that demonstrates the navigation of multimedia database, with *MediaView* support. In this example, user holds interests in the famous artist "Van Gogh".

- *Who is "Van Gogh"?*
  Set vg = INHERIT_MV("V. Gogh", {<painter>},name = "Van Gogh",);
- *What's his work?*
  Set vg_work = INTERSECTION_MV("work", {<painting>, vg});
- *Know more about his life.*
  Set vg_life = INTERSECTION_MV("life", {<biography>, vg});
- *Know more about his country.*
  Set vg_coun = INTERSECTION_MV("country", {<country>, vg});
- *See his famous painting "sunflower".*
  Set sunflower = INTERSECTION_MV("sunflower", {<sunflower>, <painting>});
- *Set vg_sunflower = INTERSECTION_MV("vg_sunflower", {vg_work, sunflower});*
- *Any other famous painters other than Van Gogh?*
  Set other_pt = DIFFERENCE (<painter>, vg);

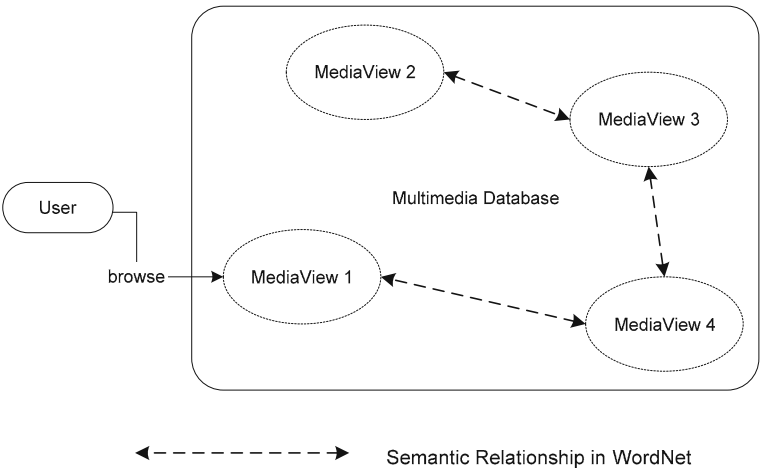


Figure 11 Navigating the database via media views.

The above sample shows how to navigate the database with *MediaView* operators. Admittedly, more complex navigation, say, "*Other painters in Van Gogh's time*" may not be carried out by only using *MediaView*'s knowledge. It requires more advanced ontology modelling to be incorporated into *MediaView*, which is however outside the scope of this paper.

4.3 Data integration

Developing reusable architectures is an important development in the field of multimedia application. Whereas primary emphasis has been placed on media processing, given the complexity of media analysis, it would be beneficial to address the issues of developing frameworks to support integration of multimedia data. Due to the ability of associating multi-model media data into a context, *MediaView* provides an effective and natural way to integrate those multimedia data in databases for application. Taken as an example, multimedia document authoring is one of the distinctive applications facilitated by the *MediaView* framework.

4.3.1 Multimedia document authoring

When authoring a multimedia document, users will encounter the problem of finding enough theme-relevant media materials. From this point of view, multimedia document authoring could be greatly enhanced by leveraging *MediaView*, more specifically, by retrieving the theme relevant media materials from database easily, as Figure 12 shows.

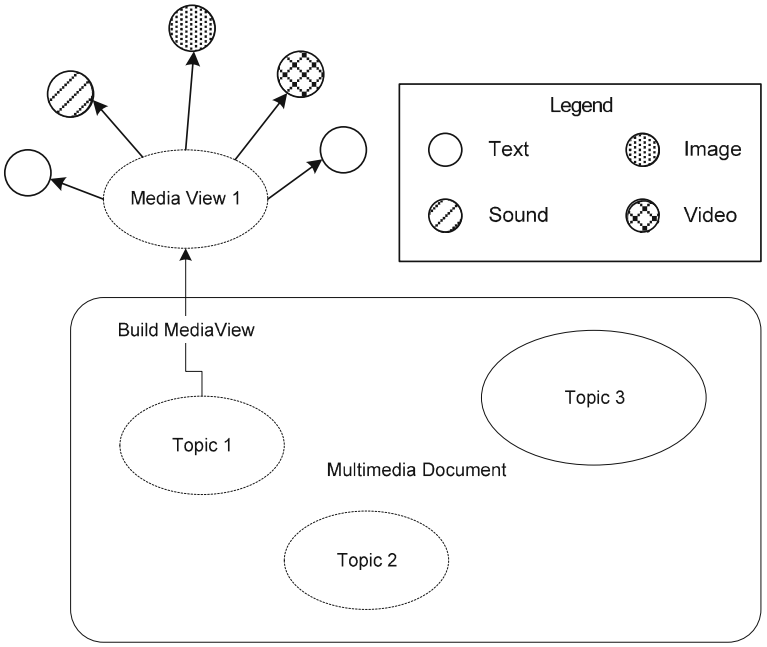


Figure 12 Multimedia document authoring with media views.

Here, we propose an application-architecture for enhancing the intelligence of multimedia document authoring system. Figure 13 illustrates how intelligence could be presented. The system is composed of two main modules: script engine and MediaView engine.

– *Script Engine*

This module accepts a user-defined script, describing the content of the multimedia document, and synthesizes the main contexts of the document with the aid of NLP technology.

– *MediaView Engine*

This module builds media views on the base of results from script engine. With media views, user could browse and choose from the collection of media objects relevant to the theme context for future authoring.

4.4 Retrieval with relevance feedback

As described in Section 4.1, a specific media view called *KB* has been created to model the relevance between any two media objects in the *Octopus* database (cf. Figure 10). In addition, to accommodate the results of each query, a media view named *Result(n)* is created where *n* is the serial number. The global aspect of the query is described by its view-level properties, such as the sample object used, while member-level properties are assigned on each object to describe its characteristics as a query result, such as its relevance score, and users' feedback opinion towards it (relevant, neutral, or irrelevant). For the latter, the feedback process is essentially an improvement of the original query point q by moving it towards the points representing relevant documents and away from the points of irrelevant ones. To facilitate relevance feedback, a unique two-levelled profiling approach has been used, in which both the *common profile* and *user profile* provide the semantic associations of all the media objects (such as text, images, videos) in the database. The distinction lies in that the common profile represents what most people agree upon (e.g. what is the meaning of an image), while the user profile represents what a specific user thinks (about the meaning of the image). Consequently, the common profile, shared by all

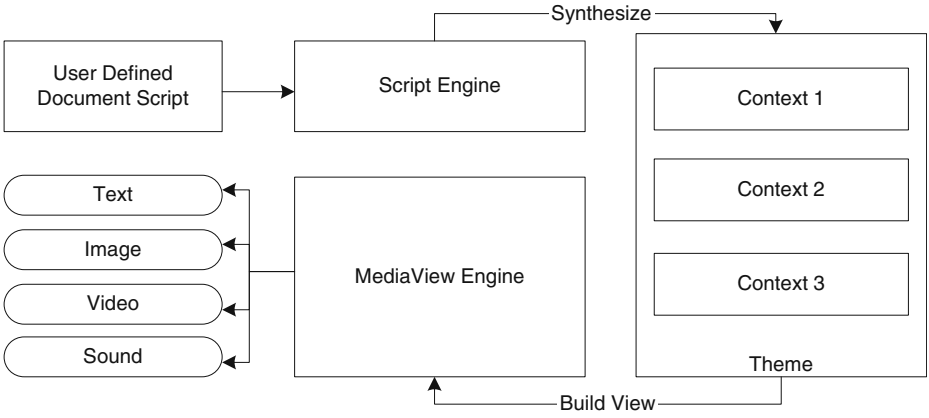


Figure 13 Architecture of multimedia document authoring system.

the users, is unique in the system and stored in *KB*, yet a user profile is created for each user and accessible only to this user through a specific *Result(n)* that she/he creates.

4.4.1 Performance of common profile

To demonstrate the utility and effectiveness of these "profiles" embedded inside the media views, some preliminary but operational experiments are devised, which simulate the Web environment with the help of Corel Image Gallery and some "virtual" users. In particular, we select 5,000 images from Corel as the test data, which are pre-classified into 50 categories with exactly 100 images in each category. Each category has a title that can be used to describe all the images within the category. Therefore, if the title is used as a query, all the images within the corresponding category are regarded as the relevant results to the query.

We simulate the behaviors of real users by creating some "virtual" users, who can perform queries and feedbacks automatically. A virtual user starts by searching for the images of a particular category in the database, using the title of the category as the query. The first 100 images with highest similarity to the query are returned by the system. The virtual user then randomly choose some images from the top 100 for evaluation, by marking it as relevant if it belongs to the intended category, or otherwise as irrelevant. Based on these user evaluations, the system updates the profiles and performs the feedback to improve the retrieval results. The loop of evaluation and feedback may repeat for more iteration. The number of images evaluated by a user in each round of feedback is set to 25 in our experiments, since a real user is unlikely to make evaluations more than that.

The performance of the common profile is examined on the aspect of how fast it can be learned from user feedbacks. In our experiment, the learning rate of the common profile is measured through its coverage in the database, interpreted as the percentage of images that are annotated with the title of the corresponding category in the common profile. To estimate this learning rate, the common profile is firstly clear and all the user profiles are disabled. Then, for each category we activate a virtual user to execute one retrieval operation (totally 50 operations), with 10 loops of feedback in each operation. The percentage of annotated images at each loop of feedback is recorded for each category. The average profile coverage (percentage of annotated images) over 50 categories against the number of feedback loops is plot in Figure 14. As we can see, the coverage of the common profile increases steadily with the feedbacks, reaching about 50% after 12 feedback loops.

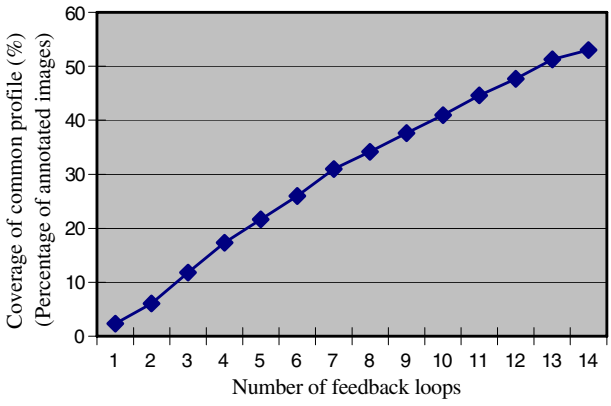


Figure 14 Learning rate of the common profile.

The six curves shown in Figure 15 illustrate the relationship between the retrieval accuracy and the coverage of the common profile. Each curve is obtained by tracing the retrieval precision after each loop of feedback at a certain level of profile coverage. Clearly, the common profile with a large coverage greatly helps to enhance the retrieval accuracy, especially at the starting several loops of feedback. On the other aspect, higher retrieval accuracy usually encourages users to make more evaluations, which in turn enlarges the profile coverage. Therefore, it forms a "self-reinforcing" loop between profile coverage and retrieval accuracy. The retrieval accuracy of our system is inferior to that reported by Lu *et. al.* [17], which evaluated the performance of keyword propagation in a similar experiment setting. However, we argue that this is due to the different number of evaluations made in each round of feedback: we give only 25 evaluations, while they make 100.

4.4.2 Performance of user profile

The performance of user profiles is more challenging, because the behavior of a particular user is usually unpredictable. To evaluate its performance, we use the same test data collection as the above, but further dividing the category of "car" into four sub-categories according to the color of the car (red, black, white or yellow), with 25 images in each sub-category. We assume that the user who queries for "car" actually targets at a specific sub-category of car, so that the retrieval precision is calculated as the percentage of the images belonging this sub-category among the top 25 images in the returned list. We compare the average precision of 8 random queries when a user profile is present to the case when it is absent, at different levels of the common profile coverage. In the case of using a user profile, some relevant and irrelevant images are manually "inserted" into the user profile. The results are shown in Figure 16. The precision when using a user profile is considerably higher than that without using it, at any coverage level of the common profile.

Unlike in Figure 15, we do not examine the change of retrieval precision during the feedback process in Figure 16. In contrast, we focus on the precision of original queries, because a user profile is effective mainly in the original query by providing some "pseudo" feedback examples that can adjust retrieval results towards a particular user's interest.

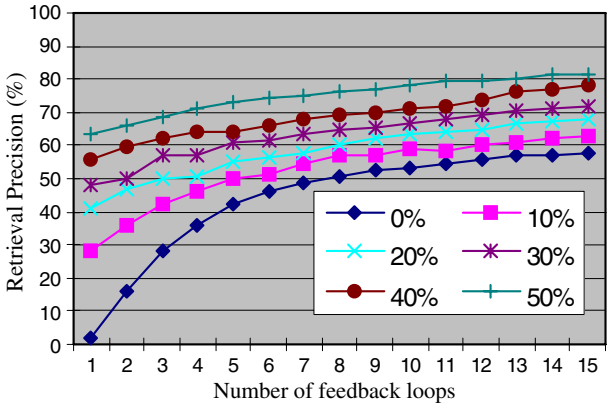


Figure 15 Retrieval precision at each level of profile coverage.

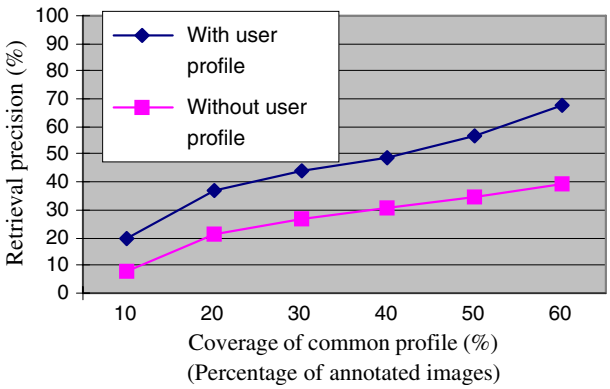


Figure 16 Performance of user profiles.

5 Comparisons to related work

In this section, we present a review on the data modeling techniques related to semantic modeling of multimedia, which are roughly classified into three categories, as (1) multimedia database techniques, (2) previous object-oriented view mechanisms, (3) MPEG-7, and (4) cross-media retrieval.

5.1 Multimedia database techniques

The proliferation of multimedia data imposes a great challenge on conventional database technology, which is inadequate to model their characteristics, such as the huge data size, spatial and temporal nature, etc. To address these limitations, much work has been proposed towards different aspects regarding the management of multimedia, including data models [2, 16, 20], presentation [14], indexing [7], and query processing [5]. Among all the data models, the object-oriented approach is generally regarded as the most suitable choice for modeling multimedia data, mainly because of its great modeling capacity and its extensibility (for new types of data) by means of inheritance. For example, OVID (Object Video Information Database) system [20] was proposed based on an object data model for video management. Similarly, the STORM [2] object-oriented database management system integrates structural and temporal aspects for managing different presentation of multimedia objects. The ORION project [13] proposed by Kim *et. al.* also adopts object-oriented methodology for multimedia management. Although these projects are successful in their respective domains, they fall short of modeling the semantic aspects of multimedia data, which is of vital importance to multimedia information system (MMIS) applications, which is exactly the focus of our proposed MediaView mechanism.

5.2 Object-oriented view mechanism

There have been many successful previous research projects on view mechanism in object-oriented data models. Most of them utilized the query language defined for their respective object model to derive a view (or virtual class); e.g., Abiteboul et al. [1] proposed a general framework for view definition based on clear semantics. These approaches differ from each other in the way that they treat the derived view in the global schema. For example, Heiler's approach [10] treats each view as a standalone object, rather than integrating into the

schema. In Kim's work [12, 13], the derived views are attached to the schema root as its direct subclasses. The approaches of Scholl et al. [23] and Tanaka et al. [27] address the issue of incorporating the derived views into the global schema. However, the consistency of the schema is not guaranteed in their approaches. Rundensteiner's proposal of MultiView [21] is a more systematic solution on object view mechanism, which not only provides an algorithm for integrating the derived virtual classes into the global schema, but also allows the generation of multiple view schemata, with the consistency and closure property enforced by automatic tools. The issue of view materialization is also addressed in MultiView [21]. Yet, the fundamental distinction between *MediaView* and the past work is that all the existing view solutions are "information customisation" mechanisms, i.e., hiding properties and instances of the base classes from a customized view, while *MediaView* is intended to be an "information augmentation" mechanism, in which objects are empowered with new properties when they are included into the media views.

5.3 MPEG-7

MPEG-7, as a standard for specifying "multimedia content description interface" developed by ISO and IEC [11], is arguably a candidate of "data model" for a multimedia database (MMDB). The objective of MPEG-7 is to provide a rich set of standardized "tools" for describing multimedia content [24]. The specific aims of MPEG-7 are mainly in two areas: 1) to standardize multimedia data components and their structure, and 2) to standardize the language to specify multimedia data descriptions. In order to adopt different levels of abstraction, MPEG-7 aims at establishing a flexible and extensible framework for defining descriptions about the multimedia contents. Therefore, what MPEG-7 defines is a set of methods and tools for different aspects of multimedia materials, and leaves space for other parties to use the framework in more specific areas and application domains. From this point of view, MPEG-7 is similar to our MediaView mechanism. To achieve the flexibility and extendibility, MPEG-7 chooses XML [29] as the language for the textual representation of content descriptors, descriptor schemes and DDL. For the semantic information in MPEG-7 metadata, however, it usually relies on the human interpretation and manual input, which is truly a labour intensive work. So even though MPEG-7 is useful for the indexing and searching of multimedia data and can be integrated into MMDBs, the construction of the semantic level descriptions and the building of the semantic relationships have a big room for improvement.

As one of the main interests in using a MMDB is to search for "similar" data, a MMDB system should be able to deal with queries for similarity searches, preferably with relevance feedback support. In this regard, SQL/MM [6] as part of the SQL-3 (SQL:1999) standard [19], introduces for MMDB systems a conceptual multimedia data model that extends the concepts of the object-relational SQL3. Compared with MPEG-7, the data model of SQL/MM covers the syntactical part of multimedia descriptions but allows no means of decomposing an image for describing its semantically meaningful content. By using MPEG-7 as a "data model" in MMDB systems, however, one should think about the enhancement of the query language aspect in terms of similarity search. In addition, the operations to produce XML output have to be considered as well. If we store the whole XML document in the database, it will be easy to respond to a user query by passing back the XML document stored in the database. However, the maintenance of the XML document imposes challenges to this method as the system will need to retrieve the XML document from the database, decompose it and then reconstruct it after altering the changed values. If the MMDB does not store the whole XML document, it means that one has to

combine a multimedia query language (e.g., SQL/MM) with XML elements. In addition, it has to ensure that the resulting XML document satisfies the XML Schema for MPEG-7, which necessitates the enhancement of query processing with type checking for MPEG-7 conformance. Furthermore, neither SQL/MM nor MPEG-7 provides support for user relevance feedback—a feature which is emphasized by MediaView together with the support of (multi-level) user profiling techniques.

5.4 Cross media retrieval

In a sense, most existing multimedia retrieval methods are not genuinely for "multi-media", but for a specific type (or modality) of non-textual data. There is, however, the need to design a real "multi-media" retrieval system that can handle multiple data modalities in a cooperative framework. First, in multimedia databases like the Web, different types of media objects co-exist as an organic whole to convey the intended information. Naturally, users would be interested in seeing the complete information by accessing all the relevant media objects regardless of their modality, preferably, from a single query. For example, a user interested in a new car model would like to see the pictures of the car and meanwhile read articles on it. Sometimes, depending on the physical conditions such as networks and displaying devices, users may want to see a particular presentation of the information in appropriate modality(-ies). Furthermore, some data types such as video intrinsically consist of data of multiple modalities (audio, closed-caption, video images). It is advantageous to explore all these modalities and let them complement each other in order to obtain better retrieval effect. To sum up, a retrieval system that goes across different media types and integrates multi-modality information is highly desirable.

Informedia [9] is a well-known video retrieval system that successfully combines multi-modal features. Its retrieval function not only relies on the transcript generated from a speech recognizer and/or detected from overlaid text on screen, but also utilizes features such as face detection and recognition results, image similarity, etc. Statistical learning methods are widely used in Informedia to intelligently combine the various types of information. There are many other systems that integrate features from at least two modalities for retrieval purpose. For example, WebSEEK system [25] extracts keywords from the surrounding text of image and videos in web pages, which is used as their indexes in the retrieval process. Although the systems involve more than one media type, typically textual information plays the vital role in providing the (semantic) annotation of the other media types. Other examples include the MediaNet [4] and multimedia thesaurus (MMT) [28], both of which seek to provide a multimedia representation of semantic concept-a concept described by various media objects including text, image, video, etc—and establish the relationships among these concepts. MediaNet extends the notion of relationships to include even perceptual relationships among media objects. More recently, in [30], cross-media retrieval is articulated to break the limitation of modalities of media objects. As an extension of multi-modality retrieval [32], cross-media retrieval can be regarded as a unified multimedia retrieval paradigm by learning some latent semantic correlation between different types of media objects.

6 Conclusion

The *MediaView* mechanism presented in this paper builds a bridge across the "semantic gap" between conventional databases and multimedia applications, the former of which are

inadequate to capture the dynamic semantics of multimedia, whereas data semantics plays a key role in the latter. This mechanism is based on the modelling construct of a *media view* which formulates a customized context, inside which *heterogeneous* media objects with related semantics are characterized by *additional properties* and *semantic relationships*. View operators have been developed for the manipulation of media views. The application of *MediaView* in a multi-modal information retrieval system has been described to demonstrate its usefulness.

In this paper, we have discussed the research issues on the implementation, evolution and utilization of *MediaView* framework. *MediaView* is designed as an extended object-oriented view mechanism to bridge the semantic gap between conventional database and semantics-intensive multimedia applications. We have provided a set of user-level operators to enable users to accommodate the specialization and generalization relationships among the media views. Users could customize specific media views according to their tasks, by using user-level operators. We have also shown the effectiveness of using *MediaView* in the problem domain of multimedia navigation and data integration, and the efficiency of accommodating similarity retrieval with profile-based relevance feedback through experiments.

Many open research issues remain in this direction to make this technology pervasive and useful. A key challenge will be the development and transition of *MediaView* to a fully-fledged multimedia database system. Moreover, advanced semantic relations such as temporal and spatial relations should be incorporated in combining media views. If successful, *MediaView* framework can improve multimedia information retrieval in two principal ways. First, it promises more effective access to the content of a media database. Users could get the right stuff and tailor it to the *context* of their applications easily. Second, by providing the most relevant content from pre-learnt semantic links between media and context, high performance database browsing, multimedia authoring, and personalized similarity retrieval can be provided and offered to the end-users in their comprehensive applications.

Acknowledgement We would like to acknowledge Mr. Dawei Ding and Mr. Jun Yang (of Zhejiang University then) for their contributions in developing the experimental studies reported in Section 4. The material presented in this paper is based upon the work funded by Zhejiang Provincial Natural Science Foundation of China under Grant No.Y107750. The work has also been supported by the Natural Science Foundation of China with the project number 60773197.

Appendix: Basic MediaView Operators

The set of view operators[1] defined below provides the basic functions of media views, while more sophisticated operations can be implemented as a combination of these basic ones. For example, a search for objects that are related with a specific object in any media view can be handled by applying *GET-ALL-MV()* and *GET-RELATED-MEM ()* in a combined fashion.

– *CREATE-MV (N: mv-name, VP: set-of-property-ref, MP: set-of-property-ref): mv-ref.* This operator creates a media view (*MV*) named as *N*, which takes the properties in *VP*

[1] In the definition of view operators, the suffix "-ref" represents the reference to object, which is actually a variable holding the *Oid* of an object. For example, *mv-ref* is the reference to a media view, *relationship-ref* is the reference to a relationship, etc.

as its view-level properties, and those in *MP* as its member-level properties. When executed successfully, it returns the reference to the created media view, which has no members and relationships initially.

- *cDELETE-MV (MV: mv-ref)*. This operator deletes a media view specified by *MV* from the database. All the members of *MV*, their properties (value) defined in *MV*, and all the relationships in *MV* are also deleted. Note that the member itself as an instance of its source class is not deleted from the database.
- *GET-ALL-MV():set-of-mv-ref*. This operator retrieves all the media views currently in the database. The return value is a set of references to these media views.
- *ADD-MEM (MV: mv-ref, O: object-ref)*. This operator adds the object referred by *O* as a member of the media view referred by *MV*. All the member-level properties for *O* are set to their default values.
- *REMOVE-MEM (MV: mv-ref, O: object-ref)*. This operator excludes the object *O* from the media view *MV*, with all its relationships and properties in *MV* deleted.
- *ADD-RELATION (MV: mv-ref, O1: object-ref, O2: object-ref, R: relationship-type): relationship-ref*. This operator establishes a relationship of type *R* between objects *O1* and *O2*, which are the members of the media view *MV*. If the operator is applied successfully, the reference to the relationship object is returned.
- *REMOVE-RELATION (MV: mv-ref, O1: object-ref, O2: object-ref[, R: relationship])*. If the last argument is not specified, this operator removes all their relationship(s) between objects *O1* and *O2* in the media view *MV*. Otherwise, it only deletes the relationships of the type specified by *R*.
- *GET-ALL-MEM (MV: mv-ref): set-of-object-ref*. This operator retrieves all the (heterogeneous) objects as the members of the media view *MV*.
- *HAS-MEM (MV: mv-ref, O: object-ref)*: *boolean*. This operator tests if object *O* is a member of the media view *MV*.
- *GET-RELATED-MEM (MV: mv-ref, O: object-ref[, R: relationship]): set-of-object-ref*. This operator returns all the objects that have relationship of any type (if the last argument is absent) or of type *R* (if the last argument is given) with object *O* in the media view *MV*.
- *GET-ALL-RELATION (MV: mv-ref): set-of-relationship-ref*. This operator retrieves all the relationships in the media view *MV*.
- *GET/SET-VIEW-PROP (MV: mv-ref, P: property-ref): value*. This operator retrieves (or sets) the value of the view-level property *P* of media view *MV*.
- *GET/SET-MEM-PROP (MV: mv-ref, O: object-ref, P: property-ref, V: value)*. This operator retrieves (or sets) the value of the member-level property *P* of object *O* in media view *MV*.

References

1. Abiteboul, S., Bonner, A.: Objects and Views. In Proc. of ACM SIGMOD Conf. on the Management of Data, 238–247 (1991)
2. Adiba, M.: STORM: Structural and Temporal Object-Oriented Multimedia Database Systems. IEEE Int. Workshop on Multimedia DBMS, NY, USA, August, (1995)
3. Apers, P., Blanken, H., Houtsma, M. (eds.): Multimedia Databases in Perspective. Springer, London (1997)
4. Benitez, A.B., Smith, J.R., Chang, S.F.: MediaNet: A Multimedia Information Network for Knowledge Representation". Proceeding of the SPIE 2000 Conference on Internet Multimedia Management Systems, **4210**, (2000)

5. Bertino, E., Catania, B., Ferrari, E.: Query processing. In: Apers, P.M.G., Blanken, H.M., Houtsma, M. A.W. (eds.) Multimedia Databases in Perspective. vol 181–217Springer, Berlin/Heidelberg (1997)
6. Eisenberg, A., Melton, J.: SQL Multimedia and Application Packages (SQL/MM). ACM SIGMOD Record **30**(4), Dec. 2001
7. Faloutsos, C.: Indexing of multimedia data. In: Burkhard, W.A., Keller, R.M. (eds.) Multimedia Databases in Perspective. vol 219–245Springer, Berlin/Heidelberg (1997)
8. French, J.C., Watson, J.V.S., Jin, X., Martin, W.N.: Integrating Multiple Multi-channel CBIR Systems. International Work Shop on Multimedia Information System (MIS'03), Ischia, Italy, May 2003
9. Hauptmann, A., et al.: Video Classification and Retrieval with the Informedia Digital Video Library System. Text Retrieval Conference (TREC02), Gaithersburg, MD, 2002
10. Heiler, S., Zdonik, S. B.: Object views: Extending the vision. Proc. of Int'l Conf. on Data Eng. (ICDE'90), pp. 86–93, Feb, 1990, IEEE
11. ISO/IEC, MPEG-7 Overview: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm (2006)
12. Kim, W.: Object-Oriented Database Systems: Promises, Reality, and Future. Proc. of 19th Very Large Database, 676–687 (1993)
13. Kim, W., Garza, J.F., Ballou, N., Woelk, D.: Architecture of the ORION Next-Generation Database System. IEEE Trans. Knowl. Data Eng. **2**, (1), 109–124 (1990)
14. Klas, W., Bool, S., Lohr, M.: Integrated database services for multimedia presentation. Multimedia Information Storage and Management. Kluwer Academic, USA (1996)
15. Li, Q., Özsu, M.T.: Editorial: Introduction to web media information systems (special issue on web media information systems). World Wide Web. **5**, (3), 179–180 (2002)
16. Li, Q., Yang, J., Zhuang, Y. *MediaView*: A Semantic View Mechanism for Multimedia Modeling. Proceedings of the Pacific Rim Conference on Multimedia (PCM'02), pp. 729–736, IEEE, 2002.
17. Lu, Y., Hu, C.H., Zhu, X.Q., Zhang, H.J., Yang, Q. A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems. In: Proceedings of the 8th ACM Multimedia conference, 2000
18. Miller, G.A.: WordNet: A lexical database for English. Comm. ACM. **38**, (11), 39–41 (1995) Nov
19. NCITS H2, SQL:1999, available from: http://www.service-architecture.com/database/articles/sql1999.html
20. Oomoto, E., Tanaka, K.: OVID: Design and implementation of a video-object database system. In IEEE Trans. Knowl. Data Eng. **5**, 629–641 (1993)
21. Rundensteiner, E. A.: MutiView: A Methodology for Supporting Multiple Views in Object-Oriented Databases. Proc. of the 18th Conf. on Very Large Database, Canada (1992)
22. Santini, S., Jain, R.: Interface for emergent semantics in multimedia database. Proceedings of the IS&T/SPIE Conference on Storage and Retrieval for Image and Video Database 167–175 (1999)
23. Scholl, M.H., Lassch, C., Tresch, M.: Updateable Views in Object-Oriented Databases. Proceedings of the 2nd DOOD Conference, Germany, Dec. 1991
24. Smith, J.: MPEG-7 multimedia content description standard. In: Feng, D., Siu, W.C., Zhang, H.J. (editors), Multimedia Information Retrieval and Management, Chap. 6, Springer, (2003).
25. Smith, J.R., Chang, S.F.: Visually searching the Web for content. IEEE Multimedia. Magazine. **4**, (3), 12–20 (1997)
26. Song, Y., Wang, W., Zhang, A.: Automatic annotation and retrieval of images. World. Wide. Web. **6**, (2), 209–231 (2003)
27. Tanaka, K., Yoshikawa, M., Ishihara, K.: Schema Virtualization in Object-Oriented Databases. Proceedings of the Int'l Conf. on Data Engineering (ICDE'88), 23–30, Feb, IEEE (1988)
28. Tansley, R.: The Multimedia Thesaurus: An Aid for Multimedia Information Retrieval and Navigation. Master Thesis, Computer Science. University of Southampton, UK (1998)
29. W3C, Extensible Markup Language (XML) 1.0 (Fourth Edition): http://www.w3.org/TR/REC-xml/ (2006)
30. Wu, F., Zhang, H., Zhuang, Y-T.: Learning Semantic Correlations for Cross-Media Retrieval. ICIP:1465–1468, 2006
31. Yang, J., Li, Q., Liu, W., Zhuang, Y.: Searching for flash movies on the Web: A content and context based framework. World Wide Web **8**, (4), 495–517 (2005)
32. Yang, J., Li, Q., Zhuang, Y.T.: Octopus: Aggressive Search of Multi-Modality Data Using Multifaceted Knowledge Base. Proceedings of the 11th International Conference on World Wide Web (WWW'02):54–64 (2002)
33. Zakos, J., Verma, B.: A novel context-based technique for Web information retrieval. World Wide Web **9**, (4), 485–503 (2006)
34. Zhang, H., Chen, Z., Li, M., Su, Z.: Relevance feedback and learning in content-based image search. World Wide Web **6**, (2), 131–155 (2003)