Published as a chapter in *Advances in Multimedia: Systems, Standards, and Networks,* A. Puri and T. Chen (eds.). New York: Marcel Dekker, 1999.

Multimedia Search and Retrieval

Shih-Fu Chang[1], *Qian Huang*[2] , *Thomas Huang*[3], *Atul Puri*[2] , *and Behzad Shahraray*[2]

1. Columbia University, Department of Electrical Engineering, New York, NY 10027. sfchang@ee.columbia.edu
2. AT&T Labs – Research, 100 Schulz Dr., Red Bank, NJ 07701. huang@research.att.com, apuri@research.att.com, behzad@research.att.com
3. University of Illinois at Urbana-Champaign, Dept. of Electrical and Computer Engineering, Urbana, IL 61801. huang@ifp.uiuc.edu

1. Introduction

Multimedia search and retrieval has become an active research field thanks to the increasing demand that accompanies many new practical applications. The applications include large-scale multimedia search engines on the Web, media asset management systems in corporations, audio-visual broadcast servers, and personal media servers for consumers. Diverse requirements derived from these applications impose great challenges and incentives for research in this field.

Application requirements and user needs often depend on the context and the application scenarios. Professional users may want to find a specific piece of content (e.g., an image) from a large collection within a tight deadline, while leisure users may want to browse the clip art catalog to get a reasonable selection. On-line users may want to filter through a massive amount of information to receive information pertaining to their interests only, while off-line users may want to get informative summaries of selected content from a large repository.

With the increasing interest from researchers and application developers, there have been several major publications and conferences dedicated to survey of important advancements and open issues in this area. Given the dynamic nature of applications and research in this broad area, any survey paper is also subject to the risk of being incomplete or obsolete. With this perspective, we focus this chapter on several major emerging trends in research, as well as standards related to the general field of multimedia search and retrieval. Our goal is to present some representative approaches and discuss issues that require further fundamental studies.

Section 2 addresses a promising direction in integrating multimedia features in extracting the syntactic and semantic structures in video. It introduces some domain-specific techniques (*i.e.*, news) combining analysis of audio, video, and text information in analyzing content at multiple levels. Section 3 focuses on a complementary direction in which visual objects and their features are analyzed and indexed in a comprehensive way. These approaches result in search tools that allow users make direct manipulation visual content to form multimedia queries. Section 4 shows a new direction, incorporating knowledge from machine learning and interactive systems, to break the barriers of decoding semantics from multimedia content. Two complementary approaches are presented: probabilistic graphic model and semantic template. Section 5 covers an important trend in the multimedia content description standard, MPEG-7, and its impact on several applications such as interoperable meta search environment.

2. Video Segmentation, Indexing, and Browsing

As discussed in the previous section, different methods are suitable for each context when people access large collections of information. Video content has unique characteristics that further affect the role of each access method. For example, the sequential browsing method may not be suitable for long video sequences. In this case, methods using content summaries, such as those based on the Table of Contents (ToC), are very useful in providing quick access to structured video content.

Different approaches have been used to analyze the structures of video. One technique is to do it manually as in the cases of books (ToC) or broadcast news (closed captions) delivered by major American national broadcast news companies. Since manual generation of an index is very labor intensive, and thus, expensive, most sources of digital data in practice are still delivered without meta information about content structures. Therefore, a desirable alternative is to develop automatic or semi-automatic techniques to extract semantic structure from the given linear data of video. The particular challenges we have to address include identification of the semantic structure embedded across multiple media, and discovery of the relationship among the structures across time and space (so that higher levels of categorization can be derived to further facilitate automated generation of a concise index table).

Typically, to address this problem, a hierarchy with multiple layers of abstractions is needed. To generate this hierarchy, data processing in two directions has to be performed. First, hierarchically *segmenting* the given data into smaller retrievable data units, and the second, hierarchically *grouping* different units into larger yet meaningful categories. In this section, we focus on issues in segmenting multimedia news broadcast data into retrievable units that are directly related to what users perceive as meaningful. The basic units after segmentation can be indexed and browsed with efficient algorithms and tools. The levels of abstraction include commercials, news stories, news introductions, or news summaries of the day. A particular focus is development of a solution that effectively integrates the cues from video, audio, and text.

Much research has concentrated on segmenting video streams into "shots" using low level visual features [44][45]. Based on such segmentation, the retrievable units are low level structures such as clips of video represented by key frames. The insufficiency associated with such approaches is multifold. First, these low-level structures do not correspond in a direct and convenient way to the underlying semantic structure of the content, hence, making it difficult for users to browse. Second, due to the large amount of low-level structures generated from the segmentation, it provides little improvement in browsing efficiency compared with linear search. When users interacts with an information retrieval system, they expect the system to provide a condensed, concise characterization of the content available. The system should offer a mechanism for users to construct clear, unambiguous queries, and also return requested information that is long enough to be informative and as short as possible to avoid irrelevant information [46]. Thus, multimedia processing systems need to understand, to a certain extent, the content embedded in the multimedia data so that they can recover the semantic structure originally intended to be delivered. The conventional "scene cut" based systems can not meet such requirements although they are able to facilitate retrieval in a limited sense. Therefore,

recently, more attention has been directed towards automatically recovering semantically meaningful structures from multimedia data.

Integrated Semantic Segmentation Using Multimedia Cues

One recent direction in semantic-level segmentation focuses on detecting an isolated, predefined set of meaningful events. For example, Chang *et al.* [47] attempted to extract "touchdown" events from a televised football game video by combining visual and audio data. Another recent approach generates a partition over the continuous data stream into distinct events, so that each segment corresponds to a meaningful event such as a news story [46][48][49][50][51][52][53][54][55][56]. In other words, the effort aims to recover the overall semantic structure of the data that reflects the original intention of the information creator (such structure is lost when the data is being recorded on the linear media). Some work along this line has made use of the information from a single media (visual or text) [48][49][50][51][52][53]. With visual cues only, it is extremely difficult to recover the true semantic structure [51][52][53]. With text information only, due to the typical use of fixed processing windows, it has been shown that obtaining a precise semantics boundary is difficult and the accuracy of segmentation varies with the window size used [49][50]. Other more recent works integrated the cues from different media to achieve story segmentation in broadcast news [56][54][46]. By combining cues, the story boundaries can be identified more precisely.

Some existing approaches rely on fixed textual phrases or exploit the known format of closed captions. In the works of Merlino and Maybury [54][55][56], particular phrases (e.g., “still to come on the news...") are exploited as cues for story boundaries. In Hauptmann's work [46], particular structures of closed captions (markers for different events) are used to segment both commercials and stories. Other techniques aimed at developing an approach that can be used in more general settings, i.e., solutions that do not make assumptions about certain cue phrases, the availability of closed captions, or even the particular format of closed captions [57]. Acoustic, visual, and textual signals are utilized at different stages of processing to obtain story segmentation. Audio features are used in separating speech and commercials; anchorperson segments are detected based on speaker recognition techniques; story level segmentation is achieved using text analysis that determines how blocks of text should be merged to form news stories, individual story introductions, and an overall news summary of the day. Once different semantic units are extracted, they are aligned in time with the media data, so that proper representations of these events can be constructed across different media, in such a way that the audio-visual presentation can convey the semantics effectively.

In this section, we review an integrated solution for automated content structuring for broadcast news programs. We use this example to illustrate the development of tools for retrieving information from broadcast news programs in a semantically meaningful way at different levels of abstraction.

A typical national news program consists of news and commercials. News consists of several headline stories, each of which is usually introduced and summarized by the anchor prior to and following the detailed reports by correspondents, quotes, and interviews of news makers. Commercials are usually found between different news stories. With this observation, we try to recover this content hierarchy by utilizing cues from different media whenever it is appropriate. Figure 1 shows the hierarchy we intend to recover.

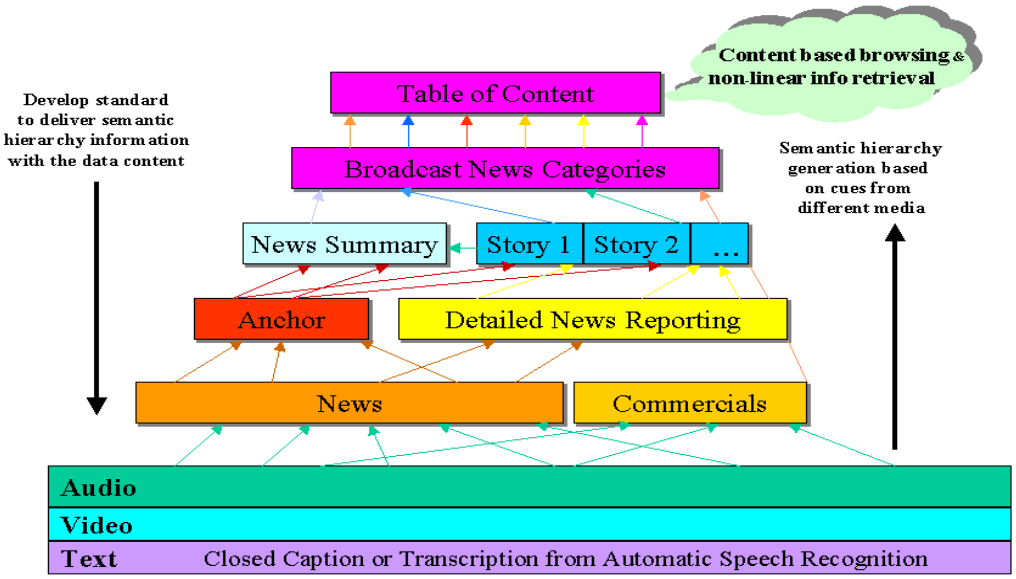

Figure 1. Content hierarchy of broadcast news programs.

In this hierarchy, the lowest level contains the continuous multimedia data stream (audio, video, text). At the next level, we separate news from commercials. The news is then segmented into the anchorperson's speech and the speech from others. The intention of this step is to use the recognized anchor's identity to hypothesize a set of story boundaries that consequently partition the continuous text into adjacent blocks of text. Higher levels of semantic units can then be extracted by grouping the text blocks into news stories and news introductions. In turn, each news story can consist of either the story by itself or augmented by the anchorperson's introduction. Detailed semantic structure at the story level is shown in Figure 2.

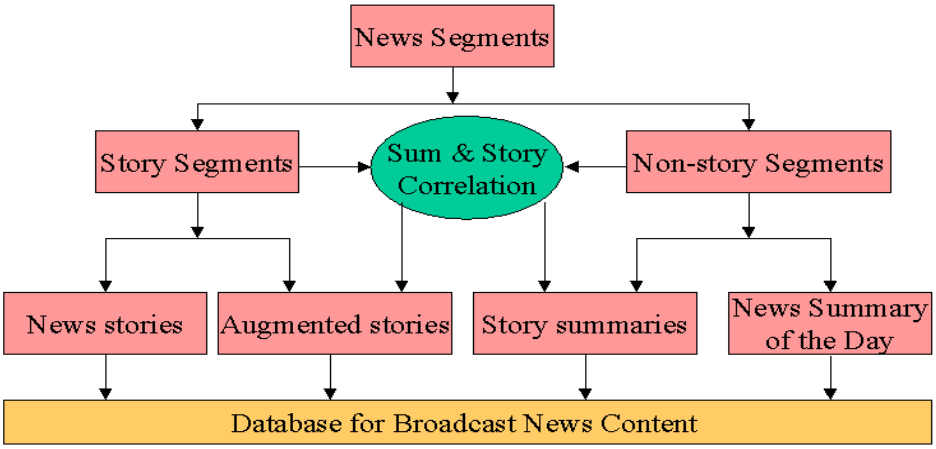

Figure 2. Relationship among the semantic structures at the story level.

In this Figure, input consists of news segments with boundaries determined by the location of anchorperson segments. Commercial segments are not included. Using duration information, each news segment is initially classified as either the story body (having longer duration) or news introduction or non-story segments (having shorter duration). Further text analysis verifies and refines the story boundaries, the introduction associated with each news story, and the news summary of the day.

The news data is segmented into multiple layers in a hierarchy to meet different needs. For instance, some users may want to retrieve a story directly; some others may want to listen to the news summary of the day in order to decide which story sounds interesting before making further choices; while yet others (e.g., a user employed in the advertising sector) may have a totally different need to monitor commercials from competitors in order to come up with a

competing commercial. The segmentation mechanism partitions the broadcast data in different ways so that direct indices to the events of different interests can be automatically established.

Representations and Browsing Tools

Once semantic structures are recovered and indexed, efficient tools are needed to present the extracted semantic units in a form that is compact, concise, easy to understand, and at the same time visually pleasing. Now, we discuss three aspects of this task. First, how to present the semantic structure to the users; second, how to represent the particular semantics based on the content of the news story; and third, how to form the representation for news summary of the day.

A commonly used presentation for semantic structure is in the form of a table of content. In addition, in order to give users a sense of time, we also use a streamline representation for the semantic structure. Figure 3 shows one presentation for the semantic structure of a news program. On the left side of the screen, different semantics are categorized in the form of a table of contents (commercials, news, and individual news stories, etc.). It is in a familiar hierarchical fashion which indexes directly into the time stamped media data. Each item listed is color coded by an icon or a button. To play back a particular item, a user simply clicks on the button of the desired item in this hierarchical table. On the right of this interface is the streamline representation where the time line runs from left to right and top to bottom. The time line has two layers of categorization. The first layer is event based (anchor's speech, others' speech, and commercials) and the second layer is semantic based (stories, news introduction, and news summary of the day). Each distinct section is marked by a different color and the overall color codes correspond to the color codes used in the table of content. Obviously, the content categorized in this representation is aligned with time.

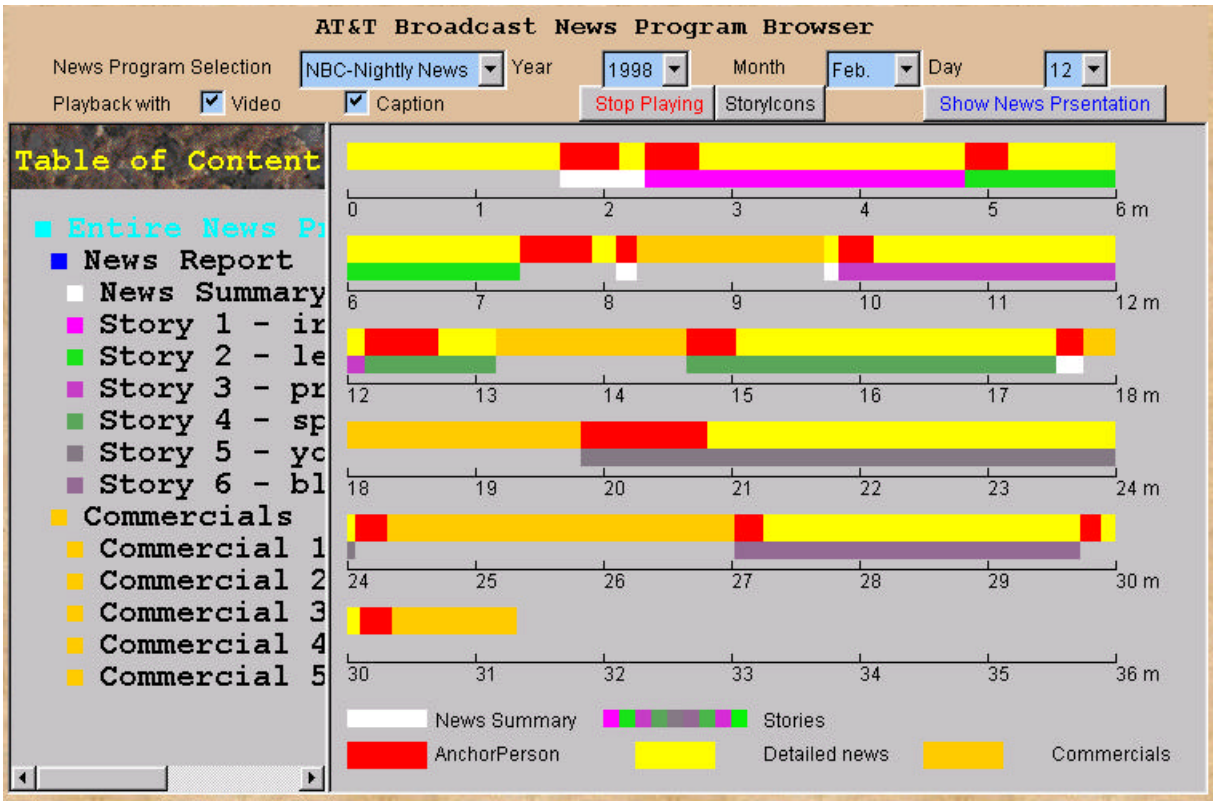

Figure 3. Representation for extracted semantic structures.

For each extracted news story, we developed two forms of representation. One is textual and another is a combination of text with visual. Our goal is to automatically construct the representation in a form that is most relevant to the content of the underlying story. For textual representation, keywords are chosen from the story according to their importance computed as weighted frequency. In the table of content shown in Figure 3, next to each story listed, a set of 10 keywords are given. The intention is that users will get a feeling about the content of the story. Another more detailed representation for a story is called ``story icon''. Figure 4 demonstrates this kind of story representation.

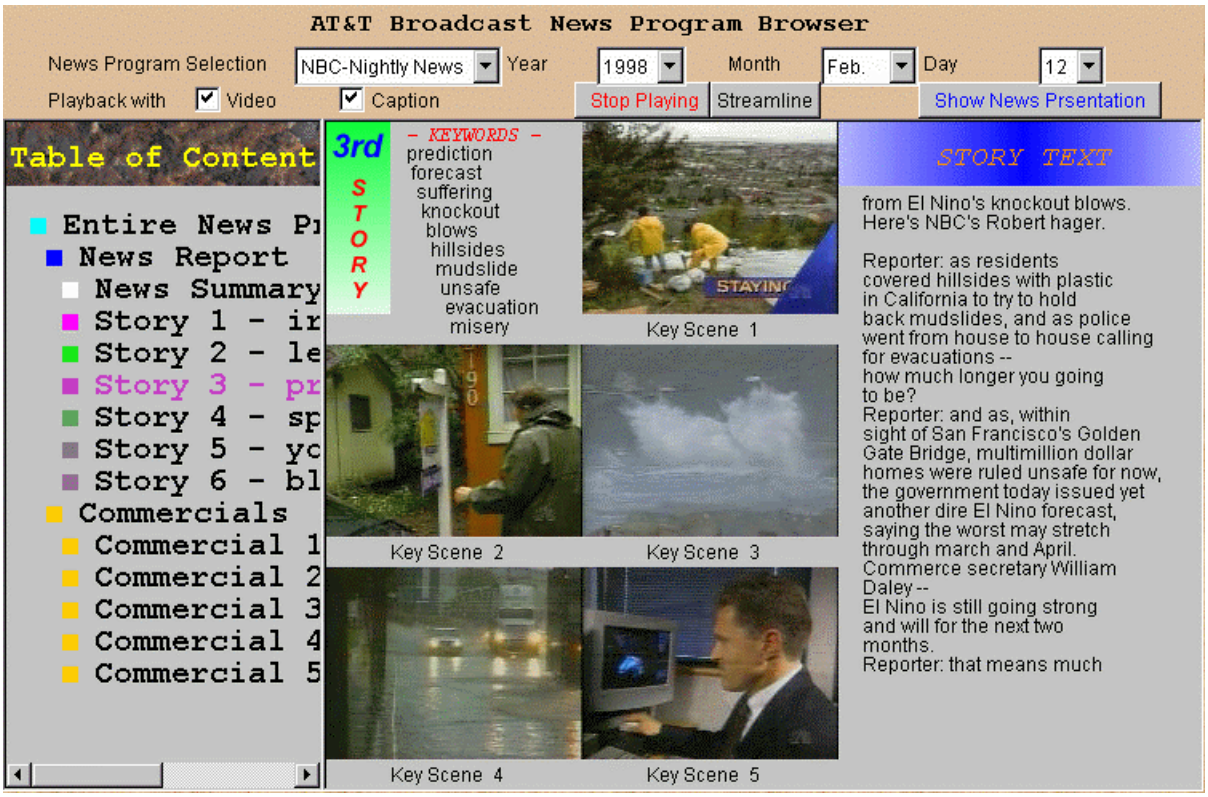

Figure 4. Visual representation for stories about El Nino.

The presentation for each story has three parts: the upper left corner is a set of 10 keywords automatically chosen from the segmented story based on the relative importance of the words; the right part displays the text of the story; the rest is the visual presentation of the story consisting of five images chosen from video in the content based manner described above. Figure 9a is the visual representation of the story about El Nino. It is compact, semantically revealing, and visually informative with respect to the content of the story. A user can choose either to scroll the text on the right to read the story or to click on the button of that story to playback synchronized audio, video, and text, all starting from where the story begins. Compared with linear browsing or low level scene cut browsing, our system allows for more effective content based non-linear information retrieval.

Finally, we construct the representation for the news summary of the day. It is composed of *K* images, where *K* is the number of headline stories on a particular day. The *K* images are chosen so that they are the most important in each story, measured by the covered area size in the keyword histogram. Figure 5 gives the visual presentation for the news summary of the day for the NBC Nightly News on the 12th of February, 1998. From this presentation, a user can see immediately that there are a total of six headline stories on that particular day. Below the representative image for each story, the list of its keywords is displayed as a dynamic right-to-left flow so that users can get a sense of the story from the keywords (The movement of keywords is not apparent in Fig. 5 because we can not show a dynamic video sequence). From these examples, the effectiveness of this story-telling visual representation for the news summary is evident.

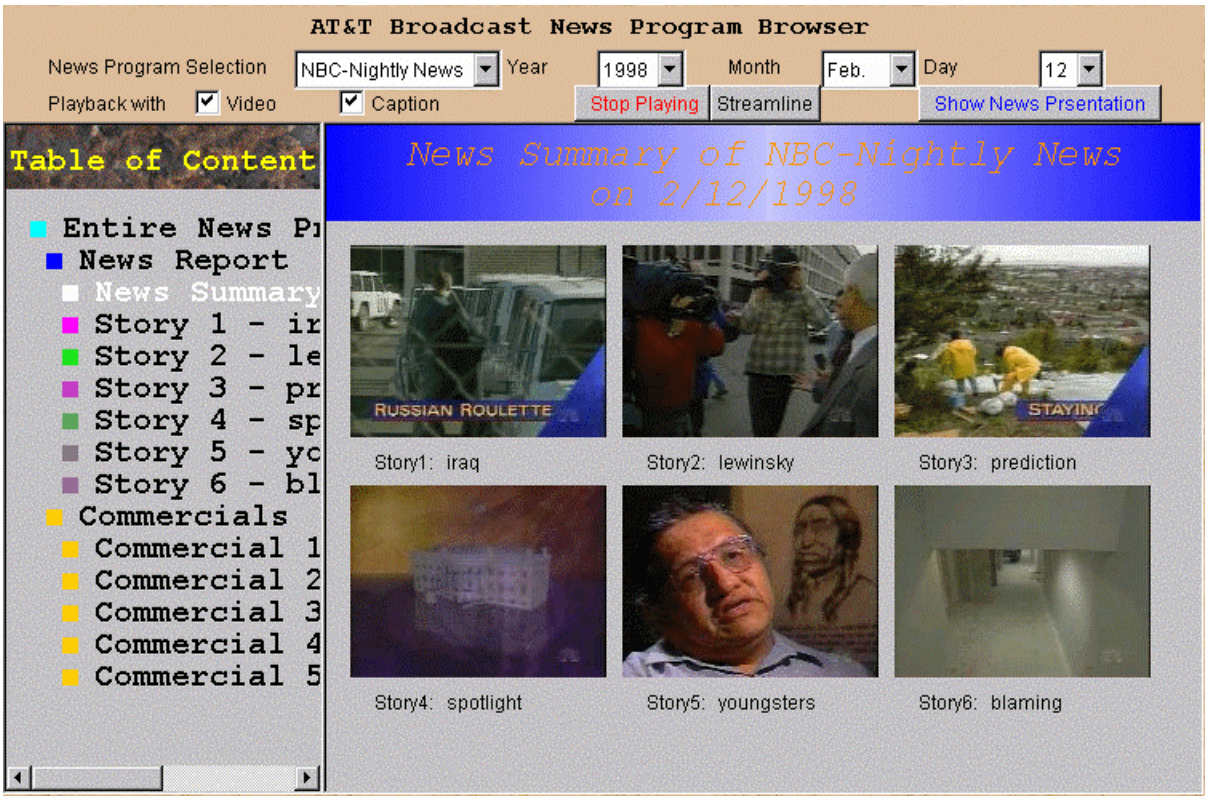

Figure 5. Representation for news summary of the day.

3. Object-based Spatio-Temporal Visual Search and Filtering

An active research direction complementary with the above one using semantic-level structuring is the one that directly exploits low-level objects and their associated features in images or videos. An intuitive and popular approach is to segment and provide efficient indexes to salient objects in the images. Such segmentation processes can be implemented using automatic or semi-automatic tools [6][8]. Examples of salient objects may correspond to meaningful real-world objects such as houses, cars, and people, or low-level image regions with uniform features, such as color, texture, or shape.

Several notable image/video search engines have been developed using this approach, namely searching images/videos by example, by features, or by sketches. Searching for images by examples or templates is probably the most classical method of image search, especially in the domains of remote sensing and manufacturing. From an interactive graphic interface, users select an image of interest, highlight image regions, and specify the criteria needed to match the selected template. The matching criteria may be based on intensity correlation or feature similarity between the template image and the target images.

In feature-based visual query, users may ask the computer to find similar images according to specified features such as color, texture, shape, motion, and spatio-temporal structures of image regions [1][3][4][5]. Some systems also provide advanced graphic tools for users to directly draw visual sketches to describe the images or videos they have in mind [5][6][7]. Users are also allowed to specify different weightings for different features. Figure 6 shows

query examples using the object color and the motion trail to find a video clip of a downhill skier, and using color and motions of two objects to find football players.

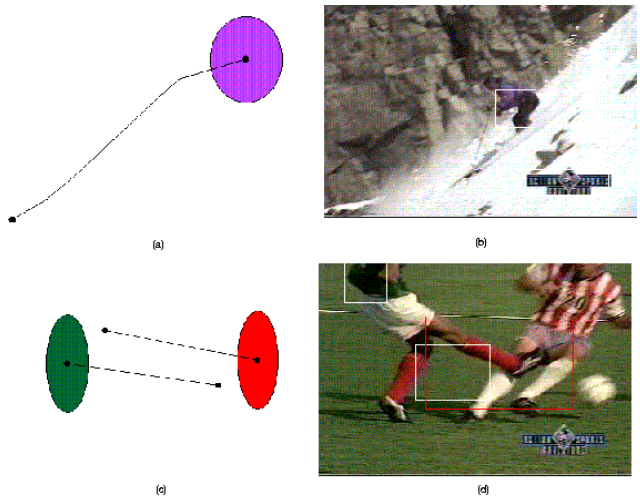

Figure 6. Object-oriented search using example image objects, features, and their relationships. By matching the object features and their spatio-temporal relationships (e.g., query sketches shown in (a) and (c)), video clips containing most similar objects and scenes (e.g., (b) downhill skiing and (d) soccer players) are returned. (Video courtesy of Actions, Sports, and Adventure, Inc.)

Object-Based Video Segmentation and Feature Extraction

To demonstrate the processes involved in developing the above object-based video search tools, we review the process of developing the VideoQ search system [6] in this subsection.

Video sequences are first decomposed into separate shots. A video shot has a consistent background scene, although the foreground objects may change dynamically (they may even occlude each other, disappear, or re-appear). Video shot separation is achieved by scene change detection. Scene change may include abrupt scene change, transitional changes (e.g., dissolve, fade in/out, and wipe). Once the video is separated into basic segments (i.e., video shots), salient video regions and video objects are extracted. Video object is the fundamental level of indexing in VideoQ. Primitive regions are segmented according to color, texture, edge, or motion measures. As these regions are tracked over time, temporal attributes such as trajectory, motion pattern, and life span are indexed. These low-level regions may also be used to develop a higher level of indexing that includes links to conceptual abstractions of video objects.

Based on the object-based representation model, a visual feature library (VFL) has also been developed. The VFL includes a rich set of visual features automatically extracted from the video objects. For example, color may include single color, average color, representative color, color histogram, and color pairs [10][11][12]. Texture may include wavelet-domain textures, texture histogram, Tamura texture, and Laws Filter-based texture [13][14][39][40]. Shape may include geometric invariants, moments of different orders, polynomial approximation, spline approximation, algebraic invariants *etc.* Motion may include trajectories of the centroid of each object and the affine models of each object over time (with camera

motion compensation). The concept of VFL is to capture distinctive features that can be used to distinguish salient video objects efficiently and effectively. The final selection of features and their specific representations should depend on the resource (computing and storage) available, and the application requirements (e.g., geometric invariance in object matching).

In measuring the similarity between video objects, various distance functions for measuring the feature similarity could be used, such as the Euclidean distance, the Mahalalobis distance, the quadratic distance with cross-element correlation, and L1 distance. Usually it's difficult to find the optimal distance function. Instead, it's more beneficial to develop a flexible framework that can support various search methods, feature models, and matching metrics.

The above visual paradigm using the object-based representation and the VFL opens up a great opportunity for image/video search. In VisualSEEk [5], we integrated image search based on visual features and spatial constraints. It used a generalized technique of 2D-strings to provide a framework for searching for and comparing images by the spatial arrangement of automatically segmented feature regions. Multiple regions can be queried either by their absolute or relative locations. As shown in Figure 7, the overall query strategy consists of "joining" the queries based on the individual regions in the query image. Each region in the query image is used to query the entire database of regions. The resulting list of matched regions are then combined by the join operation, which consists of intersecting the results of region matches. The join operation identifies the candidate images which contain matches to all the query regions, and then evaluates the relative spatial locations to determine the best matched image that satisfies the constraints of relative region placement. For video, more sophisticated spatio-temporal relationships can be verified at this stage to find video clips including rich spatio-temporal activities.

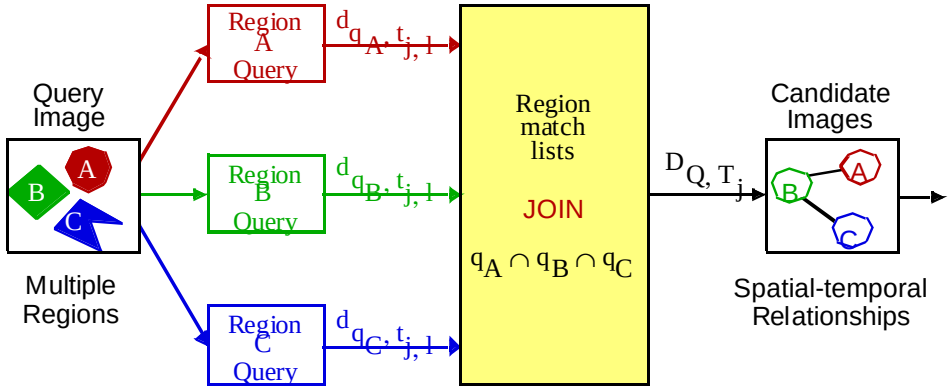

Figure 7. Query processing architecture supporting multiple regions and verification of spatio-temporal relationships among regions

The above video object search system achieves fully automatic object segmentation and feature extraction at a low-level. However, the results of object segmentation remain at a low level (e.g., image regions with uniform features). These usually do not correspond well to the real world physical objects. One way to solve this problem is to use some input from users or content providers. For example, in the new MPEG-4 standard [25], video scenes consist of separate video objects that may be created, compressed, and transmitted separately. The video objects in MPEG-4 usually correspond to the semantic entities (e.g., people, car, and house). Region segmentation and feature extraction mentioned above can be applied to analyze these

objects and obtain efficient indexes for search and retrieval. Some systems combine some initial user interaction and automatic image region tracking to extract semantic video objects in a semi-automatic process [42]. Figure 8 shows the interface of a search engine for MPEG-4 semantic video objects [43]. Note that in addition to the single-level spatio-temporal search functions available in VideoQ, the semantic object search engine indexes spatio-temporal relationships at multiple levels (i.e., regions and objects) and thus allows for more flexible search.

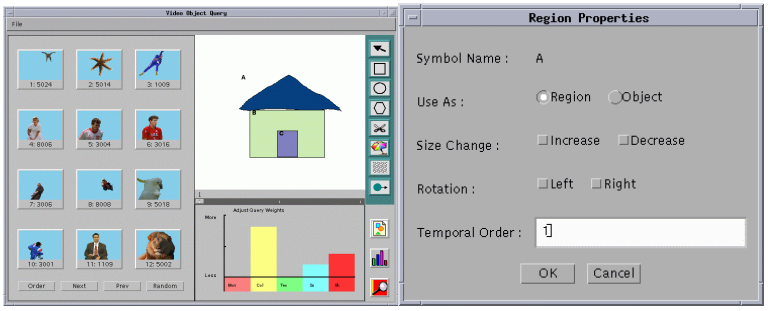

Figure 8 Search interface for the AMOS semantic object search engine. Objects are at the semantic level (e.g., people, animals, cars) rather than the region-level objects used in VideoQ.

4. Semantic-Level Content Classification and Filtering

The spatio-temporal search tools described above provide powerful capabilities for searching videos or images at the low level. In many situations, users prefer to use simple and direct methods. For example, users may just want to browse through the content categories in which each image or video is classified into one or several meaningful classes. In other cases, we have found that the combination of feature-based similarity search tools and the subject navigation utilities achieves the most successful search results.

Recently, more research interest has emerged in developing automatic classification or filtering algorithms for mapping images or videos to meaningful classes, e.g., indoor, outdoor, or people [15][16][17][18]. In this section, we discuss two complementary approaches for identifying semantic concepts in multimedia data.

Content Modeling Using Probabilistic Graphic Models

The first approach uses probabilistic graphic models to identify events, objects, and sites in multimedia streams by computing probabilities like *P*(underwater AND shark | segment of multimedia data). The basic idea is to estimate the parameters and structure of a model from a set of labeled multimedia training data.

A *multiject* (multimedia object) has a semantic label and summarizes the time sequences of low-level features of multiple modalities in the form of a probability, *P*(semantic label | multimedia sequence). Most multijects fall into one of the three categories: *sites*, *objects*, and *events*. While some multijects are supported mainly by video (*e.g.*, shark), others are supported mainly by audio (*e.g.*, interior of traveling train), still others are strongly supported by both audio and video (e.g., explosion).

The *lifetime* of a multiject is the duration of multimedia input that is used to determine its probability. In general, some multijects (*e.g.*, family quarrel) will live longer than others (*e.g.*, gun shot) and the multiject lives can overlap. For simplicity, we could break the multimedia into shots and within each shot fix the lifetimes of all multijects to the shot duration. Given the multiject probabilities, this leads to a simpler static inference problem within each shot. Although this approximation ignores event sequences within a shot (e.g., gun shot followed by family quarrel), it leaves plenty of room for useful inferences, since directors often use different shots to highlight changes in action and plot.

We model each modality in a multiject with a Hidden Markov Model (HMM) [18]. We investigate what combinations of input features and HMM structure give sufficiently accurate models. For example, we found that in the case of the explosion multiject, a color histogram of the input frames and a 3-state HMM give a reasonably accurate video model [20]. In contrast, a video model for a bird may require object detection and tracking.

Each HMM in a multiject summarizes the time sequence for its corresponding modality. In order to summarize modalities, it is necessary to identify likely correspondences between events in the different modalities. For example, one of our experiments calculated the posterior probabilities that an explosion occurred in a movie clip at or before time *t* under an audio and a video HMM that were each trained on examples of explosions. In one case, the sound of the explosion begins roughly 0.15 seconds (8 audio frames at 50 Hz or 5 video frames at 30 Hz) later than the video of the explosion. In other cases, we found that the audio and video events were more synchronized. To accurately detect explosions, the explosion multiject should be invariant to small time differences between the audio and video events. However, if the multiject is overly tolerant of the time difference, it may falsely detect non-explosions. For example, a shot consisting of a pan from a sunrise to a waterfall will have a flash of bright red followed, after a large delay, by a thundering sound with plenty of white noise. Although the audio and video features separately match an explosion, the large time delay indicates it is not an explosion.

We have used two methods [20][21] for summarizing the modalities in a multiject. In the first method, each HMM models an event in a different modality and the times at which the event occurs in the different modalities are loosely tied together by a kernel (*e.g.*, Gaussian). An example of a pair of such event-coupled HMMs is shown in Figure 9, where t^A and t^V are the times at which the explosion event begins in the audio and video. In [20], we discussed an efficient algorithm for computing the probability of the multiject, *e.g.*, *P*(explosion | video sequence, audio sequence). In the second method [21], a high-level HMM is trained on the states of the HMMs for the different modalities. Using a fast, greedy, bottom-up algorithm, the probability of the multiject can be approximated quite accurately. Note that the above model and structure can be greatly expanded to determine a multiject representation that is simple enough for learning and inference while being powerful enough to answer useful queries.

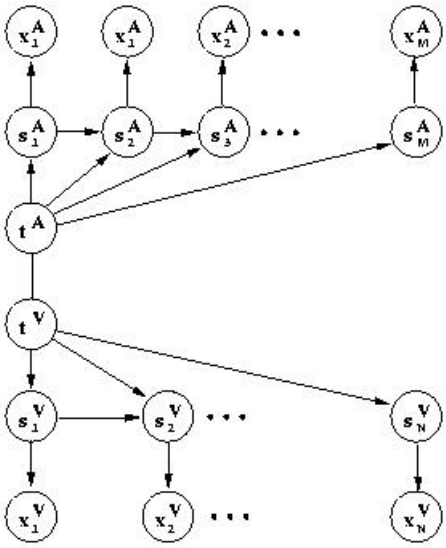

Figure 9. A graphic model for a *multiject* (multimedia objects) that couples together the two HMMs.

Suppose we are interested in automatically identifying movie clips of exotic birds from a movie library. We could compute the probability of the bird multiject for each multimedia shot in the library and then rank the shots according to these probabilities. However, by computing the probabilities of other related multijects, we may be able to derive additional support for the bird multiject. For example, an active waterfall multiject provides evidence of an exotic location where we are more likely to find an exotic bird. As a contrasting example, an active underwater multiject decreases the support for a bird being present in the shot.

We use the *multinet* (multiject network) as a way to represent higher-level probabilistic dependencies between multijects. Figure 10 shows an example of a multinet. In general, all multijects derive probabilistic support from the observed multimedia data (directed edges), as described in the previous section. The multijects are further interconnected to form a graphical probability model. Initially, we will investigate Boltzmann machine models [22] that associate a real-valued weight with each undirected edge in the graph. The weight indicates to what degree two multijects are correlated *a priori* (before the data is observed). In Figure 10, plus signs indicate the multijects are correlated *a priori* whereas minus signs indicate the multijects are anti-correlated.

Returning to the bird example, a plus sign on the connection between the bird multiject and the waterfall multiject indicates that the two multijects are somewhat likely to be present simultaneously. The minus sign on the connection between the bird multiject and the underwater multiject indicates that the two multijects are unlikely to be present simultaneously. The graphical formulation highlights interesting second-order effects. For example, an active waterfall multiject supports the underwater multiject, but these two multijects have opposite effects on the bird multiject.

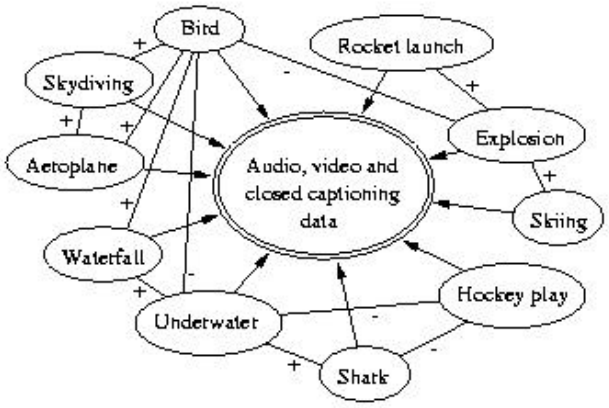

Figure 10. A multinet (multimedia network) probabilistically links multijects (multimedia objects) to the data and also describes the probabilistic relationships between the multijects. Plus signs indicate the multijects are correlated a priori (before the data is observed); minus signs indicate the multijects are anticorrelated.

In general, exact inference such as computing *P*(bird | multimedia data) in a richly connected graphical model is intractable. The second-order effects described above imply that many different combinations of multijects need to be considered to find likely combinations. However, there has been promising work in applying approximate inference techniques to such intractable networks in areas including pattern classification, unsupervised learning, data compression and digital communication [23]. In addition, approaches using Markov chain Monte Carlo methods [22] and variational techniques [24] are promising.

Indexing Multimedia with Semantic Templates

In this subsection, we discuss a different semantic-level video indexing technique, Semantic Templates (STs) [26]. Semantic templates associate a set of exemplar queries with each semantic. Each query in the template has been chosen since it has been successful at retrieving the concept. The idea is that since a single successful query rarely completely represents the information that the user seeks, it is better to *cover* the concept using a set of successful queries. Semantic Templates can be defined over any type of indexable media. Here we focus on video, semantic visual templates (SVTs). Figure 11 shows example SVTs for the "high jumper" concept and the "sunsets" concept.

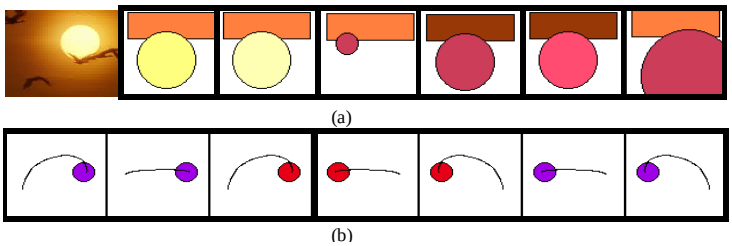

Figure 11. A semantic-level search paradigm using semantic visual templates. Image icons shown are subsets of optimal templates for each concept ((a) “sunset” (b) “high jumper”) generated through the two-way interactive system.

The goal of the ST is similar to that of multijects mentioned above to detect and recognize video objects, sites, or events at the semantic level. However, it is based on different but synergistic principles. The generation of a semantic template involves no labeled ground truth data. What we do require is that some positive examples of the concept be present in the database so that they can be used in the interactive process when users interact with the system to generate the optimal set of STs. Development of STs utilizes the following unique principles.

Two-way Learning: The template generation system emphasizes the two-way learning between the human and the machine. Since the human being is the final arbiter of the “correctness" of the concept, it is essential to keep the user in the template generation loop. The user defines the video templates for a specific concept with the concept in mind. Using the returned results and relevance feedback the user and the system converge on a small set of queries that *best* match (*i.e.*, provide maximal recall) the user's concept.

Intuitive Models: Semantic templates are intuitive, understandable models for semantic concepts in the videos. The final sets of SVT's can be easily viewed by the user. Users can have direct access and make manipulation to any template in the library.

Synthesizing new Concepts: Different ST's can be graphically combined to synthesize more complex templates. For example, templates for *high-jumpers* and *crowds* can be combined to form a new template for “high-jumpers in front of a crowd." The audio templates for “crowd" and “ocean sounds" and the visual template for “beach" can be combined to form the “crowds at a beach" template.

The template framework is an extended model of the object-oriented video search engine that has been described in Section 3. The video object database consists of video objects and their features extracted in the object segmentation process. A visual template may consist of two types of concept definitions: *object icons* and *example scenes/objects*. The object icons are animated sketches like the ones used in VideoQ. In VideoQ, the features associated with each object and their spatial and temporal relationships are important. The example scenes or objects are represented by the feature vectors extracted from these scenes/objects. Typical examples of feature vectors that could be part of a template are histograms, texture information and structural information (i.e., more or less the global characteristics of the example scenes). The choice between an icon-based realization or an example based realization depends upon the semantic that we wish to represent. For example, a “sunset" can be very well represented by

using a couple of objects, while a waterfall or a crowd is better represented using example scenes characterized by a global feature set. Hence, each template contains multiple icons, and example scenes/objects to represent the idea. The elements of the set can overlap in their coverage. The goal is to come up with a minimal template set with maximal coverage.

Each icon for the concept comprises multiple objects which are associated with a set of visual attributes. The relevance of each attribute and each object to the concept is also specified using a context specification questionnaire. For example, for the concept “sunsets", color and spatial structures of the objects (sun and sky) are more relevant. The object “sun" may be non-mandatory since some sunset videos may not have the sun visible. For the concept “high jumper", the motion attribute of the foreground object (mandatory) and the texture attribute of the background object (non-mandatory) are more relevant than other attributes.

Development and application of semantic templates requires the following components:

> *Generation*: This is used to generate ST's for each semantic concept. We will describe an interactive learning system in which users can interactively define their customized ST's for a specific concept.
>
> *Metric*: This is used to measure the “fitness" of each ST in modeling the concept associated with the video shot or the video object. The fitness measure can be modeled by the spatio-temporal similarity between the ST and the video.
>
> *Applications*: An important challenge in applications is to develop a library of semantic concepts that can be used to facilitate video query at the semantic level. We will describe the proposed approaches to achieving such systems later.

Automatic generation of the ST's is a hard problem. Hence we use a two-way interaction between the user and system in order to generate the templates (shown in Figure 12). In our method, given the initial query scenario and using relevance feedback, the system converges on a small set of icons (exemplar queries for both audio and video) that gives us maximum recall. We now explain the mechanisms for generation of semantic visual templates.

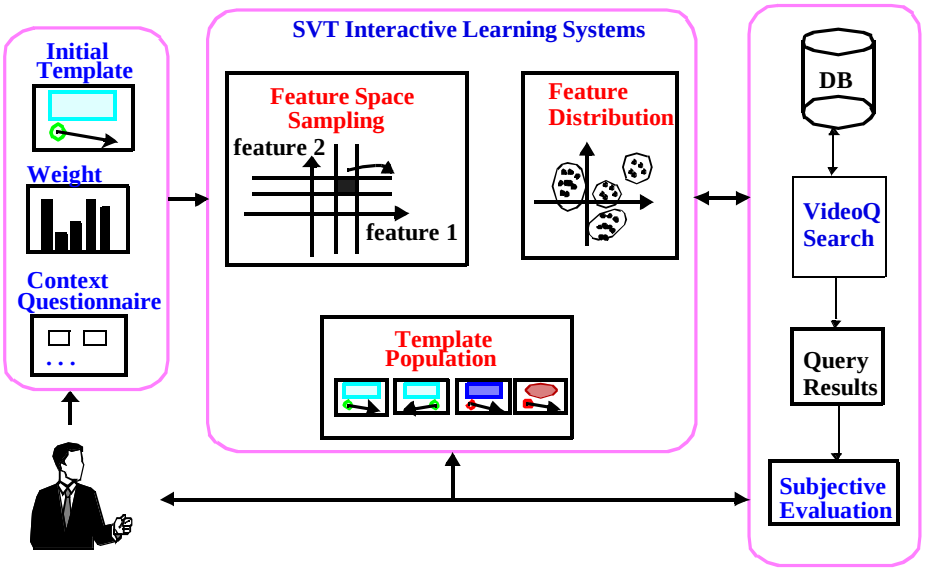

Figure 12. System architecture for Semantic Visual Template.

The user comes to the system and sketches out the concept that for which he wishes to generate a template. The sketch consists of several objects with spatial and temporal constraints. The user can also specify whether the object is mandatory or not. Each object is comprised of several features. The user also assigns relevance weights to each feature of each object. This is the initial query scenario that the user provides to the system.

The initial query can also be viewed as a point in a high dimensional feature space. Clearly, we can also map all videos in the database in this feature space. Now, in order to generate the possible icon set automatically, we need to make jumps in each of the features for each object. Before we do so, we must first determine the jump step size i.e. quantize the space. This we do with the help of the weight that the user has input along with the initial query. This weight can be thought of as the user's belief in the relevance of the feature with respect to the object to which it is attached. Hence, a low weight gives rise to coarse quantization of the feature and vice versa.

Since the total number of icons possible using this technique increases very rapidly, we don't allow for joint variation of the features. For each feature in each object, the user picks a plausible set for that feature. The system then performs a join operation on the set of features associated with the object. The user then picks the joins that are most likely to represent variations of the object. This results in a candidate icon list.

In a multiple object case, we do an additional join with respect to the candidate lists for each object. Now, as before, the user picks the plausible scenarios. After we have generated a list of plausible scenarios we then query the system using the icons the user has picked. Using relevance feedback on the returned results (the user labels the returned results as positive or negative), we then determine the icons that provide us with maximum recall.

We now discuss a detailed example showing the generation mechanism for creating the semantic visual template for slalom skiers.

- We begin the procedure by answering the context questionnaire shown in Figure 13(a). We label the semantic visual template “slalom". We specify that the query is object-based and will be composed of two objects. Then (Figure 13(b)), we sketch the query. The large, white background object is the ski slope and the smaller foreground object is the skier with its characteristic zigzag motion trail.

- We assign maximum relevance weights to all the features associated with the background and skier. We also specify that the features belonging to the background will remain static, while those of the skier can vary during template generation. Then, the system automatically generates a set of test icons, and we select plausible feature variations in the skier's color and motion trajectory.

- A set of potential icons including both the background and the foreground skier are shown in Figure 13(c). The user then chooses a candidate set to query the system. The 20 closest video shots are retrieved for each query. The user provides relevance feedback, which guides the system to a small set of exemplar icons associated with slalom skiers.

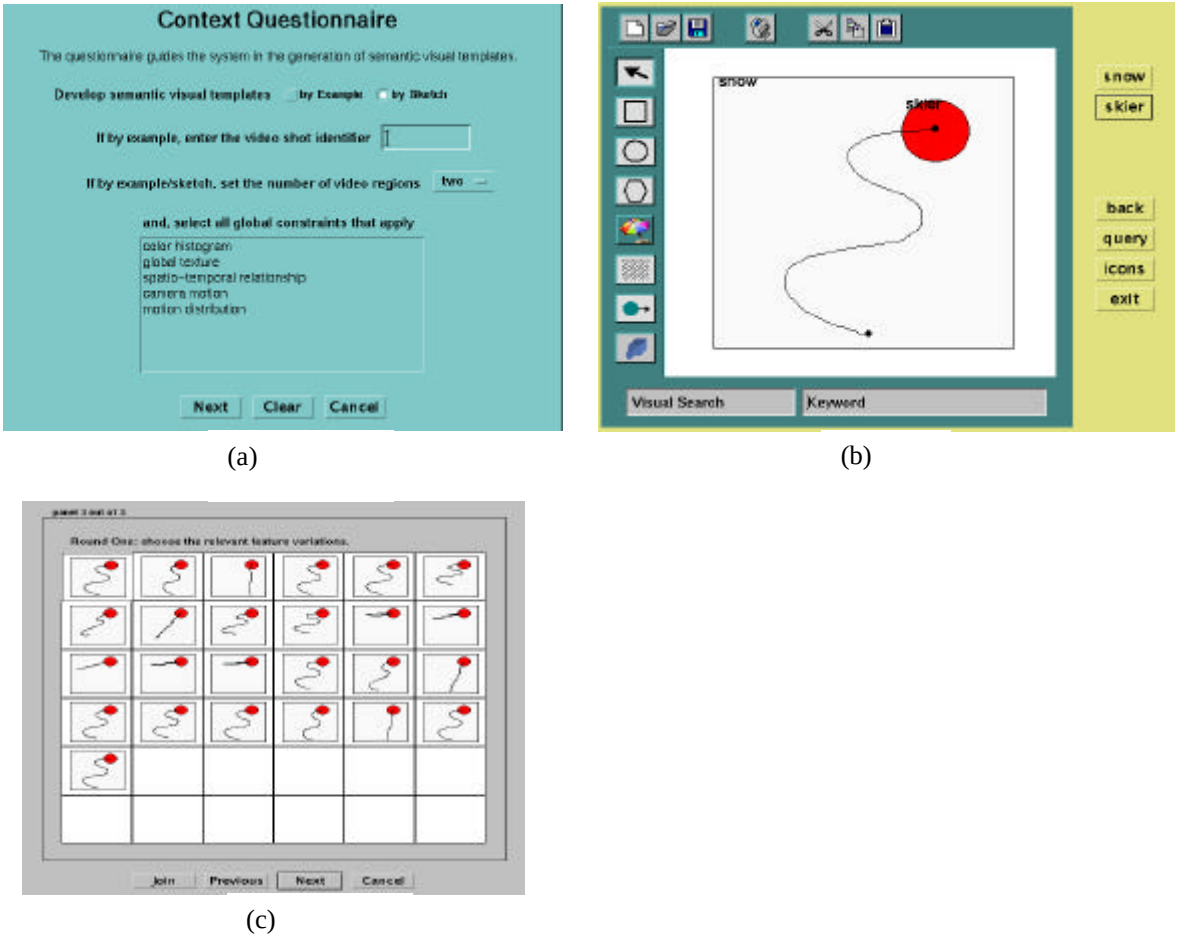

Figure 13. Example of semantic template development – Slalom. (a) questionnaire for users to define the objects and features in a template; (b) initial graphic definition of the template; and (c) candidate icons generated by the system.

As mentioned earlier, the framework of the semantic templates can be applied to multiple medium modalities including audio and video. Once we have the audio and the visual templates for different concepts such as “skiing", “sunsets" *etc*. the user can interact with the system at the concept level. The user can compose a new multimedia concept that is built using these templates. For example if the user wants to retrieve a group of people playing beach volleyball, he would use the visual templates of beach volleyball and beach sounds to generate a new query: {{Video: Beach volleyball, Beach}, {Audio: Beach Sounds}}. Then, the system would search for each template and return a result based on the users search criterion. For example he may indicate that he needs only some of the templates to be matched or that he needs all templates to be matched. Also, once we have a collection of audio and video templates for a list of semantics, we then use these templates to match with the new videos and thereby generating a list of potential audio and video semantics (thereby generating a semantic index) that are associated with the video clip. Early results on querying using SVT's indicate that the concept works well in practice. For example, in the case of sunsets, the original query over a large heterogeneous database yields only 10% recall. Using eight icons for the template, we boost this result to 50%.

5. Interoperable Content Description Schemes and Meta Search Engines

Techniques discussed above contribute to the state of the art in multimedia search and retrieval. Efficient tools and systems have been developed at different levels, including the physical level (*e.g.,* image object matching) and the semantic level (*e.g.,* news video content structure and semantic labeling). These tools can be optimized for the maximum power in specialized application domains. However, in many cases customized techniques may be used by different service providers for specialized content collection. Interoperability among the content indexes and the search functions becomes a critical issue. How do we develop a transparent search and retrieval gateway to hide the proprietary indexes and search methods and for users to access content in heterogeneous content sources?

Content-describing Schemes and MPEG-7

To describe various types of multimedia information, the emerging MPEG-7 standard [27] has the objective of specifying a standard set of descriptors as well as Description Schemes (DSs) for the structure of descriptors and their relationships. This description (*i.e.,* the combination of descriptors and description schemes) will be associated with the content itself to allow fast and efficient searching for material of a user's interest. MPEG-7 will also standardize a language to specify description schemes, (*i.e.,* a Description Definition Language (DDL)), and the schemes for encoding the descriptions of multimedia content.

In this section, we briefly describe a candidate interoperable content description scheme [30] and some related research on image meta search engines [28]. The motives of using these content description schemes for multimedia can be explained with the following scenarios.

- *Distributed processing*: The self-describing schemes will provide the ability to interchange descriptions of audio-visual material independently of any platform, any vendor, and any application. The self-describing schemes will enable the distributed processing of multimedia content. This standard for interoperable content descriptions will mean that data from a variety of sources can be easily plugged into a variety of distributed applications such as multimedia processors, editors, retrieval systems, filtering agents, *etc*.

- *Content exchange*: A second scenario that will greatly benefit from an interoperable content description is the exchange of multimedia content among heterogeneous audio-visual databases. The content descriptions will provide the means to express, exchange, translate, and reuse existing descriptions of audio-visual material.

- *Customized views*: Finally, multimedia players and viewers compliant with the multimedia description standard will provide the users with innovative capabilities such as multiple views of the data configured by the user. The user could change the display's configuration without requiring the data to be downloaded again in a different format from the content broadcaster.

To ensure maximum interoperability and flexibility, our description schemes use the eXtensible Markup Language (XML), developed by the World Wide Web Consortium (W3C) [29]. We briefly discuss the benefits of using XM and its relationship with other languages such as SGML here. SGML (Standard Generalized Markup Language, ISO 8879) is a standard language for defining and using document formats. SGML allows documents to be self-describing, i.e. they describe their own grammar by specifying the tag set used in the document and the structural relationships that those tags represent. However, full SGML contains many optional features that are not needed for Web applications and has proven to be too complex to current vendors of Web browsers.

The W3C has created an SGML Working Group to build a set of specifications to make it easy and straightforward to use the beneficial features of SGML on the Web [58]. This subset, called XML (eXtensible Markup Language), retains the key SGML advantages in a language that is designed to be vastly easier to learn, use, and implement than full SGML. A major advantage of using XML is that it allows the descriptions to be self-describing, in the sense that they combine the description and the structure of the description in the same format and document. XML also provides the capability to import external document type definitions (DTDs) into the image description scheme DTD in a highly modular and extensible way.

The candidate description scheme consists of several basic components*: object, object hierarchy, entity relation graph,* and *feature structure*. Each description includes a set of objects. The objects can be organized in one or more object hierarchies. The relationships between objects can be expressed in one or more entity relation graphs.

An object element represents a region of the image for which some features are available. It can also represent a moving image region in a video sequence. There are two different types of objects: *physical* and *logical* objects. Physical objects usually correspond to continuous regions of the image with some descriptors in common (semantics, features, etc.). Logical objects are groupings of objects (which may not occupy a continuous area in the image) based on some high-level semantic relationships (e.g. all faces in the image). The object element comprises the concepts of group of objects, objects, and regions in the visual literature. The set of all objects identified in an image is included within the object set element.

The description scheme also includes one or more object hierarchies to organize the object elements in the object set element. Each object hierarchy consists of a tree of object node elements. Each object node points to an object. The objects in an image can be organized by their location in the image or by their semantic relationships. These two ways to group objects generate two types of hierarchies: physical and logical hierarchies. A physical hierarchy

describes the physical location of the objects in the image. On the other hand, a logical hierarchy organizes the objects based on a higher level semantics, such as information in categories of *who, what object, what action, where, when, why*.

In addition to the object hierarchy, an *entity-relationship* model is also used to describe general relationships among object elements. Examples include spatial relationships, temporal relationships, and semantic-level relationships (e.g., A is shaking hands with B).

Each object includes one or more associated features. Each object can accommodate any number of features in a modular and extensible way. The features of an object are grouped together according to the following categories: *visual, semantic,* and *media*. Multiple abstraction levels of features can be defined. Each object may be associated with objects in other modalities through modality transcoding.

Examples of visual features include color, texture, shape, location, motion *etc*. These features can be extracted or assigned automatically or manually. Semantic features include annotations and semantic-level description in different categories (people, location, action, event, time, *etc*). Media features describe information such as compression format, bit rate, file location, *etc*.

Each feature of an object has one or more associated descriptors. Each feature can accommodate any number of descriptors in a modular and extensible way. External descriptors may also be included through linking to external DTD. For a given descriptor, the description scheme also provides a link to external extraction code and similarity matching code.

The unified description scheme may be applied to image, video, and combinations of multimedia streams in a coherent way. In the case of multimedia, a multimedia stream is represented as a set of multimedia objects that include objects from the composing media streams or other multimedia objects. Multimedia objects are organized in object hierarchies. Relationships among two or more multimedia objects that can not be expressed in a tree structure can be described using multimedia entity relation graphs. The tree structures can be efficiently indexed and traversed, while the entity relation graphs can model more general relationships.

Details of the description schemes mentioned above can be found in [59][60][61].

Multimedia Meta Search Engines

The above self-describing schemes are intuitive, flexible, and efficient. We have started to develop an MPEG-7 testbed to demonstrate the feasibility of our self-describing schemes. In our testbed, we are using the self-describing schemes for descriptions of images and videos that are generated by a wide variety of image/video indexing systems. In this section, we will discuss the impact of the MPEG-7 standard on a very interesting research topic, image metasearch engines.

Metasearch engines act as gateways linking users automatically and transparently to multiple search engines. Most of the current metasearch engines work with text. Our earlier work on metasearch engine, MetaSEEk [28], explores the issues involved in querying large, distributed, on-line visual information systems. MetaSEEk is designed to intelligently select and interface

with multiple on-line image search engines by ranking their performance for different classes of user queries. The overall architecture of MetaSEEk is shown in Figure 14. The three main components of the system are standard for metasearch engines; they include the *query dispatcher*, the *query translator*, and the *display interface*. The procedure for each search is as follows:

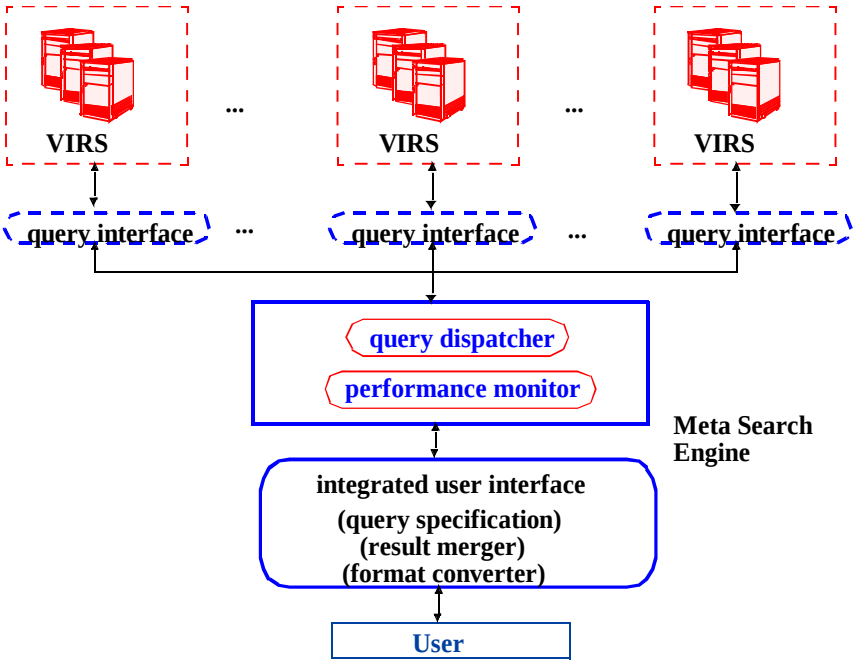

Figure 14. System architecture of metasearch engines such as MetaSEEk

- Upon receiving a query, the dispatcher selects the target search engines to be queried by consulting the performance database at the MetaSEEk site. This database contains performance scores of past query successes and failures for each supported search engine. The query dispatcher only selects search engines that provide compatible capabilities with the user's query (e.g. visual features and/or keywords).

- The query translators, then, translate the user query to suitable scripts conforming to the interfaces of the selected search engines.

- Finally, the display component uses the performance scores to merge the results from each search engine, and displays them to the user.

MetaSEEk evaluates the quality of the results returned by each search engine based on the user's feedback. This information is used to update the performance database. The operation of MetaSEEk is very restricted by the interface limitations of current search engines. For example, most existing systems can only support query by example, query by sketch, keyword search. Results usually are just a flat list of images (with similarity scores, in some cases).

As discussed in the previous section, we envision a major transformation in multimedia search engines thanks to the development of the MPEG-7 standard. Future systems will accept not only queries by example and by sketch, but also queries by MPEG-7 multimedia descriptions. Users will be able to submit desirable multimedia content (specified by MPEG-7 descriptions)

as the query input to search engines. In return, search engines will work on a best effort basis to provide the best search results. Search engines unfamiliar with some descriptors in the query multimedia description may just ignore those descriptors. Others may try to translate them to local descriptors. Furthermore, queries will result in a list of matched multimedia data as well as their MPEG-7 descriptions. Each search engine will also make available the description scheme of its content and maybe even proprietary code.

We envision the connection between the individual search engines and the metaseach engine to be a path for MPEG-7 streams, which will enhance the performance of metasearch engines. In particular, the ability of the proposed description schemes to dynamically download programs for feature extraction and similarity matching by using linking or code embedding will open the door to improved metasearching capabilities. Metasearch engines will use the description schemes of each target search engine to learn about the content and the capabilities of each search engine. This knowledge also enables meaningful queries to the repository, proper decisions to select optimal search engines, efficient ways of merging results from different repositories, and intelligent display of the search results from heterogeneous sources.

6. Discussion

Multimedia search and retrieval involves multiple disciplines, ranging from image processing, computer vision, database, information retrieval, and user interfaces. The content types contained in multimedia data can be very diverse and dynamic. This paper focuses on the multimedia content structuring and searching at different levels. It also addresses the interoperable representation for content description. The impact of the emerging standard, MPEG-7, is also discussed from the perspective of developing metasearch systems.

There are many other important research issues involved in developing a successful multimedia search and retrieval system. We briefly discuss several notable ones here. First, user preference and relevance feedback is very important and has been used to improve the search system performance. Many systems have taken into account user relevance feedback to adapt the query features and retrieval models during the iterated search process [31][32][33][34].

Secondly, content-based visual query poses a challenge due to the rich variety and high dimensionality of features used. Most systems use techniques related to pre-filtering to eliminate unlikely candidates in the initial stage, and to compute the distance of sophisticated features on a reduced set of images [35]. A general discussion of issues related to high-dimensional indexing for multimedia content can be found in [36].

Investigation of search and retrieval for other types of multimedia content has also become increasingly active recently. Emerging search engines include those for music, audio clips, synthetic content, and images in special domains (e.g., medical and remote sensing). Wold *et al* [37] developed a search engine that matches similarities between audio clips based on the feature vectors extracted from both the time and spectral domains. Paquet and Rioux [38] presented a content-based search engine for 3D VRML data. Image search engines specialized for remote sensing applications have been developed in [39][40] with focus on texture-based search tools.

Finally, a very challenging task in multimedia search and retrieval is performance evaluation. The uncertainty of user need, the difficulty in obtaining the ground truth, and the lack of a standard benchmark content set have been the main barriers to developing effective mechanisms for performance evaluation. To address this problem, MPEG-7 recently has incorporated the evaluation process as a mandatory part of the standard development process [41].

7. References

[1] S.-F. Chang, A. Eleftheriadis, Robert McClintock, "Next-Generation Content Representation, Creation and Searching for New Media Applications in Education," Proceedings of the IEEE, special issue on Multimedia Signal Processing, Vol. 86, No. 5, pp.884-904, May 1998.

[2] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain and C. Shu, "Virage image search engine: an open framework for image management", Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV, IS&T/SPIE, Feb. 1996.

[3] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker, "Query by Image and Video Content: The QBIC System," IEEE Computer Magazine, Sep. 1995, Vol.28, No.9, pp. 23-32.

[4] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," Proc. Storage and Retrieval for Image and Video Databases II, Vol. 2185, SPIE, Bellingham, Wash., 1994, pp. 34-47.

[5] J. R. and S.-F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," ACM Multimedia Conference, Boston, MA, Nov. 1996 (Demo http://www.ctr.columbia.edu/VisualSEEk).

[6] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "VideoQ-An Automatic Content-Based Video Search System Using Visual Cues," ACM Multimedia 1997, Seattle, WA, November 1997 (Demo: http://www.ctr.columbia.edu/videoq).

[7] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," ACM SIGRAPH, pp. 277-286, August, 1995.

[8] D. Zhong and S.-F. Chang, "AMOS-An Active System for MPEG-4 Semantic Object Segmentation," IEEE Intern. Conference on Image Processing, October 1998, Chicago, IL.

[9] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System," ACM Multimedia Conference, Boston, MA, Nov. 1996. (demo: http://www.ctr.columbia.edu/webclip)

[10] A. Nagasaka and Y. Tanaka,"Automatic Video Indexing and Full-Video Search for Object Appearances", Visual Database Systems II, 1992

[11] J.R. Smith and S.-F. Chang, "Tools and Techniques for Color Image Retrieval," SPIE Conference on Storage and Retrieval for Image and Video Database, San Jose, Feb. 1996.

[12] M. Swain and D. Ballard, "Color Indexing," International Journal of Computer Vision, &:1, pp. 11-32, 1991.

[13] T. Chang and C.-C. J. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," IEEE Transactions on Image Processing, Vol. 2, No. 4, Oct., 1993.

[14] J.R. Smith and S.-F. Chang, "Transform Features for Texture Classification and Discrimination in Large Image Databases," Proceedings, IEEE 1st Intern. Conf. on Image Processing, Nov. 1994, Austin, Texas.

[15] M. Szummer and R. Picard, "Indoor-Outdoor Image Classification," IEEE International Workshop on Content-Based Access of Image and Video Databases CAIVD '98, Bombay, India, Jan. 1998.

[16] A. Vailaya, A. Jain, and H.J. Zhang, "ON Image Classification: City vs. Landscape," IEEE Workshop on COntent-Based Access of Image and Video Libraries, June 1998, Santa Barbara, CA.

[17] D. Forsyth and M. Fleck, "Body Plans," IEEE Conf. Computer Vision and Pattern Recognition, June 1997, Puerto Rico.

[18] A. Jaimes and S.-F. Chang, "Model Based Image Classification for Content-Based Retrieval," SPIE Conference on Storage and Retrieval for Image and Video Database, Jan. 1999, San Jose, CA.

[19] L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Ch. 5, Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[20] T.T. Kristjansson, B.J. Frey, and T.S. Huang, "Event-coupled Hidden Markov Models," Submitted to *Advanced in Neural Information Processing Systems*, 1998.

[21] M.R. Naphade, T.T. Kristjansson, B.J. Frey, and T.S. Huang, "Probabilistic Multimedia Objects (Multijects): A Novel Approach to Video Indexing and Retrieval in Multimedia Systems," IEEE International Conference on Image Processing, Oct. 1998, Chicago, IL.

[22] G.E. Hinton and T.J. Sejnowski, "Learning and Relearning in Boltzmann Machines," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition (D.E. Rumelhart and J.L. Mc-Clelland, eds.), vol. I, p. 282-317, Cambridge MA: MIT Press, 1986.

[23] B.J. Frey, Graphic Models for Machine Learning and Digital Communication. Cambridge MA: MIT Press, 1998.

[24] M.I. Jordab, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul, "An Introduction to Variational Methods for Graphic Models," In *Learning and Inference in Graphic Models* (M.I. Jordan, ed.), Norwell MA: Kluwer Academic Publishers, 1998.

[25] ISO/IEC JTC1/SC29/WG11 N1909, MPEG-4 Version 1 Overview, October 1997.

[26] S.-F. Chang, W. Chen, and H. Sundaram, "Semantic Visual Templates - Linking Visual Features to Semantics," IEEE Intern. Conference on Image Processing, Chicago IL, October 1998.

[27] MPEG-7's Call for Proposals for Multimedia Content Description Interface, http://drago.cselt.it/mpeg/public/mpeg-7_cfp.htm

[28] A. B. Benitez, M. Beigi, and S.-F. Chang, "Using Relevance Feedback in Content-Based Image Metasearch," IEEE Internet Computing Magazine, Vol. 2, No. 4, pp. 59-69, July, 1998. (demo: http://www.ctr.columbia.edu/MetaSEEk)

[29] World Wide Web Consortium's (W3C) XML web site http://www.w3.org/XML.

[30] S. Paek, A.B. Benitez, and S.-F. Chang, "Self-Describing Schemes for Interoperable MPEG-7 Multimedia Content Descriptions," SPIE VCIP'99, Visual Communication and Image Processing, San Jose, CA, Jan. 1999.

[31] J. Huang, S.R. Kumar, and M. Mitra, "Combining Supervised Learning with Color Correlograms for Content Based Image Retrieval," ACM Multimedia '97, Seattle, WA, Nov. 1997.

[32] T.P. Minka and R. Picard, "Interactive Learning Using a "Society of Models"," MIT Media Lab Perceptual Computing Section Technical Report 349.

[33] Y. Rui, T. Huang, S. Mehrotra, and M. Ortega, "A Relevance Feedback Architecture for Content-Based Multimedia Information Retrieval Systems," CVPR'97 Workshop on Content-Based Image and Video Library Access, June 1997.

[34] I.J. Cox, M.L. Miller, S.M. Omohundro, and P.Y. Yianilos, "The PicHunter Bayesian Multimedia Retrieval System," ADL '96, Forum, Washington D.C., May 1996.

[35] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack ,"Efficient Color Histogram Indexing for Quadratic Form Distance Functions", IEEE Trans. PAMI, July, 1995.

[36] C. Faloutsos, Searching Multimedia Databases by Content, Kluwer Academic Publishers, 1997.

[37] E. Wold, T. Blum, D. Keislar, and J. Wheaton "Content-Based Classification, Search, and Retrieval of Audio," IEEE Multimedia Magazine, Vol. 3, No. 3: FALL 1996, pp. 27-36.

[38] E. Paquet and M. Rioux, " A content-based search engine for VRML databases," Proceedings of the 1998 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). Santa Barbara, CA. June, 1998. pp. 541-546.

[39] L. Bergman, V. Castelli, and C.-S. Li, "Progressive Content-Based Retrieval from Satellite Image Archives," CNRI Digital Library Magazine (on-line), Oct. 1997. (http://www.dlib.org)

[40] W.Y. Ma and B.S. Manjunath, "Texture Features for Browsing and Retrieval of Image Data," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, pp.837-842, Aug. 1996.

[41] MPEG Requirements Group, "MPEG-7 Evaluation Procedure", Doc. ISO/MPEG N2463, MPEG Atlantic City Meeting, October 1998

[42] D. Zhong and S.-F. Chang, "AMOS-An Active System for MPEG-4 Semantic Object Segmentation," IEEE Intern. Conference on Image Processing, October 1998, Chicago, IL.

[43] D. Zhong and S.-F. Chang, "An Integrated System for Content-Based Video Object Segmentation and Retrieval, part II: Content-Based Searching of Video Objects," submitted to IEEE Transactions on Circuits and Systems for Video Technology, 1998.

[44] H. Zhang, A. Kankanhalli, and S. Smoliar, "Automatic Partitioning of Full-motion Video," A Guided Tour of Multimedia Systems and Applications, IEEE Computer Society Press, 1995.

[45] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques, *Proceedings of SPIE Conf. on Storage and Retrieval for Still Image and Video Databases IV*, SPIE Vol. 2670, pp.170-179, San Jose, Feb. 1996.

[46] A. Hauptmann and M. Witbrock, "Story Segmentation and Detection of Commercials in Broadcast News Video," Proc. of Advances in Digital Libraries Conference, Santa Barbara, April, 1998.

[47] Y.L.Chang, W. Zeng, I. Kamel, and R. Alonso, "Integrated Image and Speech Analysis for Content-Based Video Indexing," Proc. of Multimedia, Sept. 1996, pp. 306-313.

[48] J. Nam and A. H. Tewfik, "Combined Audio and Visual Streams Analysis for Video Sequence Segmentation," Porceedings of International Conference on Acoustic, Speech, and Signal Processing, Vol. 4, 1997, pp. 2665-2668.

[49] M. A. Hearst, "Multi-Paragrah Segmentation of Expository Text," The 32nd Annual Meeting of the Association For Computational Linguistics, New Mexico, June, 1994.

[50] M.G. Brown, J. Foote, G.J.F. Jones, K.S. Jones, and S.J. Young "Automatic Centent-Based Retrieval of Broadcast News," Proceedings of ACM Multimedia Conference, San Francisco, 1995, pp. 35-42.

[51] M. Yeung and B.-L. Yeo, "Time-Constrained Clustering for Segmentation of Video Into Story Units," Proceedings of International Conference on Pattern Recognition, Vienna, Austria, August, 1996, pp. 375-380.

[52] M. Yeung and B.-L. Yeo and B. Liu, "Extracting Story Units From Long Programs For Video Browsing and Navigation," Proceedings of International Conference on Multimedia Computing and Systems, June, 1996.

[53] Y. Rui and T.S. Huang and S. Mehrotra, "Constructing Table-of-Content For Videos," ACM Journal of Multimedia Systems, 1998.

[54] M. Maybury, M. Merlino, and J. Rayson, "Segmentation, Content Extraction and Visualization of Broadcast News Video Using Multistream Analysis," Proceedings of ACM Multimedia Conference, Boston, 1996.

[55] I. Mani, D. House, D. Maybury, and M. Green, "Towards Content-Based Browsing of Broadcast News Video," Intelligent Multimedia Information Retrieval, 1997.

[56] A. Merlino, D. Morey, and D. Maybury, "Broadcast News Navigation Using Story Segmentation," in *Proc. of ACM Multimedia*, Nov. 1997.

[57] Q. Huang, Z. Liu, and A. Rosenberg, "Automated Semantic Structure Reconstruction and Representation Generation for Broadcast News ," *Proc. SPIE, Storage and Retrieval for Image and Video Databases VII,* San Jose CA, Jan. 1999.

[58] World Wide Web Consortium's (W3C) XML web site http://www.w3.org/XML.

[59] S. Paek, A. B. Benitez, S.-F. Chang, C-S. Li, J. R. Smith, L. D. Bergman, A. Puri, "Proposal for MPEG-7 Image Description Scheme", **ISO/IEC** JTC1/SC29/WG11 MPEG98/P480 MPEG document, Lancaster, UK, Feb. 1999.

[60] S. Paek, A. B. Benitez, S.-F. Chang, A. Eleftheriadis, A. Puri, Q. Huang, C-S. Li, J. R. Smith, and L. D. Bergman, "Proposal for MPEG-7 Video Description Scheme", **ISO/IEC** JTC1/SC29/WG11 MPEG98/P481 MPEG document, Lancaster, UK, Feb. 1999.

[61] Q. Huang, A. Puri, A. B. Benitez, S. Paek, and S.-F. Chang, " Proposal for MPEG-7 Integration Description Scheme for Multimedia Content", **ISO/IEC** JTC1/SC29/WG11 MPEG98/P477 MPEG document, Lancaster, UK, Feb. 1999.