Video Query Formulation[1]

Gulrukh Ahanger, Dan Benson[†], and T.D.C. Little

Multimedia Communications Laboratory
Department of Electrical, Computer and Systems Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877, (617) 353-6440 fax
{*gulrukh,tdcl*}*@bu.edu*

[†]Image & Video Management
Siemens Corporate Research, 755 College Road East
Princeton, New Jersey 08540, USA
(609) 734-3668, (609) 734-6565 fax
dbenson@scr.siemens.com

MCL Technical Report 01-09-1995

Abstract–For developing advanced query formulation methods for general multimedia data, we describe the issues related to video data. We distinguish between the requirements for image retrieval and video retrieval by identifying queryable attributes unique to video data, namely audio, temporal structure, motion, and events. Our approach is based on visual query methods to describe predicates interactively while providing feedback that is as similar as possible to the video data. An initial prototype of our visual query system for video data is presented.

Keywords: Video databases, motion query, query formulation.

[1]In *Proc. Storage and Retrieval for Images and Video Databases III, IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, Vol. 2420, February 1995, pp. 280-291.

1 Introduction

With rapidly decreasing costs of storage, advancements in compression techniques, and higher transmission rates, the arrival of very large digital video databases is becoming increasingly imminent. Existing database technology is not designed to manage digital video as "first class" media. By this we mean that very little support is available for indexing and querying video based on its content. To be able to do this, it is necessary to extract distinguishing features that will allow access to finer granularities of the video content. To treat video data as more than named BLOBs (Binary Large Objects), a database system must cover the range of tasks associated with the management of video content including feature extraction, indexing, querying, browsing, and developing representation schemes and operators.

The interactive query process consists of three basic steps: formulating the query, processing the query, and viewing the results returned from the query. This requires an expressive method of conveying what is desired, the ability to match what is expressed with what is there, and ways to evaluate the out come of the search. Conventional text-based query methods that rely on keyword look-up and string pattern-matching are not adequate for all types of data, particularly auditory and visual data. Therefore, it is not reasonable to assume that all types of multimedia data can be described sufficiently with words alone, not as meta-data when it is first entered in the database, nor as queries when it is to be retrieved. In this paper, we focus primarily on the retrieval of video data, although all the techniques are applicable to the retrieval of other types of multimedia data.

A video database system can contain tens of thousands of video sequences. The ability to quickly and easily access and retrieve a target item is critical. If multimedia databases are to achieve widespread commercial use, query formulation, retrieval, evaluation, and navigation techniques that support multiple types of data must be developed. Some examples of areas where this potential may be realized include: medical research in dermatology, analysis of tumor characteristics, particle and material identification for forensic, cataloging and comparative studies for museums and archaeology, scientific taxonomy and classification in botany and zoology, security systems, analysis of motion of spatial bodies to understand physical phenomena, and prediction of weather conditions based partly on the movement of pressure systems and storms. In all these fields the information filtering is based on the content of the video data. For example, if a user wants to retrieve videos of a particular team playing from a vast database of sports videos, a query might be formulated by combining the colors of the team's uniform with patterns that resemble the various motions of the game.

While not abandoning the use of conventional text-based methods, we concentrate on visual means of expressing and constructing queries based on attributes that are otherwise impossible to express textually. Since our work centers on video data retrieval, we begin (Section 2) with a review of recent work in the area of image and video retrieval. In Section 3 we identify queryable attributes unique to video data. From this set of features, we present the issues and requirements related to query formulation in Section 4. This is followed in Section 5 by an example of our initial prototype video retrieval system MovEase (Motion Video Attribute Selector). We conclude in Section 6 with a summary and discussion of future work.

2 Related Work

Several multimedia retrieval systems are based on retrieving data from images databases [5, 8, 12, 13, 17], whereas very few systems have been developed for retrieving video data. All the properties inherent to image data are also part of video data; therefore, working with image databases may be thought as a stepping stone towards retrieval from video databases. Several techniques have been proposed for retrieval of multimedia data using visual methods. Most of the systems fall mainly under two categories, (QBE) Query by Example, and (IQ) Iconic Query. QBE queries are formulated using sample images, rough sketches, or component feature of an image (outline of objects, color, texture, shape, layout). These systems make extensive use of image processing and pattern recognition techniques. Some examples of QBE systems include:

- IMAID [8] is an integrated image analysis and image database management system. By using pattern recognition and image processing manipulation functions, pictorial descriptions can be extracted from example images. When a user requests a description, all pictures satisfying the selection criteria are retrieved and processing is continued until the desired precision is achieved.

- ART MUSEUM [12] is based on the notion that a user's visual impression of a painting consists of the painting's basic composition. Images of paintings are entered into the database via an image scanner and the system automatically derives their abstract images and adds them to the pictorial index. The user draws a rough "sketch" of the image to be retrieved.

- QBIC [17] is a more flexible system compared to the previous two. Thumbnail images

are stored in the database along with text information. Object identification is performed based on the images and the objects. Features describing their color, texture, shape, and layout are computed and stored. The queries are run on these computed features. Individual features or a combination of features can be used in query formulation. The user can also apply objects in the queries by drawing outlines around the object in an image. Any number of objects are permitted in an image, disconnected or overlapping.

Retrieval systems based on IQ [6, 9, 14, 19] take advantage of a user's familiarity with the world. An icon represents an entity/object in the world, and users can easily recognize the object from the icon image. Query is formulated by selecting the icons that are arranged in relational or hierarchical classes. A user refines a query by traversing through these classes. Iconic queries reduce the flexibility in a query formulation as queries can only utilize the icons provided, i.e., iconic databases tend to be rigid in their structure. In fact, most systems do not allow the user to create customized icons or to create icons on the fly. Some examples of this type of system include:

- The 3D iconic environment for image database [5], developed on the principal that 2D-string representation of spatial data may not exactly represent the spatial relationship. Therefore, the spatial relationship is considered in 3D image scenes. A presentation language is described that aids in expressing position and directional relationship between objects, while still preserving spatial extension after projection.

- Virtual Video Browser [14] (VVB), an iconic movie browser. Icons represent different categories of movies and subsequently the movies available within these categories are represented by icons. Movies and scenes of movies can also be identified and viewed through the VVB, based on movie-specific attributes including actor names, director names, and scene characteristics. Movie and scene contents are facilitated by summary, keyword, and transcript searching stored in a meta-database.

- Video Database Browser [19] (VDB), a mixed-mode query interface which allows selection of desired videos/frame sequences. It utilizes a data model developed on the basis of sample queries (collected by surveying a variety of users). The selection process occurs by incremental reduction on a set of videos until the desired set of videos is obtained. The interface includes general relational operators, navigational operators for structural queries, and keyword search operators.

- Media Streams [9], based on an iconic visual language and stream-based representation of video data. This system allows users to create multi-layered, iconic annotations of video content. Icon palettes enable users to group related sets of iconic descriptors, use these descriptors to annotate video content, and reuse the effort. Media time lines is used to visualize and browse the structure of video content and its annotations.

The systems that retrieve images or video data based on feature components make extensive use of on-the-fly image processing techniques [5, 8, 11, 12, 13, 17]. These techniques are not suitable for very large collections of video, as they require a great deal of computational power and processing time. VVB is a domain specific application, it retrieves videos based on bibliographic data (title, director, producer, actor, etc.). VDB is modeled on frequently asked questions which limits the flexibility of query formulation. Moreover, the query by content is limited to scripts, objects, actors, and keyframes. Media Streams retrieves video data on the basis of their content using manual, semi-automatic, and automatic annotations. It also incorporates motion and audio as query attributes, but the motion information is very general, e.g., a person walking or talking.

An efficient video retrieval system should provide flexible and easy to use ways of formulating the query. Such a system should make effective and appropriate use of the video data attributes. In the next section we discuss the attributes of video data that can be incorporated for efficient data retrieval.

3 Video Attributes

Each database record, or data object, must have distinctive features to distinguish it from other data objects. This is required for proper organization within the database as well as retrieval based on properties of the data. Since video data consists of sequences of images, they share all the attributes of image data such as color, shape, objects, positions of objects, relative layouts, and texture. Unlike image data, video data have additional temporal and relational attributes and, in most cases, video data are hierarchical in structure, i.e., contiguous frames form shots, groups of shots form scenes. A given shot can be combined with a different subset of shots to produce a different semantic scene. Another difference between image and video data is the sheer volume of data that makes up a video sequence. A typical video can contain thousands of individual frames. Just as an image may have sub regions-of-interest, a video sequence can have sub segments-of-interest. On the physical level, each

video sequence can be described in terms of its frame size and intensity values (width, height, depth). It also has a length that can be specified in terms of its total number of frames or in terms of its total viewing time (e.g., 90,000 frames @ 15 frames per second). In terms of its content, a video sequence can contain practically anything. The items of interest are usually considered as objects and the spatial and semantic relationships between the objects. Video content has the additional properties of implied motion, sequential composition, inter- and intra-frame temporal relationships, and (possibly) synchronized audio signals.

The implied motion in image sequences can be attributed to a camera (global) motion and an object (local) motion. A camera has six degrees of freedom representing translation along each axis (x : track, y : boom, z : dolly) and rotation about each axis (x : tilt, y : pan, z : rotate). In addition, change in the camera's focal length produces scaling or magnification of the image plane (zoom in, zoom out). These basic camera operations are illustrated in Fig. 1, with the camera situated at the origin, the image plane coincident with the xy plane, and the optical axis along the z axis.

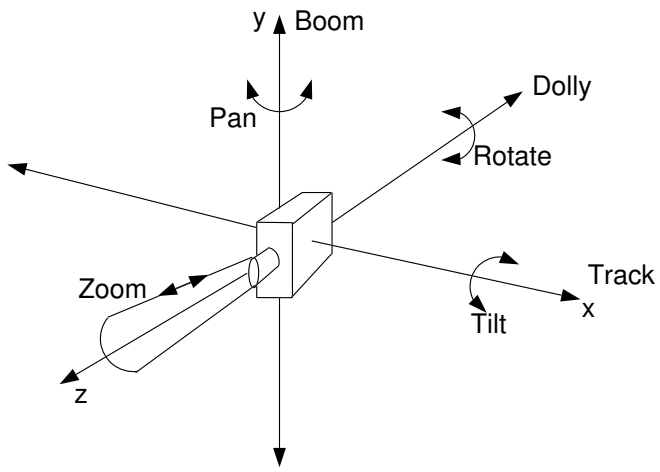

Figure 1: Basic Camera Operations

In general, video data can contain independently moving objects in three-dimensional space. Object motion typically refers to changes in an object's location, orientation, trajectory, velocity, or shape. From an image analysis point of view, objects may be subdivided according to the form of body (formation), namely:

- Rigid – the distance between two points on the object remains the same even if the object undergoes a transformation, e.g., a block of wood.

- Articulated – comprised of two or more rigid objects, e.g., a pair of scissors.
- Non-rigid – deformable or flexible objects, e.g., snake, clay, fluid, etc.

Though the combination of camera motion and object motion produces the recorded scenes, it is necessary to separate the two in order to capture the apparent motion of objects, that is, the "real" motion as perceived by the viewer. In other words, when watching a scene in a video, most viewers interpret what they see as objects moving in a three-dimensional space, often unaware of camera movement. For example, a person who watched a speed skating event might describe one of the closeup scenes as "skaters skating towards the right" (of the screen). In fact, coverage of the speed skating might employ a moving camera focused on the skaters down the long stretch. From the camera's point of view the skaters remained in the center of the screen during the shot, but from the viewer's point of view the skaters were moving along the ice. In this example, it would be possible to simply subtract the camera's motion (tracking) to derive the object's (skaters) apparent motion (from left to right), thereby obtaining the perceived motion of objects in the scene.

Another reason to separate camera from object motion is to capture camera motion explicitly. This is useful for retrieving shots in which particular types of camera operations were performed. For instance, someone studying cinematography might want to retrieve all the scenes of tracking, or scenes of booming while panning, or tilting followed by zooming. We classify camera and object motion in a motion hierarchy as shown in Fig. 2. This serves as a representation schema for the video database and facilitates development of query formulation methods.

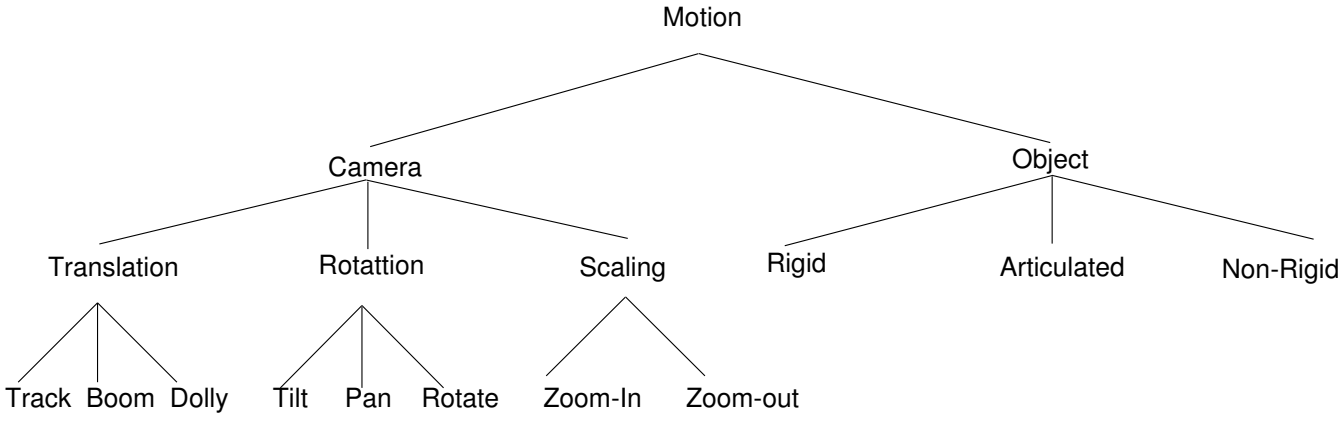

Figure 2: Motion Classification Hierarchy

By sequential composition we refer to the ordering of segments that make up a complete video sequence. For example, a nightly news broadcast will typically follow a common pattern of edited segments, such as headlines, national news, local news, sports, weather, and commentary. This pattern of segments describes the sequential composition attributes of a recorded news broadcast. Because a video sequence consists of several image frames, there are typically many frames that contain similar objects. Therefore, there is likely to be inter-frame relationships within a video sequence, such as non-contiguous segments containing common objects. In a large collection of video sequences there is also likely to exist inter-sequence relationships, and so on. When audio information is a part of video data it provides a valuable attribute that can also be used to describe and distinguish video content. Just as image sequences can be analyzed into objects, patterns, and relationships, audio signals can be analyzed into sounds, words, and so on.

In our video query formulation system, MovEase (Motion Video Attribute Selector), we combine several video content attributes including color, shape, objects, positions, relative layouts, and texture. Discussions of these types of attributes can be found in recent publications e.g., shape, size, texture, and layout [17], B-spline curve representations for motion [10], and color histograms for an object identification [21, 23]. Most importantly, we also treat motion as an explicit attribute, both camera motion and object motion. For this reason, we concentrate primarily on the issues regarding motion as a queryable attribute. Video attributes can be generated manually, programatically, or semi-automatically using various techniques. Regarding motion, there have been several methods reported for video segmentation [3, 16, 18, 24], extraction of camera operations[1, 15], and rigid object movement [7]. For the purposes of this paper, we assume that the attributes have already been generated.

4 Query Formulation

Perhaps the most common operation on any database is retrieval of information. Therefore, we need efficient and accurate methods for retrieving data. Queries can be formulated using several techniques. These techniques fall broadly into two categories: textual and visual. Text can be used to formulate queries for visual data (images, video, graphs), but such queries are not very efficient and cannot encompass the hierarchical, semantic, spatial, and motion information. Visual data can also be very difficult to describe adequately by keywords as keys are chosen by each user based on his or her impression of the image and video. Thus it is difficult, if not impossible to know under what keyword the target has been indexed.

In addition, keywords must be entered manually, are time consuming, error prone, and thus cost prohibitive for large databases. A query system that allows retrieval and evaluation of multimedia data should be highly interactive to facilitate easy construction and refinement of queries. Due to the visual nature of the data, a user may be interested in results that are similar to the query, thus, the query system should be able to perform exact as well as partial or fuzzy matching. The results of a query should be displayed in a decreasing order of similarity, from the best match to the nth based matched result for easy browsing and information filtering. Due to the complex nature of video queries, there should be a facility to allow a user to construct queries based on previous queries.

As mentioned in Section 2 several systems have been developed to retrieve visual data based on color, shape, size, texture, image segments, keyword, relational operators, objects, and bibliographic data, but little attention has been paid to the use of object and camera motion information found in video data. Describing different types of motion, particular paths of object movement, and combinations of motions are not adequately expressed using only keywords and relational operators. For these reasons, we explore visual means of query formulation. We take advantage of the powerful human visual perception by using a visual approach to query formulation. This provides the closest way of describing visual information interactively while providing immediate feedback for relevance and verification.

Most of the work done in motion analysis utilizes motion information in various image processing tasks such as segmentation, pattern recognition and tracking, structure, and scene interpretation [2, 4, 20, 22]. Our interest in the use of motion differs from conventional motion analysis in three ways. First, we consider motion itself as a primary attribute to be used in organizing and retrieving video data. This is in contrast to using motion primarily as a means to compute other information. Second, we do not require the same degree of precision and accuracy in the motion information extracted. We are primarily interested in the general motion of objects, in the sense of how they might be conceived or described by a person viewing the video. And third, rather than analyzing sequences of only two or three images in a fixed domain, we seek robust methods of acquiring motion information that can be applied to long sequences containing a wide variety of video content.

Motion queries can be formulated by specifying any combination of camera motions, objects and their motion paths, and whether the motion is domain dependent or independent. Therefore, there can be varying levels of motion complexities in specification of a motion query. A motion query may consist of an object running parallel to the camera and the camera panning at the same time, camera zooming into an object, the camera tilting and

dollying at the same time, an object moving orthogonally to the camera. As a video retrieval system, in addition to executing queries using features inherent to images, MovEase allows a user to specify camera motions and associate motion information with each object (path, speed).

A single object, or multiple objects, can be specified in a query. If multiple objects of the same type with similar motion attributes are specified in the same query then these independent objects can be grouped. The grouped objects can be perceived as one (optical flow), consider a mass of people moving in a certain direction or a crowded highway. Grouping the objects reduces the flexibility of associating types of motions with an object but at the same time reduces the complexity of handling multiple objects in a query. Object motion can be domain independent or dependent. If the motion is domain independent the retrieval mechanism looks for an object that is similar to the object specified but might be rotated or scaled up/down, the path of an object may not be necessarily fixed, an object can follow the same path anywhere in a sequence. For example, in a sequence of a boy running: the boy can be running in, across the middle, top, bottom, or diagonally across the sequence. Domain dependent motion restricts the path translation but the object can still be rotated or scaled. Scenes can be composed of a single camera motion or combination of camera motions (pan, tilt, zooming while panning etc.).

Next, we describe our initial prototype of a video query formulation system that is based on the above discussions.

5 Motion Video Attribute Selector

The main objective of developing MovEase was to consider motion as a primary attribute in retrieving video data. A user is able to specify information related to a sequence of frames rather than just two or three frames, therefore, more information is provided for the system to extract video data. MovEase is also designed to incorporate other types of video data features but we concentrate here on the motion query aspects.

Raw video data is annotated off-line (manual, or semi-automatically) and then stored. This annotated data are represented by icons that are utilized in query formulation. As video data is massive, so are the number of icons representing the data. Only the icons of the data actually present in the database are stored. If the database is extended so is the set of icons. These icons represent objects, textures/patterns, actions (talking, eating,

fighting), and shapes. For easy accessibility and better management, these icons are divided into classes, and the user can retrieve icons based on these classes, e.g., car, bike, train, plane, and ship belong to the class "Transport."

Users might execute certain queries frequently, so instead of formulating the same query repeatedly, we introduce query re-use. A previously formulated query can be stored and represented by an icon. These queries can be used as they are or can be modified and used, such that the user need not start from scratch. Some of the generic motions are represented by icons, i.e., if a user wants to retrieve scenes in which the camera is panning and the degree of panning is not of importance, then just including the information that the camera is panning will suffice. A user might create some complicated motion scenarios; these too can be stored and later incorporated into subsequent queries. Therefore, we require an icon catalog manager to manage object, motion, and query catalogs. It is extremely difficult to foresee all possible query scenarios, therefore, we make it possible to annotate data on-line. An object can be extracted from the video and an icon is created for the object and stored in the icon database. The video data can be annotated on-line which is a very costly process in terms of resources or it can be done off-line when the system is not being used. Fig. 3 illustrates the general query information flow in the system. In this system, because motion is a dynamic action, we provide a preview feedback to simulate the described motion before submitting a query to the system.

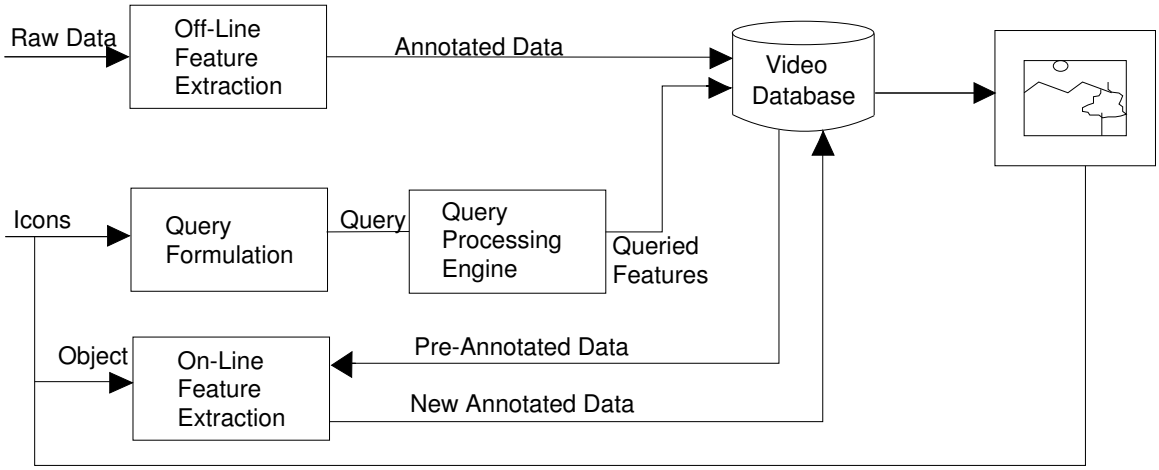

Figure 3: Query Flow Diagram

In the rest of this section we describe the details of the MovEase operations.

5.1 MovEase Operation

The interface is divided into three regions: work area or query builder, icon catalog browser, and the result browser. A query is composed in the query builder, the contents of the object, motion, or query catalog are viewed in the icon catalog browser, and the results of a query (the data retrieved from the database) can be evaluated in the result browser.

In the object catalog, the icons are displayed hierarchically as thumbnail images. A user can move deeper into the class hierarchy by clicking on the icons and further subclasses (thumbnail) will be displayed (if any). A user can also magnify the thumbnail images for better view. In the motion catalog, the motions are also displayed as thumbnails, but the motion can be previewed by clicking on the motion icon which displays the motion simulation in the motion specification/viewing window. As mentioned above, previously formulated queries can be stored for later use, the query is stored as the complete string and represented by an icon. The icon is identified by the name given by the user. The user can mouse over these query icons to see a list of objects in the query or to see detailed information that can be displayed in the query builder. Once the query is displayed in the query builder it can be edited for any changes needed. For example, consider a user who wants to retrieve all the segments with green trees in them, this query could be formulated in following steps:

- Browse through the object catalog and select an icon of a "tree" from class Plants.
- Open the color pallete by choosing "Create Color" and select the desired color for the objects, if the user clicks on the "tree" icon while color pallete is open, then the color gets automatically associated with the tree. The system will then ask the user for exact or fuzzy color match, for fuzzy match the user can specify the tolerance for variance.
- Click on "Execute Query."

Finally, assume the user saved aforementioned query with a name "summer-forest." Later on, if the user wanted to retrieve clips which have people climbing the hills, then "summer-forest" can be reused by incorporating people into it and associating uphill motion with them.

On selection an icon automatically gets pasted in the work area for use in the query. If the query is specified as domain dependent, i.e., invariant to rotation, translation, and scaling then the positioning of the objects in the work area matters, otherwise, objects can

be placed anywhere. To further refine the query (to reduce the target domain) bibliographic data (title, director, producer, year, location, keyword) can also be incorporated in the query.

In this system we also assume that most of the queries will be fuzzy in nature, they may not represent actual scenarios but only the closest semblance of what the user wants. Therefore, instead of specifying the exact relative layout between the objects, generic directions will suffice e.g., left, right, front-left. The same applies for specifying color, default or customized colors can be selected from the color palette and the user can specify acceptable degrees of variance. Default textures are provided as icons but a user has an option of creating new ones using the drawing tool, this tool can also be used for drawing shapes and extracting objects from video frames. Fig. 4 shows an example of a query that retrieves video clips that have a bike to the left of the car and both are in motion (relative layout generic motion). Some more examples of the queries that can be formulated are: the user can ask the system to show a clip with large crowd at the Washington Monument (shape, object), show demonstrators marching with signs that are covered in reddish shade(fuzzy color, texture), or show speeches where the speaker is in front of a huge red, white, blue flag (color, object).

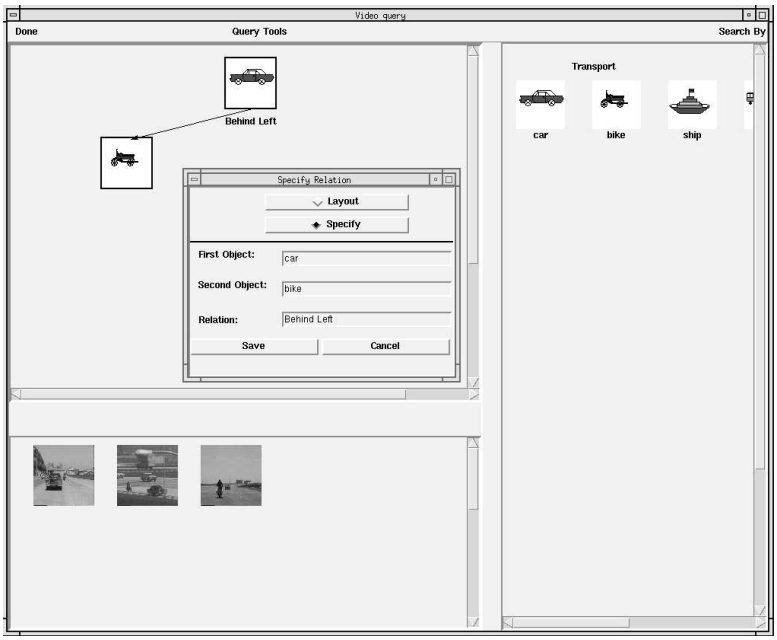

Figure 4: Query Formulation Screen

The system has provision for on-the-fly indexing, i.e., if a user notices an object in a video clip that is not in the database then this object can be indexed. The frame can be loaded in

the "Edit Tool" and the object cut out from the frame. A user can submit this picture to the database for on-line indexing. This technique will require heavy use of computer resources, therefore, the indexing can also be done off-line. The icon for this object will be represented by the cut object picture. Currently, this feature is under development.

We can associate generic or specific motion with the objects. For example, if we query for an object that is in motion irrespective of the type of motion then we simply associate generic motion with it, but if we want an object following a zig-zag path then we can specify this particular motion. In the motion query specifier/viewer the user can specify the motion of multiple objects in three dimensional space with respect to the camera, the z axis being orthogonal to the camera. An object path can be specified along with the information whether the object is invariant to translation or not, i.e., the object should be in a particular section of the frame or anywhere in the frame.

So far the object motions covered are pure translation, rotation with translation, and scaling & rotation with translation. In our initial system, only two camera motions "pan" and "zoom" can be incorporated in the motion query formulation. Various degrees of pan and zoom can also be specified (25 percent pan, or 50 percent zoom). An object is represented by a box and increasing or decreasing the size of the box (object) specifies the distance of the object from the camera. Only rigid objects are considered in the query, the articulated and non-rigid objects are represented by their center of mass. Three types of object motions are represented by icons that aid in drawing the motion of an object. First, an object moving orthogonally towards the camera, called "toward" icon. Second, object moving away from the camera, called "away" icon. Both, first and second type of motions are for objects getting displaced along the z co-ordinate. Third, object moving along all or any of the three co-ordinates (x, y, and z co-ordinates) called "3D" icon. There are also icons representing camera pan and zoom. Fig. 5 shows an example of a motion query in which a person is doing a down hill slalom with the camera panning 50 percent at the same time. The query was formulated by drawing the path of an object with the "3D" icon and then selecting the "pan" icon. The user can choose the degree of pan from the "pan" button at the bottom of the screen. The speed of the object can also be varied by a slider at the bottom of the screen. The query can be viewed by animating it, such that the user can get instant feedback and can make modifications to the query.

The video browser section is used to display the results of the queries. Thumbnail size of the first frame of retrieved data is displayed and we call it VICON (Video ICON). As a result of a fuzzy query more than one video segment might be retrieved; therefore, VICONs

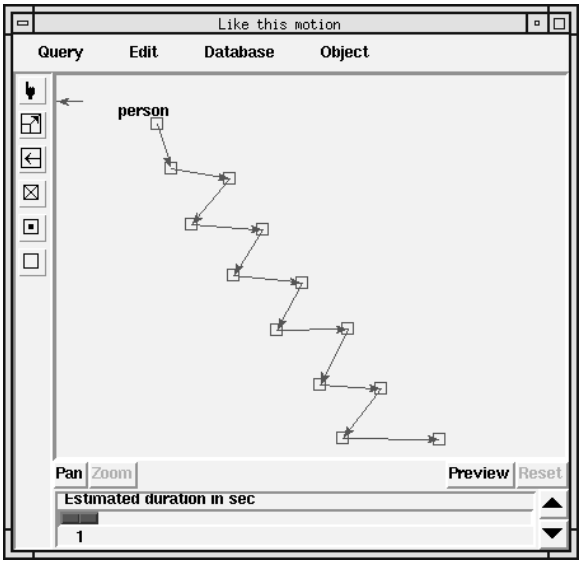

Figure 5: Motion Query Formulation Screen

are ordered in descending order of match. By clicking on a VICON, detailed information about the video segment gets displayed. The information includes, title, producer, director, year of the video to which the segment belongs, and the range of frames contained in the segment. The user can view the segment, give ranges of frames to view, or the entire video. There are many functionalities associated with the video player, e.g., looping, jumping to random frames, fast-forward, etc. Fig. 6 shows an example of people moving in a wooded area.

6 Summary and Future Work

Large video databases can benefit from existing work on image databases but additional accommodations must be made for the unique characteristics of video data. Namely, the large number of image sequences, the existence of synchronized audio information, and the existence of motion must be dealt with. Motion has been used in image analysis largely as a means of extracting other information such as segmentation of objects. In this paper, we introduce the use of motion as a primary feature to distinguish segments of video. We outlined the research issues to be addressed in achieving more robust support for motion data in large video database and presented some of our initial work in motion query formulation including our prototype system called MovEase.

Figure 6: Query Formulation Screen with Video Player

MovEase demonstrates that our techniques are viable for the formulation of queries that cannot be expressed textually. Specifying camera and object motion in queries helps retrieval of data on a temporal basis. Instead of specifying the contents of each frame we can specify a sequence of frames; therefore, it is fast and requires fewer details. Incorporating specific (path dependent) motion information about the camera and objects in a query provides an additional dimension to content-based retrieval. By providing a mechanism to animate the motion query we see that users are more readily able to immediately evaluate and rectify the query. Therefore, animation provides an intermediate check to the effectiveness of the query. Furthermore, the use and organization of icons into classes and subclasses provides another simplification for the user.

From our assessment, MovEase represents the first step towards the general use of motion in video database systems. As we proceed in this area, we are motivated by a longer-term goal, to transform these motion primitives onto higher-level actions and events. Rather than simple movements (e.g., an object moving left-to-right), a person viewing a video segment would more-likely choose to describe a scene in terms of actions (e.g., a person waving) or events (e.g., a person leaving the room). An interesting approach to conceptual descriptions of events take low-level motion primitives and builds successively higher-level descriptions of objects and events. For example, repeated back-and-forth movements might be recognized as

"rocking" or "swinging" actions. Given more domain knowledge, this could be further refined into a higher-level description of a hand waving goodbye. This level of video understanding is much closer to a human's conception of video contents and will require much more advanced processing of several visual features of which motion information is an essential component.

7 Acknowledgments

We gratefully acknowledge Arding Hsu and Farshid Arman of Siemens Corporate Research for their contributions during the initial stage of this research work.

References

[1] A. Akutsu, and Y. Tonomura, "Video Tomography: An Efficient Method for Camerawork Extraction and Motion Analysis," *Proc. ACM Multimedia'94*, San Francisco, CA, pp. 349-356, 1994.

[2] P. Anandan, J. R. Bergen, and K. J. Hanna, "Hierarchical Model-Based Motion Estimation," *Motion Analysis and Image Sequence Processing*, M. Sezan & R. Lagendijk, eds., Kluwer Academic Publishers, Norwell, MA, pp. 1-22, 1993.

[3] F. Arman, A. Hsu, and M-Y. Chiu, "Image processing on encoded video sequences," *Multimedia Systems Journal*, Vol. 1, pp. 211-219, 1994.

[4] N.I. Badler, "Temporal Scene Analysis: Conceptual Descriptions of Object Movements," *Ph.D. Thesis*, Technical Report No. 80, Department of Computer Science, University of Toronto, p. 224, 1975.

[5] A.D. Bimbo, M. Campanai, P. Nesi, "A Three-Dimensional Iconic Environment for Image Database Querying," *IEEE Trans. on Software Engineering*, Vol. 19, No. 10, pp. 997-1011, 1993.

[6] G. Bordogna, I. Gagliardi, D. Merelli, P. Mussio, M. Padula, and M. Protti, "Iconic Queries on Pictorial Data," *Proc. IEEE Workshop on Visual Languages*, pp. 38-42, October 1989.

[7] P. Bouthemy, and E. Francois, "Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence," *Intl. journal of Computer Vision*, Vol. 10, No. 2, pp. 157-182, 1993.

[8] N.S. Chang, and K.S. Fu, "Picture Query Languages for Pictorial Data-Base Systems," *Computer*, Vol. 14, No. 11, pp. 23-33, November 1981.

[9] M. Davis, "Media Stream: An Iconic Language for Video Annotation," *Proc. IEEE Symposium on Visual Languages*, Bergen, Norway, pp. 196-202, 1993.

[10] N. Dimitrova, and F. Golshani, "R_x for Semantic Video Database Retrieval," *Proc. ACM Multimedia'94*, pp. 219-226, San Francisco, October 1994.

[11] T. Hamano, "A Similarity Retrieval Method for Image Databases Using Simple Graphics," *IEEE Workshop on Languages for Automation, Symbiotic and Intelligent Robotics*, University of Maryland, pp. 149-154, August 1988.

[12] K. Hirata, and T. Kato, Query By Visual Example, *Proc. Third Intl. conf. on Extending Database Technology*, Viennna, Austria, pp. 56-71, March 1992.

[13] T. Joseph, and A.F. Cardenas, "PICQUERY: A High Level Query Language for Pictorial Database Management," *IEEE Trans. on Software Engineering*, Vol. 14, No. 5, pp. 630-638, May 1988.

[14] T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, F.W. Reeves, D.H. Schelleng, and D. Venkatesh, "A Digital On-Demand Video Service Supporting Content-based Queries," *Proc. ACM Multimedia'93*, Anaheim, CA, pp. 427-433, August, 1993.

[15] J. Maeda, "Method for Extracting Camera Operations to Describe Sub-scenes in Video Sequences," *Storage and Retrieval for Images and Video Databases II, IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, February 1994.

[16] A. Nagasaka, and Y. Tanaka, "Automatic Video Indexing and Full-Video Search for Object Appearances," *Visual Databases Systems II*, E. Knuth and L. M. Wegner, eds., pp. 113-128, 1992.

[17] W. Niblack, R. Barber, W. Equitz, M. Flickner, D. Petkovic, and P. Yanker, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," *IS&T/SPIE Symposium on Electronic Imaging; Science & Technology*, San Jose, CA, February, 1993.

[18] K. Perez-Lopez, and A. Sood, "Comparison of Subband Features for Automatic Indexing of Scientific Image Databases," *Storage and Retrieval for Images and Video Databases II, IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, February 1994.

[19] L.A. Rowe, J.S. Boreczky, and C.A. Eads, "Indexes for User Access to Large Video Databases," *Storage and Retrieval for Images and Video Databases II, IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, February 1994.

[20] M.I. Sezan, and R.L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, Norwell, MA, p. 489, 1993.

[21] M.J. Swain, and D.M. Ballard, "Color Indexing," *Intl. Journal of Computer Vision*, Vol. 7, No. 1, pp. 11-32, 1991.

[22] J. Weng, T.S. Huang, and N. Ahuja, "Motion and Structure from Image Sequences," *Springer Series in Information Sciences 29*, Springer-Verlag, New York, p. 444, 1993.

[23] L.E. Wixcon, and D.H. Ballard, "Color Histograms for Real-Time Object Search," *Proc. SPIE Sensor Fusion II: Human and Machine Strategies Workshop*, Philadelphia, PA, 1989.

[24] H.J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-Motion Video," *Multimedia Systems*, Vol. 1, pp. 10-28, 1993.