# Script for Query Interface Demo

In our project, we adopt a new approach for handling video data. We view the video as a well-defined data type with its own description, parameters, and applicable methods. Our query model uses the features approach in accessing video by content. A video processing tool extracts MPEG7 compatible visual descriptors, and the semantic annotations of domain-experts.

We support different types of queries like query by example, query by motion information, query by keywords, query by an SQL statement, or batch execution of SQL statements. In query by example, the user selects an image to query with, the query features, and the number of results to display. The features are extracted from the query image online to construct an SQL statement that is sent to the server for execution.

The user can select the level of the query.

- At the shot level the user submits a query image to match against the aggregate features of shots in the database. In this query the color histogram feature is used to match against the aggregate color histogram of video shots in the database to get the 10 shots most similar to the query image. The SQL statement constructed from the query and displayed here shows the 32-dimensional color histogram descriptor extracted online from the query image. After executing the query, the representative frames of the matching shots are returned. Results are ordered according to their similarity to the query image. Users can navigate through the results and play any of them.
- At the frame level, the user submits a query image to match against frame features. Users can choose to have the system display the representative frames of the shots containing the matching frames. In this query the dominant color, homogeneous texture, and edge histogram features are used to match against the same features at the frame level to get the video shots containing the 5 frames most similar to the query image. Only one shot is displayed here since the 5 most similar frames all happen to be in the same shot. We can see that when we play the shot.
- For frame level queries, users can also choose to get the frames with features most similar to the features of the query image. In this query the homogeneous texture and texture edges features are used to find the 5 frames in the database that are most similar to the query image.

The SQL statement generated for each query is always displayed.

An SR-tree index structure is implemented as our high-dimensional access path to the extracted visual features. A new rank-join query operator supports queries based on multiple video features. This operator is used in the displayed SQL statement to query using the 32-dimensional homogeneous texture feature vector and the 9-dimensional texture edges feature vector.

User can also query using motion information. The user selects the type of motion and the number of contiguous frames with this motion type. In this query the user retrieves the parts of stored videos with more than 100 frames of zoom-in camera motion. The first frame for each video part is displayed.

The system includes search-based buffer management techniques to support continuous media streaming. A stream manager layer is implemented above the buffer manager to handle streams.

We also support query by Key Words. The semantic annotation is searched to get the logical shots associated with these key words. The representative frames of the logical shots are displayed and any of these shots can be played.

The SQL statement that is constructed online for each query is submitted to our server at Purdue University for execution. Results retrieved from the server are then displayed.