Proceedings, 33rd Annual Clinic on Library Applications of Data Processing:
Digital Image Access and Retrieval (Invited paper)

# Efficient Techniques for Feature-Based Image/Video Access and Manipulation

Shih-Fu Chang, John R. Smith, and Jianhao Meng

Department of Electrical Engineering &
Center for Telecommunications Research
Columbia University, New York, NY 10027, U.S.A.
{sfchang, jrsmith, jmeng}@ctr.columbia.edu
http://www.ctr.columbia.edu/{~sfchang, ~jrsmith, ~jmeng}

## 1. INTRODUCTION

Efficient, effective methods for indexing, searching, and retrieving images and videos from large archives are critical techniques required in visual information systems (VIS) applications. Users need a more powerful method for searching images than just traditional text-based query (*e.g.*, keywords). Manual creation of keywords is too time-consuming for many practical applications. Subjective descriptions based on users' input will be neither consistent nor complete. Also, the vocabulary used in describing visual contents is usually domain specific.

There is a recent effort, called content-based visual query (CBVQ) [1, 23], aiming at effective solutions to the above problem. Before we discuss the technical issues involved in the CBVQ, we first briefly discuss its relationships with other image indexing/retrieval approaches.

There are several possible ways of indexing and retrieving visual material. From users' point of view, the more methods available, the higher the flexibility they can exercise and adapt to the specific information they are seeking. However, from the system designer's point of view, different methods imply different cost and efficiency. It's important to achieve good overall system performance. The traditional keyword-based retrieval methods can be extended to more general semantic-level descriptions such as "a red car in front of a house" or "a person running on the beach". This type of retrieval requires semantic information given by users in the indexing stage. The second type of query allows users to specify a complete image or an image region as the query key. Specific images can be retrieved based on the similarity with the input image itself or the image features derived from the input image. This is usually called *query by image example*. The last type of query is *feature-based image retrieval*. Visual signal features are extracted in the indexing stage and compared in the search stage to find the "similar" images/videos. Typical features include texture, color, shape, object layout, motion, camera operations, face, logo, associated audio and speech, *etc*. Some features are for still images and others are for videos, although in general all still image features are applicable to video as well. The formulation of input features to the search engine can be provided by user's raw data (*e.g.*, drawing and sketch) or user's selection from system templates. The population of feature sets from each image or video in the database can be automatic or semi-automatic (*i.e.*, with user's assistance). Knowledge of application domain also helps significantly in developing reliable, automatic retrieval techniques.

In CBVQ, the term "content" refers to the structure and semantics of images and videos at various levels, ranging from pixel patterns, physical objects, spatial/temporal structures, to high-level semantics of the visual material. The content-based approach is not intended as a replacement for the keyword approach. Instead, it is considered as a complementary tool, particularly for applications which have large data collection and require fast search response. Provision of the content-based visual retrieval techniques also brings in new synergy between the text-based information and the visual information of the same material. Fusion of different information channels (text and visual in this case) has been used to achieve performance improvement in multimedia databases such as news archive [4].

A content-based visual query system requires several key components, including

- Choice of effective visual features and their integration with textual indexes.
- Visual feature extraction (automatic or semi-automatic) and object segmentation.
- Design of effective discrimination measures.
- Efficient indexing data structure for high-dimensional feature space.
- Efficient user interface for query specification and visual browsing.
- Association with domain knowledge and other datatypes.
- Exploration of functionalities in the compressed domain.
- Evaluation criterion and methodologies.

The above broad list reflects the broad cross-disciplinary nature of this area. However, to achieve the goal of searching images/videos based on visual content, a very challenging issue is visual content analysis and automatic extraction of “prominent” features. The definition of “prominence” may depend on the actual application requirements, but a general bottom-up approach is as follows. Start with a rich set of automatically extracted low-level, generic visual features, then derive high-level semantics by applying the domain knowledge provided by users or applications.

We focus on issues directly related to image/video processing and associated algorithms for efficient indexing, computation, and visualization for improving overall system performance. Several groups have reported promising work in this area [1, 13, 14, 17, 20, 21, 22, 24, 28]. We contrast our work with others by using fully automatic process for localized feature extraction (e.g., local color and texture regions), compressed-domain feature extraction without full decoding of compressed images/videos, and efficient user interfaces for users to specify localized features and their spatial layout.

In this paper we describe two example prototype systems of CBVQ, VisualSEEk and CVEPS. VisualSEEk is a fully automated content-based image query system which allows users to search images by localized colors and their spatial layout. It includes a Java-based Web interface for interactive visual content specification, integrated visual/textual search, and performance evaluation. It’s the first CBVQ system which supports local content specification and integrated query using visual feature and spatial layout. CVEPS (Compressed Video Editing and Parsing System) is a software prototype of video indexing and manipulation system which support automatic video segment decomposition, video indexing based on key frames or objects, and compressed video editing. It’s the first software-based system supporting both indexing and manipulation of compressed video. The compressed domain approach provide great benefits in reducing computational complexity, storage space requirement, and improving picture quality. Implementation of a simple example editing function, random cut and paste, can be improved in speed by about 100 times by using the compressed-domain approach.

## 2. VisualSEEk

VisualSEEk is a new content-based image query system that provides for querying by both image region visual properties and spatial layout. VisualSEEk is a hybrid system in that it integrates feature-based image representation with spatial query methods. The integration relies on a recently proposed representation of color regions by color sets [29]. Color sets provide for a convenient system of region extraction through back-projection. In addition, a decomposition of the quadratic form of the color set distance function consists of terms that are easily indexed. This allows for the efficient computation of color set distance and the indexing of color sets. As a result, unconstrained images are decomposed into near-symbolic images which lend to efficient spatial query.

## Content-based image query

A recurring problem in image database research is how to efficiently retrieve items from the database that optimize some function. In content-based image query applications this function typically approximates either the perceptual distance or similarity between the user's query image and target images in the database. In general, when the database is large and the image features have many dimensions the exhaustive search of the database is not computationally expedient. Furthermore, recent approaches towards content-based image query have neglected one important aspect of visual perception, spatial information and spatial relationships.

## Spatial image query

A significant aspect of discriminating among images depends on the spatial locations and relationships between objects or regions within the image. However, the problem of content-based image query is only exacerbated by introducing multiple image regions and spatial information into the query process. This is due to the combinatorial explosion resulting from comparison between multiples of regions or objects. On the other hand, by representing images symbolically, spatial query methods utilize the locations and spatial relationships of symbols. However, spatial queries do not consider the similarity of the "symbols" such as that based upon visual features of objects or regions.

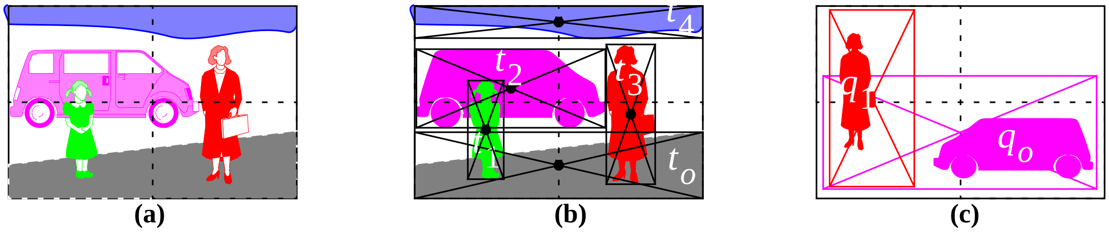

**FIGURE 1. Query decomposition (a) target image, (b) regions with both feature and spatial properties,** $T = \{\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_4\}$**, (c) query image,** $Q = \{\hat{a}_0, \hat{q}_1\}$**.**

## Joint content-based/spatial image query

In VisualSEEk, we propose a new system that provides both feature comparison and spatial query for unconstrained color images. To illustrate, see Figure 1, each image is decomposed into regions which have feature properties such as color, texture and shape, and spatial properties such as location and spatial relationships. The most desirable image query system allows users to query by both visual features and spatial properties. Recent approaches for image retrieval do not provide for both types of querying. The QBIC system [1] provides querying of manually segmented regions by color, texture and shape but not spatial relationships. The Virage system [30] allows querying of only image global features such as color, composition, texture and structure.

To solve the problem of integrating content-based and spatial image query we decompose the parameters of the image distance function into two classes: intrinsic and derived variables. Then we design the representations for intrinsic variables such as region color, spatial location and size to require minimal computation in matching. For example, color matching is achieved efficiently through color sets. Furthermore, the intrinsic variables are indexed directly to allow for maximal efficiency in queries. In this way, a query specified by the user is translated into pruning operations on intrinsic variables. The derived variables, such as region relative locations and special spatial relations are resolved only in the final stage of the query. This is because these evaluations have the highest complexity. The pruning performed by the queries on the intrinsic variables reduce the number of candidate images that need to be evaluated at the final stage.

### Unique Features of VisualSEEk

- joint image feature/spatial querying
- automated region extraction
- direct indexing of color features

The VisualSEEk project has emphasized several unique objectives in order to enhance the functionality and usability of image retrieval systems: (1) automated extraction of localized regions and features [31], (2) querying by both feature and spatial information [29], (3) extraction from compressed data [11], (4) development of techniques for fast indexing and retrieval and (5) development of highly functional user tools. The VisualSEEk client application was developed in the Java language to allow for maximum functionality, client platform independence and accessibility on the World Wide Web. As illustrated in Figure 2, VisualSEEk consists of several components: the set of user tools, the query server, the image and video retrieval server, the image and video archive, the meta-data database and the index files. Currently, the VisualSEEk system allows searching on a test-bed of 12,000 miscellaneous color images. The users can search for images using color and spatial attributes.

### Query formulation

The joint color/spatial queries are formulated graphically by using the VisualSEEk user tools as illustrated in Figure 3. The user sketches regions, positions them on the query grid and assigns them properties of color, size and absolute location. The user may also assign boundaries for location and size. The relationships between regions are diagrammed by connecting regions. In this way, the interface provides for queries that include region features and combinations of both absolute and relative placement of the regions.

### VisualSEEk queries

We present some example joint content-based/spatial queries in VisualSEEk. In the first example, see Figure 4(a), the query (top) specifies the absolute location of a single region. The retrieved image (bottom) has the best match in the features (color and size) to the query region and falls within the "zero-distance" bound diagrammed in the query. In the next example, see Figure 4(b), the query specifies multiple regions. The retrieved image provides the best match in terms of the features and absolute spatial locations of the query regions. In the next example, see

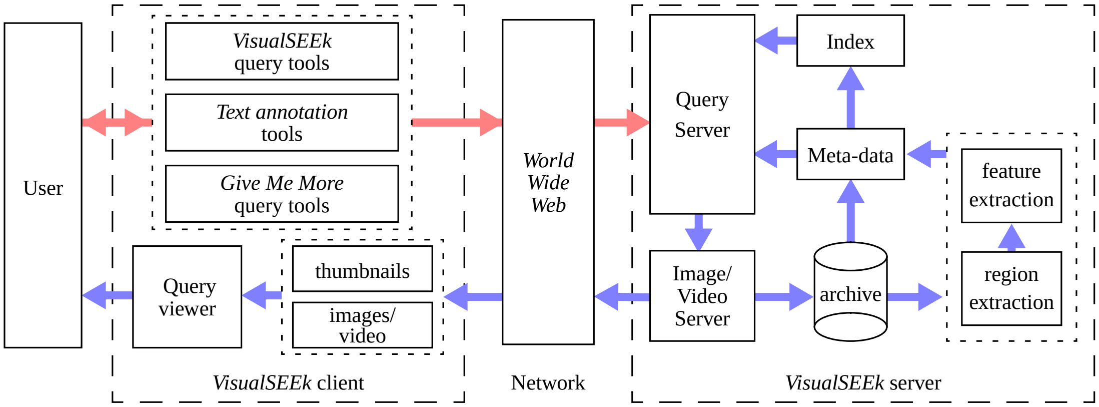

**FIGURE 2. VisualSEEk system architecture.**

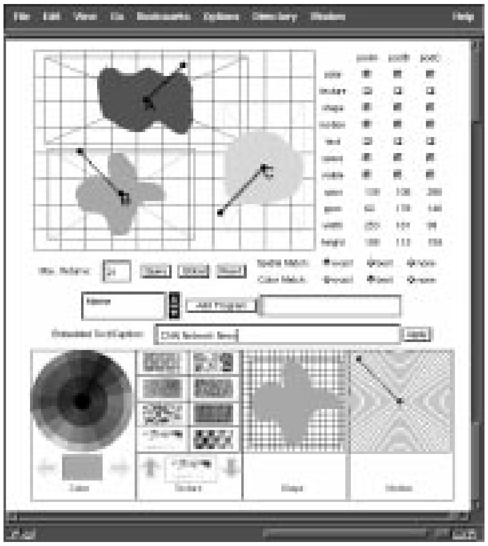

**FIGURE 3. VisualSEEk user interface provides tools for sketching query regions, assigning region properties and positioning regions to form joint content-based/spatial image queries.**

Figure 4(c), the query specifies the spatial relationships of regions. The retrieved image has three regions that best match the features of the query regions and their spatial relationship satisfies that specified in the query. Finally, see Figure 4(d), the query specifies both absolute and relative locations of regions. In this query, the match to the region positioned by absolute location (top left region in query image) considers both the features and location of this region. The match to the other regions (bottom two regions in query image) at first considers only the features of these regions. In the last stage of the query, the spatial relationship of the regions is evaluated to determine the match.

The discrimination of images is only partly provided by features such as color, texture and shape. Another important component is based upon the spatial locations and relationships of objects and regions within the images. The VisualSEEk system provides for the retrieval of images by image features and by the spatial locations, relationships and sizes of color regions. In VisualSEEk the image feature computation is efficiently integrated into the spatial querying methods. In future work will extend the VisualSEEk system to support other image features, such as texture and shape in the task of joint feature-based/spatial image query.

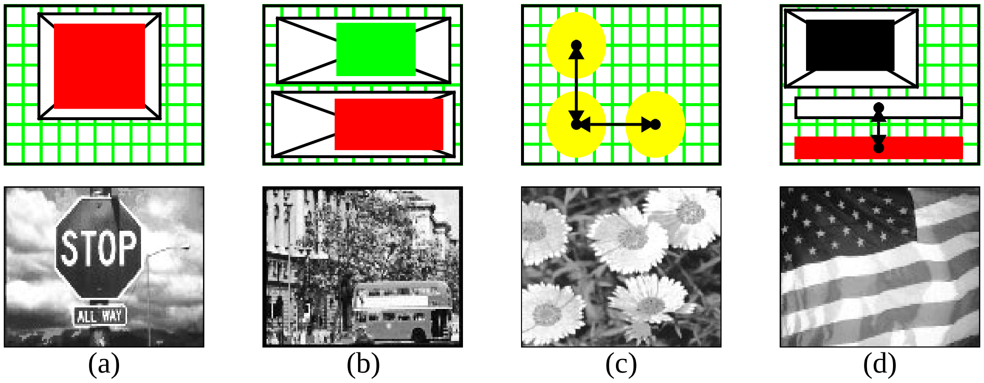

**FIGURE 4. Example VisualSEEk queries (a) single region with absolute location, (b) two regions with absolute locations, (c) multiple regions with relative locations, (d) multiple regions with both absolute and relative locations.**

## 3. CVEPS

CVEPS is a functional, software-based prototype system for compressed video editing, indexing, and browsing. It provides automatic tools for video segment (i.e., shot) decomposition, special effect detection (e.g., zoom, panning, dissolve), moving objects extraction, key frame/object indexing, and nonlinear editing, mostly in the compressed domain without full decoding of the original compressed streams. By using the compressed-domain approach, real-time performance is achieved even with software implementation.

The compressed-domain approach offers many great benefits. First, implementation of the same manipulation algorithms in the compressed domain will be much cheaper than that in the uncompressed domain because the data rate is highly reduced in the compressed domain (e.g., a typical 20:1 to 50:1 compression ratio for MPEG). Second, given most existing images and videos stored in the compressed form, the specific manipulation algorithms can be applied to the compressed streams without full decoding of the compressed images/videos. Lastly, because that full decoding and re-encoding of video are not necessary, we can avoid the extra quality degradation that usually occurs in the re-encoding process. We have shown earlier that for MPEG compressed video editing, the speed performance can be improved by more than 100 times and the video quality can be improved by about 3-4 dB if we use the compressed-domain approach rather than the traditional decode-edit-reencode approach [25].

The primary compression standard used in CVEPS is MPEG (MPEG1 or MPEG2). But the underlying approach and techniques are general enough to be applied to other video compression standards using transform coding and/or interframe motion compensation.

In order to allow users to manipulate compressed video efficiently, two types of functionalities are required (1) key content browsing and search, (2) compressed video editing. The former allows users to efficiently browse through or search for key content of the video without decoding and viewing the entire video stream. The key content refers to the key frames in video sequences, prominent video objects and their associated visual features (motion, shape, color, and trajectory), or special reconstructed video models for representing video content in a video scene. The second type of functionalities, video editing, allow users to manipulate the object of interest in the video stream without full decoding. One example is to cut and paste any arbitrary segments from existing video streams and produce a new video stream which conforms to the valid compression format. Other examples include special visual effects typically used in video production studios.

### System Components

CVEPS consists of three major modules: *Parsing, Visualization* and *Authoring*, as shown in Figure 5. In the Parsing module, MPEG compressed video is first broken into shot segments. Within each shot, camera operation parameters (e.g., zooming, panning) are estimated. Then moving objects are detected and their shape and trajectory features are extracted. In the Visualization module, the scene cut list and the camera zoom/pan information are used to extract key frames for representing each video shot. For example, a zoom-in video shot may be represented with the first frame of the shot and a detailed zoom-in frame, or a video mosaic. The key frames can be browsed with the hierarchical video scene browser based on scene clustering described in the next subsection [32]. Our content-based image query system, VisualSEEk, can be used to index and retrieve key frames or video objects based on their visual features and spatial layout. In the Authoring module, we provide tools for cutting/pasting of arbitrary MPEG video segments and adding special effects such as dissolve, keying, masking, and motion effects (variable speed, stroke motion etc.).

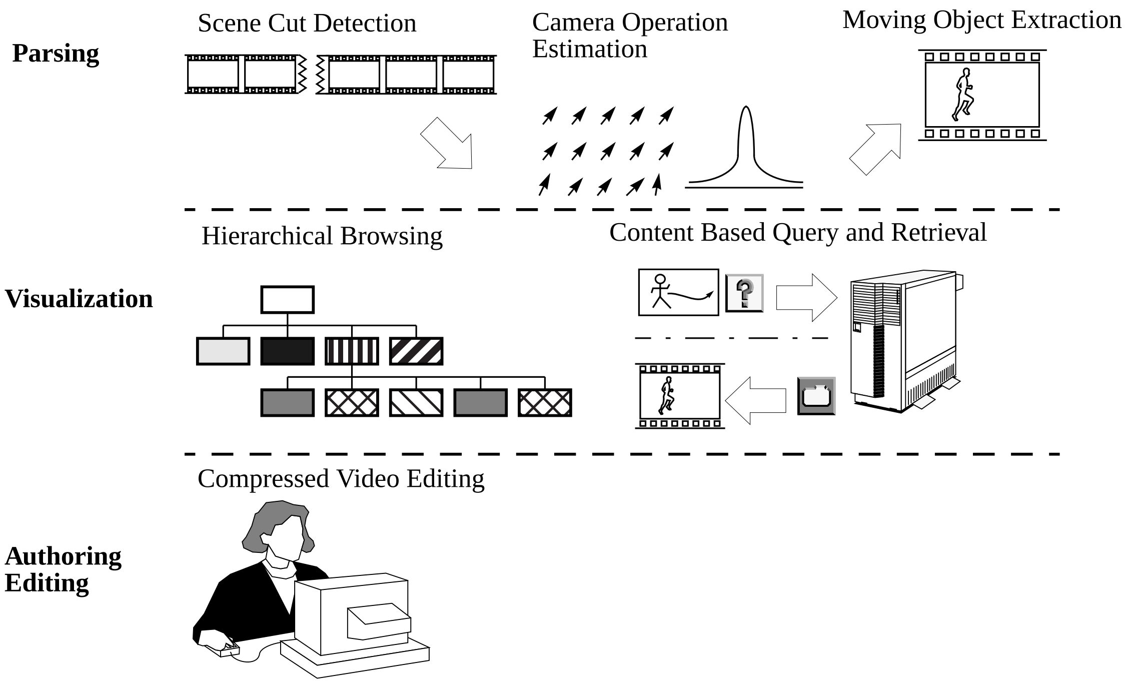

**FIGURE 5. CVEPS System Overview**

## Scene Clustering

We also develop scene clustering techniques to help users to efficiently browse and search important content in a video sequence. Each scene can be assumed to have consistent video content in most cases, except special situations like fast changing foreground, fast camera motion. Therefore, at the scene level, visual characteristics can be explored to classify various video scenes and derive higher-level semantics. Given a large collection of video sequences, a hierarchical scene browsing interface will be useful for users to quickly browse through the content contained in a long video sequence or multiple video sequences.

The hierarchical scene browsing system [32] takes the list of detected scenes from CVEPS and organize them to multiple levels according to three different criteria: *temporal order, story* (e.g., news story), and *visual features*. Organization based on the temporal order groups scenes into multi-level clusters according to their sequential order in time. N consecutive scenes form a basic segment in the lowest level, M segments form a class in the next level, and so on until all scenes are included. This hierarchy ensures a fully balanced organization with the same number of scenes in each class.

The second browsing mode uses the knowledge of story boundary and groups all video scenes contained in the same story into a class. This allows users to quickly view different stories in a long video program without going through the low-level scenes. This is particularly useful in situations where users need to quickly find a particular story from large video archive to meet short deadlines.

The third browsing mode organizes scenes based on their visual features. Users may want to find related scenes from different programs (e.g., related news of different days). Scenes with similar visual features (e.g., color, motion, and faces) may be assigned to the same class if their visual similarity is sufficient. One useful example is to find all anchorperson scenes automatically. This allows automatic detection of the news story boundary without user intervention. Currently, our system has implemented scene clustering using primitive features, color and motion. Advanced features will be incorporated later.

## Remote On-line Editing

The client-server model used in CVEPS provides for economic design of the client editing terminal. Users may use a light-weight viewing station to connect to a powerful CVEPS server. The emerging WWW technology also provides a very powerful platform for remote on-line editing. With a Java enabled Web browser, the user can first browse through the keyframes in a video stream; query and retrieve desired video clips; preview and edit video at a low resolution. Upon completion, the client software will generate a standard Edit Decision List (EDL). The EDL is sent to the server for generating a full resolution video. The CVEPS sever will handle most of the computing intensive tasks such as stream parsing and any required coding process. Once the actual manipulation and rendering of desired special effects is finished, users can request retrieval of the full-resolution video for stream download or real-time display. Note that multiple EDLs can be stored for the same video sequences before the final rendering. Final video output should be produced from as original video sources as possible to minimize multiple generation quality loss.

## CVEPS Prototype

We have developed a prototype of the CVEPS with C/Unix/Motif graphic user interface, see FIGURE 6. Every time an MPEG video is opened, its corresponding video buffer status is plotted against a timeline. This allows us to monitor and verify the bitrate of any video stream produced in CVEPS. The video bitrate needs to conform to some specification in MPEG and the parameters set in the encoded stream to avoid decoder abnormality (e.g., decoder buffer overflow or underflow).

The user may run the scene cut detection to extract a list of keyframes representing video shots. At any time, the user may invoke the MPEG software viewer and the interactive VCR control panel to do random search, step forward, fast forward/reverse, etc. The user can also use the mouse to highlight the time-line to select arbitrary video segments and apply “copy, cut and paste” operations. The CVEPS also provides options of inserting special effects, such as dissolve, fade in/out and wipe, between the connected segments.

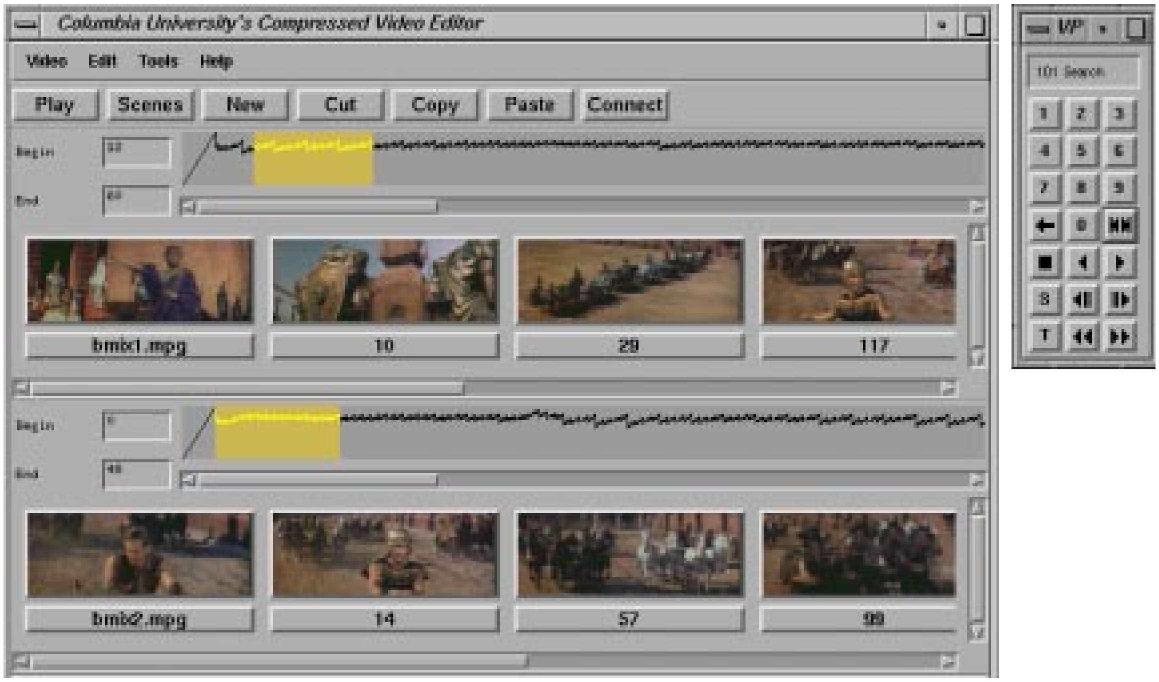

**FIGURE 6. The Compressed Video Editing and Parsing System User Interface.**

## 4. Conclusions

We discuss innovative, efficient techniques for image/video indexing, retrieval, and manipulation in large visual information management systems. We present two working systems, VisualSEEK and CVEPS, to illustrate our unique research approach in content-based visual query and manipulation.

VisualSEEk is a new content-based image query system that provides for querying by both image region visual properties and spatial layout. Unconstrained images are decomposed to local regions with prominent visual features (color, texture) by automatic tools. Spatial relationships are indexed with direct data structures to support efficient spatial queries.

CVEPS demonstrates advanced video indexing and manipulation functions in the compressed domain. The video parsing tools support automatic extraction of key visual features, e.g., scene cuts, transitional effects, camera operations (zoom/pan), shape and trajectories of prominent moving objects. These visual features are used for efficient video indexing, retrieval and browsing. The editing tools allow users to perform useful video compositing functions and special visual effects typically seen in video production studios. We contrast our compressed-domain approach with traditional decode-process-reencode approach with quantitative and/or qualitative performance comparison.

## 5. Acknowledgment

This work was supported in part by the National Science Foundation under a CAREER award (IRI-9501266), IBM under a Research Partnership (Faculty Development) Award, and sponsors of the ADVENT project of Columbia University. The scene clustering and hierarchical video browsing system includes joint work with Mr. Di Zhong and Dr. HorngJiang Zhang.

## 6. REFERENCES

1. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, “The QBIC Project: Querying Images by Content Using Color, Texture and Shape”, SPIE 1993 Intl. Symposium on Electronic Imaging: Science and Technology, Conf. 1908, Storage and Retrieval for Image and Video Databases, Feb. 1993. (also in IBM Research Report RJ 9203 (81511), Feb. 1, 1993, Computer Science.)
2. S.-F. Chang and D.G. Messerschmitt, “Manipulation and Compositing of MC-DCT Compressed Video,” *IEEE Journal of Selected Areas in Communications*, Special Issue on Intelligent Signal Processing, pp. 1-11, Jan. 1995.
3. S.-F. Chang and J. R. Smith, “Extracting Multi-Dimensional Signal Features for Content-Based Visual Query,” *SPIE Visual Communications and Image Processing*, Taipei, May 1995. (Best Paper Award)
4. R. K. Srihari, “Automatic Indexing and Content-Based Retrieval of Captioned Images,” *IEEE Computer Magazine*, Vol. 28, No. 9, pp. 49-56, Sept. 1995.
5. S.-F. Chang, A. Eleftheriadis, and D. Anastassiou, “Development of Columbia’s Video on Demand Testbed,” *Journal of Image Communication*, Special Issue on Video on Demand and Interactive TV, 1995, in press.
6. J. Meng, Y. Juan and S.-F. Chang, “Scene Change Detection in a MPEG Compressed Video Sequence,” *SPIE Symposium on Electronic Imaging— Digital Video Compression: Algorithms and Technologies*, San Jose, Feb. 1995.
7. P. Brodatz, Textures: a Photographic Album for Artists and Designers, Dover, New York, 1965.
8. J.R. Smith and S.-F. Chang, “Quad-Tree Segmentation for Texture-Based Image Query” *Proceedings, ACM 2nd Multimedia Conference*, San Francisco, Oct. 1994.

9. J. Smith and S.-F. Chang, “Automated Image Retrieval Using Color and Texture,” submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Special Issue on Digital Libraries - Representation and Retrieval, Nov. 1996. (Also CU-CTR Technical Report # 414-95-20.)

10.H. Wang and S.-F. Chang, “Adaptive Image Matching in the Subband Domain,” submitted to *SPIE/IEEE Visual Communications and Image Processing ‘96*, Orlando, Florida, March, 1996. (also CU/CTR Technical Report #422-95-28).

11.S.-F. Chang, *Compositing and Manipulation of Video Signals for Multimedia Network Video Services*, Ph.D. Dissertation, U.C. Berkeley, Aug., 1993.

12.R.W. Picard, “Light-years from Lena: Video and Image Libraries of the Future,” *IEEE International Conference on Image Processing*, Washington DC, Oct. 1995.

13.A. Pentland, R. Picard, and S. Sclaroff, “Photobook: Tools for Content-Based Manipulation of Image Databases,” *Int’l Journal of Computer Vision*, 1995, in press.

14.H. Zhang A. Kankanhalli, S.W. Smoliar,” Automatic Parsing of Full-Motion Video,” *ACM-Springer Multimedia Systems*, 1(1), pp. 10-28, 1993.

15.F. Arman, A. Hsu, and M.-Y. Chiu, “Image Processing on Compressed Data for Large Video Databases,” *Proceedings of ACM Multimedia Conference*, June 1993.

16.Y.Y. Lee and J. Woods, “Video Post Production with Compressed Images,“ *SMPTE. J. Vol. 103*, pp. 76-84, Feb. 1994.

17.N. Dimitrova and F. Golshani, “Rx for Semantic Video Database Retrieval,” *ACM Multimedia Conference*, 1994, Oct., San Francisco.

18.B.C. Smith and L. Rowe, “A New Family of Algorithms for Manipulating Compressed Images,” *IEEE Computer Graphics and Applications*, pp. 34-42, Sept., 1993.

19.S.-F. Chang, “New Algorithms for Processing Images in the Transform-Compressed Domain, “ *SPIE Symposium on Visual Communications and Image Processing*, Taipei, May, 1995.

20.B.-L. Yeo and B. Liu, “A Unified Approach to Temporal Segmentation of Motion JPEG and MPEG Compressed Video,” *IEEE Int. Conf. on Multimedia Computing and Systems*, May 1995.

21.A. Hampapur, R. Jain, and T.E. Weymouth, “Production Model Based Digital Video Segmentation,” *Journal of Multimedia Tools and Applications,* Kluwer Academic Publishers, Vol. 1, No. 1, March, 1995.

22.R. Mehrotra and J. E. Gary, “Similar-Shape Retrieval in Shape Data Management,” *IEEE Computer Magazine*, Vol. 28, No. 9, pp. 57-62, Sept. 1995.

23.R. Jain, NSF workshop on Visual Information Management Systems, Redwood, CA, Feb. 1992.

24.H.S. Stone and C.-S. Li, “Image Matching by Means of Intensity and Texture Matching in the Fourier Domain,” to appear in *Proceedings of SPIE Conference on Image and Video Databases*, San Jose, CA, Jan. 1996.

25.J. Meng and S.-F. Chang, “Tools for Compressed-Domain Video Indexing and Editing,” *SPIE Conference on Storage and Retrieval for Image and Video Database,* San Jose, Feb. 1996.

26.J. Meng and S.-F. Chang, “Buffer Control Techniques for Compressed Video Editing,” Submitted to *IEEE Int’l Conference on Circuits and Systems*, ISCAS ‘96.

27.P. P Vaidyananthan, “Orthonormal and Biorthonormal Filter Banks as Convolvors, and Convolution Coding Gain,” *IEEE Transactions on Signal Processing*, Vol. 41, June, 1993.

28.H.S. Sawhney, S. Ayer, and M. Gorkani, “Model-Based 2D and 3D Dominant Motion Estimation for Mosaicking and Video Representation,” *Proc. Fifth Int’l Conf. Computer Vision, 1995.*

29.J. R. Smith and S.-F. Chang, “Tools and Techniques for Color Image Retrieval,” In *Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV*, volume 2670, San Jose, CA, February 1996.

30.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,” In *Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV,* volume 2670, San Jose, CA, February 1996.

31.J. R. Smith and S.-F. Chang, “Querying by color regions using the VisualSEEk content-based visual query system,” In *Intelligent Multimedia Information Retrieval*. IJCAI, 1996.

32.D. Zhong, H.J. Zhang, and S.-F. Chang, “Clustering Methods for Video Browsing and Annotation,” *Storage and Retrieval for Still Image and Video Databases IV*, *IS&T/SPIE’s Electronic Imaging: Science & Technology 96*, Vol. 2670, San Jose, CA., Feb. 1996.