

COMPRESSED-DOMAIN CONTENT-BASED IMAGE AND VIDEO RETRIEVAL

Shih-Fu Chang

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

INTRODUCTION

With more and more visual material produced and stored in visual information systems (VIS) (*i.e.*, image databases or video servers), the need for efficient, effective methods for indexing, searching, and retrieving images and videos from large collections has become critical. Users of large VIS will desire a more powerful method for searching images than just traditional text-based query (*e.g.*, keywords). Manual creation of keywords for a huge collection of visual materials 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.

To overcome these shortcomings, there is a recent approach called "content-based" visual query, which allows users to specify query and execute search based on the visual content of the visual material [1, 23]. The term "content" refers to the semantic structure of images and videos at various levels, ranging from pixel patterns, physical objects, to spatial/temporal structures 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 visual feature extraction, feature indexing data structure, distance (or similarity) measurement, fast search methods, integration of different visual features, and integration of visual features with text-based indices. We focus on issues directly related to image/video processing. Our goal is to investigate automated methods using various useful visual features for content-based visual query. This is a relatively new area, in which some promising work has been reported in the literature for different applications. The QBIC system of IBM [1] provides semi-automatic mechanisms for extracting color, texture, shape, and structure for image query. Pentland *et al* [13] have demonstrated image query systems based on shape, texture, and face. Video indexing based on temporal characteristics has been described in [14, 17, 20, 21, 28]. Other researchers [24, 22] have also proposed techniques for image search based on shape, color, or texture.

In this paper, we focus on automated image retrieval using various types of visual features (including image templates). Specifically, we explore a unique approach which emphasizes the compressed-domain solutions. Given most images and videos stored in the compressed form, it is highly desirable to perform content-based visual retrieval directly in the compressed domain. In other words, extraction of visual features, matching of images and videos, or even manipulations of search results can be implemented using the compressed visual data without decompressing (or sometimes with minimal decoding). The advantage of performing these tasks in the compressed domain is obvious. The computational cost can be greatly reduced compared to the alternative which fully decodes the compressed material and executes the above tasks in the uncompressed domain. Not only that the decoding process is avoided, the amount of data in the compressed domain is also lower than in the uncompressed domain. We have applied the compressed-domain approach to image/video manipulation and

<u>Keywords</u> “King Tut”, “Egypt”	<u>Feature Specification Tools</u> <i>texture, color, shape, object layout, motion, camera operations, temporal structure, face, local, audio, speech</i>
<u>Semantic Structure</u> “A red car in front of a house”, “A person running on the beach”	<u>Query by Image Example</u> <i>feature-based image template matching</i>
—> (user assistance needed)	—> (automatic process possible)

FIGURE 1. Different Modalities of Image/Video Search and Retrieval

compositing [2,11], texture/color-based image query [9], image matching [10], video indexing and editing [6, 25].

We first describe different modalities for image/video retrieval in a VIS. We then present an overview of our current work in compressed-domain image/video retrieval, followed by a brief description of an integrated VIS prototype system with Video-on-Demand applications [5]. Finally, we discuss conclusions and future work.

DIFFERENT MODALITIES OF IMAGE SEARCH AND RETRIEVAL

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. Figure 1 shows several possible methods for image query. 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 shown in the figure 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 videos 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 this paper, we focus on the automatic techniques in the compressed domain, with necessary domain knowledge whenever available.

TEXTURE AND COLOR FEATURE EXTRACTION IN THE SPATIAL-FREQUENCY TRANSFORM DOMAIN

Among various image features, texture, color, and shape are the most typical features used in characterizing the image content for image retrieval. Texture is useful for discriminating different natural objects, such as trees, walls, and textiles. Color has been used as a powerful technique in discriminating colored objects and different video scenes. Shape gives the most direct geometric information about the object of interest. However, it is also more difficult to extract under unconstrained conditions. Spatial and temporal relations of these features can provide higher-level information about the images and usually can help achieve better performance.

We have made progress in handling the first two features, texture and color. Psychological studies have suggested that human eyes recognize different textures based on their frequency and orientation properties. Based on this assumption, spatial-frequency methods have been used to perform texture discrimination and seg-

mentation. Extending this assumption, we define the texture features from several popular transforms used in image compression. Examples include the Discrete Cosine Transform (DCT) and the Subband Transform (wavelet, uniform, and tree structure). Texture features are defined as the energy distributions over different subbands. For DCT, different subbands are formed by re-grouping transform coefficients using a polyphase transform. The above method of feature extraction is considered as a compressed-domain solution, since many practical systems use these transforms as the underlying compression technique. Given images represented by the transform coefficients, texture features can be directly computed without decoding of compressed images.

The above texture feature can be applied to texture classification and segmentation. For classification, we tested the entire Brodatz texture set of 112 texture classes [7]. Using the Fisher Discriminant Analysis technique, we derived a linear transform which transforms the above feature vectors to a new space in which the average distance between different texture classes is maximized and the average distance within each texture class is minimized. The Mahalanobis distance measure in the transformed feature space was used to compute the dissimilarity between any two images. This texture feature has been tested on a texture database containing more than 2000 random cuts from the Brodatz set. It showed satisfactory accuracy at about 89% - 93%, depending on the specific transform used.

In texture segmentation, we adopted a modified quad-tree data structure to assist the segmentation process. Thresholds for merging and splitting neighboring nodes in the quad-tree were established based on heuristics optimized in the training process using the entire Brodatz texture set. Using this quad-tree based texture segmentation method, we extracted prominent texture regions from a general image. Note that we emphasize the term “prominent” here because our goal is not to perform accurate object segmentation in unconstrained images (which is still considered hard). Instead, our primary interest is finding significant regions which have consistent visual properties such as texture.

In another effort to improve the indexing and search efficiency, we further simplified the process of extracting texture features. We still focused on the energy distributions in the transform domain, particularly the wavelet subband domain. We took advantage of the spatial resolution provided by the wavelet subband decomposition and defined a pixel-based texture signature, which is a binary vector indicating the energy level of each pixel in each subband. Neighboring pixels with the same texture signatures were grouped to the same texture region, using some non-linear smoothing operations. Each texture region was indexed by a single texture signature, which then can be used in the searching stage. Note that since the texture feature was defined by a binary vector, distance measurement between different textures can be easily computed (*e.g.*, the hamming distance).

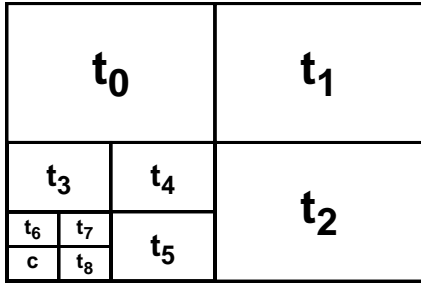
Figure 2 shows a diagram illustrating the above texture extraction algorithm in the subband domain. It also shows the concept of extracting color features from the approximation subband (*i.e.*, the lowest subband). We define the color feature in a way so that prominent image regions with a consistent “single” color or a consistent “color mixture” can be extracted. By quantizing a carefully chosen color space, we first reduce the color resolution of the images. For each quantized color, we calculate the distance map, apply the non-linear smoothing operation, and extract prominent image regions with this color. Prominent image regions with multiple colors (*e.g.*, mixture of red, white, and blue in the U.S. flag) can also be extracted using extension of this method. Each resulting color region is indexed by a binary feature vector, indicating the presence of individual colors in this region.

With the combination of multiple features (*e.g.*, texture and color in this case), users can search for desired images with greater flexibility. The relationship between the color regions and texture regions also provides useful information for image retrieval. In our feature definitions, both color and texture use binary feature vectors, which then can be concatenated easily for joint descriptions.

QUERY BY IMAGE MATCHING

As mentioned earlier, one way of image query is for users to provide an input image or image region, which is then matched against all image candidates in the database. The matching procedure can be based on derived features, as described above, or directly on the raw image data. Some papers categorized the former as “feature-descriptor based”, the latter simply as “image matching”. We focus on image matching in this section.

Image matching provides a brute-force method for finding specific images from the database. Compared to feature-based image retrieval, it does not require the operations for feature extraction. Also, if consistent,



(a)



(b)

FIGURE 2. Extracting texture and color in the wavelet subband domain (a) color from the lowest subband and texture from other subbands (b) an example illustrating the extracted texture regions.

prominent features are difficult to extract, image matching can be considered as a valuable alternative. However, image matching does not provide straightforward correspondence to subjective perception. If the input image is not generated from the existing images in the database, it may be hard to evaluate the relevance of the returned images simply based on the relative ranking of the matching scores.

Image matching can be implemented by several possible methods, including correlation, MSE (mean square error), and normalized correlation (*e.g.*, correlation coefficient). Applying the compressed-domain approach, we have investigated a fast algorithm for image correlation directly in the subband domain. It can be shown that other matching criteria, such as MSE and normalized correlation, can be computed based on the correlation operation. Given the subband components of two images, the complete image correlation can be expressed as the summation of the filtered correlations of different subbands, including intra-band and cross-band correlations. For ideal lowpass and highpass filter banks, the cross-band correlations are cancelled. In a related research, Vaidyanathan [27] has used the subband convolver (time reversed version of correlation) to improve the coding gain and accuracy of convolution.

To take advantage of energy compaction of subband decomposition, we further proposed an adaptive image matching method to discard correlation components with insignificant energy levels. By choosing energy-significant subbands, we can approximate the original correlation with minimal loss of accuracy. Also, the computation cost in the subband domain is much less than that in the pixel domain (more than 10 times faster), since many insignificant cross-band correlation components can be discarded.

Our experiment in satellite image database search showed that the normalized correlation provided the most satisfactory search results, compared to MSE and correlation. One obvious advantage of using normalized correlation was illumination invariance. However, it requires a little bit higher implementation cost than others. We also evaluated different filter banks based on their energy compaction capability and the resulting accuracy in terms of adaptive approximation of image correlation.

ADDING THE TEMPORAL DIMENSION — VIDEO SEARCH

Video provides a more dynamic dimension in presenting visual content — time. Image objects, their associated attributes, and their spatial relationship may change as time proceeds. There have been two different approaches in deciphering the information carried by the video signal. One approach is to identify and recognize different channels of information and then index their continuous temporal flow structure in the entire video sequence. For example, people, background, weather, and actions can each be an individual information channel and indexed separately. Different channels of information do not need to occupy the same temporal intervals (*e.g.*, scene cuts). Because of the rich semantic information involved, this approach needs intensive input from users. The other approach is based on the hierarchical temporal structure of videos. Each video sequence can be segmented to different scenes and each scene may have multiple shots. Shots of different scenes may be arbi-



FIGURE 3. Graphic User Interface of Columbia's Video Indexing and Editing System. The VCR control panel is displayed on the right side.

trarily interleaved on the time line based on the preference of the director. If we assume the visual content within each shot evolves in a “trackable” way, we may index each shot with few key frames or an extended view (*e.g.*, image mosaics). One benefit about the scene-based indexing is that automatic detection of scene change and extraction of key frames are feasible. (Note that most literatures used the terms “scene” and “shot” interchangeably, though they differ in film production.) Once the scene unit and key frames are extracted, various retrieval techniques for still images can be applied to the key frames or extended representation of the scenes.

We investigated techniques for scene change detection in the compressed domain (*e.g.*, MPEG). A complete scene change detection should also handle camera operations (*e.g.*, pan, zoom), special editing effects (*e.g.*, dissolve, fade in/out). For the general scene analysis task, the MPEG compression standard provides several clues, including the motion vector distribution (temporal direction, spatial direction, and magnitude) and the transform coefficients. By looking at the motion vector distribution of the B and P frames, our techniques can detect abrupt scene changes at any arbitrary frame (including I frames). For the dissolve special effect, we assumed a quadratic evolution model of the image intensity variance. Using the DC transform coefficients to approximate the image intensity variance, we can detect dissolve in the compressed domain as well. Ideal panning and zooming produce special patterns (*e.g.*, parallel or concentric motion vectors) in the extracted motion field, which can be used to detect such camera operations [25]. Motion features can sometimes be used to segment simple foreground objects and background objects as well. Once the foreground and background objects can be separated, they can be stored and manipulated separately. This has very interesting applications in video content production, such as video games.

Figure 3 shows the user interface of the current prototype of our video indexing and browsing system. Software parsing and decoding of the compressed stream has been developed. Interactive controls (*e.g.*, VCR functions) were also provided. From the individual detected scenes, users may further manipulate (*e.g.*, cut and paste) these scene units to create new visual content directly in the compressed domain, as discussed in the next section.

MANIPULATION OF SEARCH RESULTS IN THE COMPRESSED-DOMAIN

Indexing and editing are two functions that come hand-in-hand in many practical applications. Users of visual information systems need editing functions to formulate the query keys or, more importantly, to manipulate the query results for final presentation or production. Our journey in the compressed-domain world does not stop with the indexing part. Continuing our prior work on compressed-domain image/video compositing, we have developed techniques for editing compressed video in the compressed domain. One example from our prior work is compositing multiple video windows into a single displayable video window. This function was originally intended for multi-point video conferencing, though it can actually be applied to other general applications. A video bridge in the network will take multiple compressed videos, perform size scaling and spatial translation, and then overlay them to the same displayable video window. The final composited video is then distributed to all participants. Our work shows that the compressed-domain approach provides great potential for reducing the

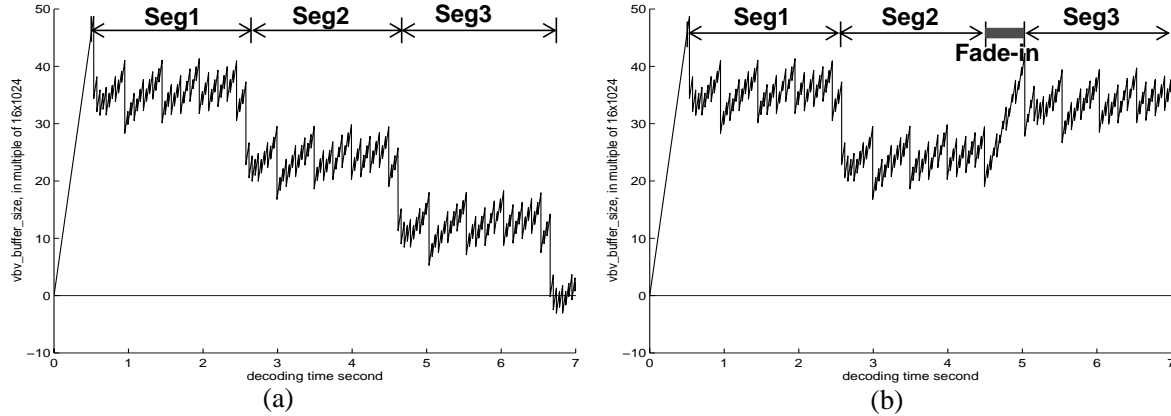


FIGURE 4. (a) A hypothetical example illustrating the buffer overflow problem in cutting and pasting compressed video. (b) The overflow problem is solved by inserting a synthetic fade-in interval.

computational cost, particularly if the video compression ratio is high and the manipulation operations are simple. Complicated operations (*e.g.*, rotation and shearing) also have compressed-domain equivalents [19]. But their compressed-domain implementations may not pay off due to the large overhead involved.

One simple but useful video editing operation is cut and paste. After retrieving several video segments from the database, users may want to “splice” some interesting segments from different sources and create a new video sequence, hopefully in the compressed domain. Unfortunately, there are some constraints imposed by the practical design of the video decoders. In order to have non-interrupted playback, the hardware decoders need to maintain a certain level of fullness in the buffer all the time. This constraint is essential whenever the input and output of the buffer have different rate patterns (constant bit rate vs. variable bit rate). In this case, the decoder buffer has CBR in and VBR out. Usually, the buffer constraint is enforced in the encoder by some rate control mechanisms (*e.g.*, VBV control in MPEG). Each encoded video stream maintains its rate integrity through this rate control mechanism. This integrity will be broken if we arbitrarily cut and paste random video segments from different streams.

Figure 4 shows a hypothetical case in which a new stream created by multiple cuts and pastes causes the decoder underflow problem due to the large picture size at the beginning of connected video segments. These large pictures are possible particularly when users cut at the scene change points. There are cases in which buffer overflow may occur as well. Examples are scenes starting with dark lighting (*e.g.*, fade in). In order to maintain valid compressed streams without causing buffer violations in the decoder, we have investigated several solutions. Possible solutions for buffer underflow include simple frame repetition at the decoder (so that the buffer has time to fill up) or insertion of a synthetic special effect (*e.g.*, fade-in) between two video segments. The former relies on the specific design of the decoder while the latter can completely remove the buffer violation and thus relieve the responsibility from the decoder. The synthetic fade-in period produces low bit rate and therefore it can help bring the decoder buffer back to the normal status. The fade-in effect minimizes the visual discontinuity of the displayed video at the cost of a minor increase of delay.

Besides the above buffer issue, boundary frames of the cut segments need to be modified so that the new stream will conform to the required format (*e.g.*, the GOP and frame structure in MPEG). But this process involves only few frames and requires relatively little computations. We have developed fast methods to handle required format modification as well [26].

AN INTEGRATED VIS/VOD TESTBED

We have developed a Video on Demand (VOD) testbed [5] in which the visual information system plays a critical component. The VOD system is equipped with real-time parallel graphic processors and high-speed ATM network resources. Real-time video pump, network adaptation, high-level application controls (DSMCC), and real-time client decoder (MPEG-1 and MPEG-2) have been developed. Envisioned applications include education, information, and instruction on demand on the campus. The same VOD testbed is being used for interoperability experiments.

Video indexing and editing tools described above are being incorporated into the VOD testbed. Key frames and/or classified scene categories are used as the basis for video browsing. From the video browsing interface, users can select any arbitrary segments from different sources and “paste” them to form a new compressed video stream, using the compressed-domain technique. The compressed-domain approach also allows us to develop fast software implementation.

CONCLUSIONS AND FUTURE WORK

We presented an overview for the compressed-domain content-based visual retrieval technologies. For still images, we have developed techniques for texture and color feature extraction in the transform domain (particularly the wavelet subband domain). We have also developed adaptive image matching technique in the subband domain. A speedup greater than 10 times can be achieved using the compressed-domain image matching. For videos, we used the compressed data (such as motion vectors and transform coefficients) to detect the low-level visual features, such as scene change, camera operation, and special effects. We have also developed preliminary techniques for editing compressed video streams in the compressed domain. A fast software-based compressed video editor has been implemented. Most of the visual indexing work described in this paper is being incorporated to a Video-on-Demand testbed in the Image and Advanced TV Lab at Columbia.

One future goal is to investigate fusion of multiple visual features and their integration with domain knowledge in specific applications. All techniques described above can be considered as different tools which should be fine-tuned and combined in an optimal way to satisfy the specific requirement of each application. Another direction of future work is to add content accessibility to new image/video compression techniques. Many constraints are imposed by today’s compression standards, resulting in limited content access functionality. The task is not easy. But with many exciting applications driving the need, it has captured great interest from researchers in various disciplines, including the new video compression standardization effort, MPEG-4.

ACKNOWLEDGEMENT

The author is thankful for Prof. David Messerschmitt at UC Berkeley for his invaluable guidance and early stimulation of the author’s interest in this field. This paper includes joint work with several people, including John Smith, Hualu Wang, and Jianhao Meng. This research is supported in part by the National Science Foundation (under a CAREER award, IRI-9501266), IBM (under a 1995 UPP award), NEC, and Columbia’s ADVENT project.

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," to appear in *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. Vaidyanathan, "Orthonormal and Biorthonormal Filter Banks as Convolvers, 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.