Published at IEEE International Conference on Image Processing (ICIP95),
Special Session on Digital Library and Video on Demand, Washington DC, Oct. 1995.

Compressed-Domain Techniques for Image/Video Indexing and Manipulation

Shih-Fu Chang

Department of Electrical Engineering and Center for Telecommunications Research
Columbia University, New York, NY 10027

Abstract

As massive amount of visual materials are captured and stored in visual information systems, effective and efficient image indexing and manipulation techniques are required. Most visual materials in visual information systems are stored in some compressed forms. Therefore, it is desirable to explore image technologies for feature extraction and image manipulation in the compressed domain. In other words, image feature extraction and manipulation are performed on compressed images/video without decoding, or with minimal decoding only. Although the compressed-domain approach imposes many constraints, it provides great potential for reducing computational complexity, because of reduction of the amount of data after compression. This paper provides an overview of our research in this area. Specifically, it describes the results and the future directions of our work on compressed-domain texture feature extraction, image matching, image manipulation, and video indexing.

1. Introduction

Effective techniques for *image indexing/searching* are required for large visual information systems (such as image databases and video servers). In addition to traditional methods that allow users to search images based on keywords, image query by example and feature-based image search provide powerful tools to complement existing keyword-based search techniques. Usually, prominent image features (such as texture, shape, color, and object motion) are extracted and stored as side information. Then, similarity retrieval is performed, based on the comparison of the features associated with each image in the database.

Another important image technology for general multimedia systems and applications is *image manipulation*. On a desktop video editing system, users would like to have general tools for image geometrical transformation, image filtering, multi-image composition, and video segment cut-and-paste. In a networked video application, users may want to subscribe multiple image/video sources from different locations and combine them to a single displayable format. In a multi-point video conferencing application, a network device such as a video bridge may receive multiple video sources and generate multiple video streams of various forms to different end users.

As mentioned above, there is a general need for efficient image searching and manipulation techniques for multimedia applications. However, these techniques are usually pursued independently of the design of image compression algorithms. Most of today's image compression methods are concerned mainly with optimization of signal distortion, bit rate, and coding complexity. There is very little emphasis on visual content accessibility, which is similar to the fourth criterion mentioned in [12]. We believe there is great synergy among image compression, manipulation, and feature extraction. Joint study of these issues should be pursued in order to improve the overall system performance. Figure 1 shows the concept of incorporating image manipulation and feature extraction flexibility into the compression arena.

A timely and important research issue would be the following: given today's existing compression techniques, such as transform coding and interframe predictive coding, what are the functionalities that one can possibly achieve in the existing compressed domain? Pursuing the maximal functionalities in the compressed domain has several advantages. First, there is great potential for reducing overall computational cost, since there is less data in the compressed domain than the original uncompressed domain. Second, most stored visual materials are compressed. Applying image searching and manipulation techniques in the compressed domain can avoid the overhead of decoding and re-encoding of existing compressed materials. In addition, many compression algorithms actually perform some forms of information filtering (such as motion estimation) and content decomposition (such as frequency decomposition), which can provide good foundations for subsequent image content analysis.

This paper gives an overview of our research on compressed-domain techniques for image/video indexing and manipulation. Specifically, we will describe examples of visual feature extraction, image matching, image manipulation, and video editing in the compressed domain. It should be noted that an ideal outcome is to have techniques that operate on the compressed data directly without any decoding. However, in some cases, some extent of image decoding may be necessary, in order to extract useful data

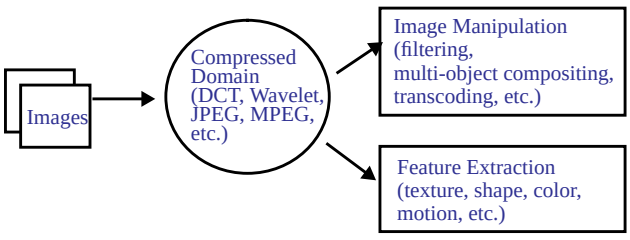

FIGURE 1. Concept of Compressed-Domain Image Manipulation and Feature Extraction.

1. Correspondence to *sfchang@ctr.columbia.edu, www: http://www.ctr.columbia.edu/~sfchang.*

from the compressed images. The "compressed domain" in this paper refers to both the ideal case without decoding and the sub-optimal case with minimal decoding.

Our previous work has investigated general image manipulation and feature extraction techniques in the transform domain (e.g., DCT and subband) and the MPEG domain [1, 2, 5, 6, 9]. Independent work on DCT-domain image manipulation techniques was reported in [17, 11]. Algorithms for video scene change detection based on the DCT transform and/or motion vector distribution were described in [13, 14]. Techniques for achieving image special effects (such as picture in picture) were described in [15]. Dynamic rate shaping of video streams in the compressed domain has been studied in [10] as well.

2. Compressed-Domain Image/Video Indexing and Searching

2.1 Texture Discrimination & Search

In an effort to provide feature-based image query, we have derived automatic algorithms for extracting low-level signal features from the transform compressed images [6]. One specific example is to define the texture features based on the spatio-frequency decomposition of the images. Textures has been used to describe content of many real-world images; for example, clouds, trees, bricks, hair, fabric all have textural characteristics. Psychophysical studies have shown that humans perceive textures by decomposing signals into components with different frequency and orientation. We use the feature sets defined in transform decomposition to approximate the texture feature. Transform decomposition of images can be obtained by taking DCT, subband transform, or wavelet transform of the images. From the decomposed signal bands, texture feature sets are defined by measuring each subband energy. For example, for a 5-level wavelet decomposition, feature vectors with 16 terms are produced. For a $N \times N$ DCT transform, N^2 signal bands can be obtained by regrouping transform coefficient (i.e., the DCT/Mandala transform). Other statistical measures such as first-order moments can also be derived from each subband in forming the transform-domain texture features. Based on our experiment of texture classification, the energy measurement seems to be the most effective one.

In order to reduce the search complexity, the above texture feature vector is further reduced by using the Fisher Discriminant technique. The criterion is to maximize the class separability among all different known texture classes in the chosen testset (i.e., the Brodatz Texture Set) [4]. The testset contains 112 different texture classes typically used in computer vision research. Based on this 112 texture classes, we generated an image database containing more than 2000 random cuts from the Brodatz set. Given an input image key, the transform-domain feature elements are mapped to a set of eigenvectors with the maximum separability significance. The Mahalanobis distance in the transformed feature space was used to measure the similarity between the input image key and every image in the database. Our experiment shows satisfactory correct classification rate. Even with only 6-8 feature elements per image, the classification rate remains at about the 90% level. In the comparison of different transform algorithms, the wavelet subband and the uniform subband have the highest classification rates compared to the DCT/Mandala transform. The widely used DCT has a decent classification rate, about 85%. The slightly lower classification rate by using the DCT transform is basically due to its less effective energy concentration capability.

The above task of texture classification operates on the entire image and does not require texture segmentation. However, in order to discriminate distinctive local features, identification of local image regions of prominent features is necessary. We recognize that robust texture segmentation is a research issue still in progress. We also argue that for texture-based image query applications, accurate boundary information is not really necessary. Therefore, we relax the requirement and just aim to extract homogeneous image regions with prominent texture features. Using a modified quad-tree and the threshold derived in [6], we were able to use the transform-domain texture feature to extract prominent regions from each image in the database. One image may have zero or multiple prominent homogeneous texture regions. Given the input image key, the texture feature vector is derived from the transform domain and compared against every region contained in every image in the database. Images containing the most similar image regions were returned as matches.

Compared to the traditional technique for texture extraction, such as pixel-domain window-based Law filters, our transform-domain texture features also include local window-based filtering. For example, a 8×8 DCT transform has a local window size of 64 pixels. Determination of the transform size actually involves an interesting tradeoff between texture homogeneity and statistical confidence. Larger transform block sizes give higher statistical confidence, but potentially lower homogeneity of the texture feature.

2.2 Image Matching

Image matching has been used in many applications including image registration, pattern recognition, and stereoscopic image correspondence matching. Two critical factors in image matching are determination of the matching criterion and the search space. One example is the minimal distortion matching used in the popular motion estimation algorithm for video coding. In [5], we have derived algorithms for doing motion estimation and inverse motion compensation in the DCT domain. For any orthonormal transforms like DCT, the Euclidean distance is preserved in the transform domain. However, because motion compensation is pixel-based while DCT is block-based, computation of the DCT of each reference block may involve significant overhead in realigning the DCT block structure. To compensate for this overhead, the search space may need to be reduced, using some heuristics such as the 3-point motion estimation technique.

If the images are encoded by wavelet or subband transforms, image matching can be implemented in an intelligent, hierarchical way as well. Suppose we adopt correlation as the matching criterion and use the exhaustive search space. Searching for the position with the highest correlation is equivalent to finding the peak value in the convolution. One can prove that the correlation criterion is closely related to the MSE or the correlation coefficient criterion. It has been shown that convolution of two 1-D sequences can be decomposed to the summation of convolutions of their subband components. Specifically, the summation of all intra-subband convolutions equals a subsampled version of the

complete convolution. In [8], we took a similar approach to implement a hierarchical image matching method. If $\{h_1, h_2\}$ and $\{g_1, g_2\}$ are subsampled low-band and high-band signal decomposition of the original sequences h and g. Their convolution can be calculated as (expressed in the z transform form)

$$h(z)g(z) = h_1(z^2)g_1(z^2)\,S_{11}(z) + h_2(z^2)g_2(z^2)\,S_{22}(z) + h_1(z^2)g_2(z^2)\,S_{12}(z) + h_2(z^2)g_1(z^2)\,S_{21}(z) \quad \text{(EQ 1)}$$

where S_{ij} are the product synthesis filters for each subband convolution. The above equation includes two intra-subband convolutions and two cross-subband convolutions. If the analysis filters are ideal half-band low-pass and high-pass filters, the cross-subband terms will be zero. For practical filters, such as the Harr filter and QMF filters, these terms are non-zero although they are relatively small compared to the intra-subband convolutions. In [8], we described an adaptive convolution scheme which adaptively approximates the complete convolution with the dominant subband convolutions. The criteria for choosing the dominant subband convolutions are based on two possible features — energy and feature. The energy-based approach chooses the subbands with the highest energy and approximate the complete convolution with the intra- and cross-subband convolutions associated with those dominant subbands. Note that the subband decomposition can be iterated more times in a uniform, logarithmic, or adaptive way to create signal decompositions at more levels. The above adaptive, hierarchical convolution can be easily repeated in each iteration. The hierarchical image searching method has been studied earlier in [7], but only low-low band convolution was used to approximate the complete result. One alternative criterion for choosing the significant subbands is to use the signal features, such as edge and texture, in each subband. For example, if one subband has strong indication of edge or texture content, it is better to include that subband in the approximation.

Another promising technique for image matching in the wavelet subband domain is to incorporate the zero-crossing representation. In [3], a stabilized zero-crossing representation was used in stereo image correspondence matching. It was shown that under certain conditions, the stabilized zero-crossing representation is complete and stable. A unique signal can be reconstructed from its stabilized zero-crossing representation. One interesting application is to use the distance of the stabilized zero-crossing representation to approximate the "distance" between two signals. Zero-crossing representation can be computed from the wavelet transform of a signal if the wavelet function is the second-order or first-order derivative of a smooth function. Since the stabilized zero-crossing representation inherently carries multi-scale signal features, typical hierarchical coarse-to-fine image matching techniques can be applied as well.

2.3 Video Indexing and Editing

Compared to still images, a video sequence can be further characterized by two additional "features" — (1)how the video is captured (i.e., the camera operations such as zooming and panning)? (2) how image features change over time (e.g., object motion and inter-scene temporal relationship)? There are existing techniques for extracting these dynamic visual features in the uncompressed domain. Work has been reported in [13, 14] to detect scene changes in the transform domain and the MPEG domain. Independently, we have applied the compressed-domain feature extraction principal and developed a *Compressed Video Editing and Parsing System (CVEPS)*, which allows automatic parsing of the MPEG-1 and MPEG-2 compressed video streams to detect scene changes, dissolve, and fade in/out. Abrupt scene changes can be detected by image intensity variance discontinuity and/or the distribution of the motion vectors in the B and P frames (e.g., the ratio among the numbers of forward predicted blocks, backward predicted blocks, and intraframe coded blocks). Dissolve scene changes can be characterized by modeling the image intensity variance with a quadratic form. Detection of scene change and dissolve requires establishment of some threshold values. We used an adaptive local threshold based on local video activities instead of a global threshold value. There is other useful information which can be derived from the compressed data. For example, a technique was proposed in [16] to approximate the object motion trajectory based on the motion vectors. Motion field has been used to detect and classify the camera operations as well [13].

Our CVEPS system currently also provides tools for cutting and pasting video segments at any random point directly in the MPEG compressed domain. Cutting and pasting MPEG video streams require solving two technical issues. First, the first few frames in the tail segment after cutting need to be re-encoded, unless the cut is exactly on the I frames. Second, the connection point in a paste operation may cause buffer overflow or underflow in the decoders (due to the rate control constraint used in MPEG). This can be solved by artificially inserting or deleting image frames near the connection points.

3. Compressed-Domain Image Manipulation

We extend the above compressed-domain approach to image manipulation in this section. Image manipulation involves many useful operations for general multimedia applications, as mentioned in Section I. In general, it includes linear and non-linear operations. We have been focusing on the compressed-domain solutions for linear operations, such as filtering, geometrical transformation, multi-object composition, pixel multiplication, and convolution.

In [1, 5], we have derived a set of algorithms for doing all above operations in any separable orthogonal transform domain, such as DCT. As an example, two dimensional separable linear filtering of the images can be expressed as

$$Y = \sum_i W_i P_i H_i \quad , \quad \text{(EQ 2)}$$

where P_i is the input image blocks, H_i and W_i are filter coefficient matrices in the horizontal and vertical directions respectively, and Y is the output filtered image block. Using the distributive property of separable orthogonal transform with respect to matrix multiplication, we can perform the same linear filtering operation in the transform domain directly, i.e.,

$$T(Y) = \sum_i T(W_i)\,T(P_i)\,T(H_i) \quad . \quad \text{(EQ 3)}$$

In other words, given the transform coefficients of the input images, we can directly calculate the transform coefficients of the output filtered images directly in the transform domain by using

the above formula. If the transform coefficients of the input images have been truncated (as done in quantizers of practical coding methods), a great number of small coefficients may be truncated to zeros. Therefore, the computational complexity associated with the compressed-domain operations (EQ 3) could be greatly reduced. In a test scenario which takes three input image sources, scaled each of them to different sizes, and translate them to different locations in the final composited scene, we have been able to reduce overall computational complexity by about 65% by using the proposal transform-domain image manipulation approach, compared to the traditional uncompressed-domain approach.

To extend the image manipulation techniques to the motion-compensation domain is not directly feasible, due to the complication of the motion compensation algorithm. In [5], we have provided a partial solution which applied the transform-domain inverse motion compensation to convert the input video to the transform domain and kept the manipulation operations in the transform domain. This will incur some overhead associated with the transform-domain (inverse) motion compensation, whose net impact on the overall computation cost actually depends on the motion vector distribution for each specific input video stream.

Some operations, such as shearing and rotation, cannot be directly modeled by a linear operation like that in EQ 2. In general, they require different operations on different rows and columns. This problem can be solved by using the divide-and-conquer approach described in [18]. The idea is to extract each row (or column) first and then apply the same linear operation mentioned above.

4. Conclusions and Future Work

We have presented an overview of various compressed-domain image technologies for image manipulation and searching in this paper. We believed by taking advantage of some nice properties of existing compression algorithms we will be able to provide some extent of content accessibility for today's compression algorithms. This will be a good evaluation criterion for comparing various existing image compression techniques.

As a more long-term research challenge, the following issue should be addressed. Given the desired image access and manipulation functions, such as content extraction, template matching, and image editing, how do we design the next-generation compression algorithm enabling efficient implementations of these functions directly in the compressed domain?

In the context of feature extraction for image query, one future direction is to find effective ways for integrating multiple features, such as color, texture, shape, and motion, in the same domain and to test them on concrete, specific applications. We believe by low-level signal features alone it will not be a sufficient solution. One critical component will be the integration of domain user knowledge and other complementary indexing techniques, such as text keywords. On the image analysis research front, techniques for defining visual features that are invariant to geometry and noise will be crucial as well.

5. Acknowledgment

The author would like to thank Prof. David Messerschmitt, Dr. Harold Stone, John Smith, Jianhao Meng, and Haulu Wang for their advice and contribution to this work.

6. References

[1] S.-F. Chang, W.-L. Chen, and D.G. Messerschmitt, "Video Compositing in the DCT Domain," IEEE Intern. Workshop on Visual Signal Processing and Communications, Raleigh, North Carolina, September, 1992.

[2] 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.

[3] S. Mallat, "Zero-Crossing of a Wavelet Transform," IEEE Transactions on Information Theory, Vol. 37, No. 4, July 1991, pp.1019-33.

[4] P. Brodatz, Textures: a Photographic Album for Artists and Designers, Dover, New York, 1965.

[5] 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, Vol. 13, No.1, Jan. 1995.

[6] J.R. Smith and S.-F. Chang, "Quad-Tree Segmentation for Texture-Based Image Query" *Proceedings*, ACM 2nd Multimedia Conference, San Francisco, Oct. 1994.

[7] E.L. Hall, R.Y. Wong, and J. Rouge, "Hierarchical Search for Image Matching," IEEE Decision and Control Conference, 1976, pp. 791-796.

[8] H. Wang and S.-F. Chang, "Adaptive Hierarchical Image Matching in the Subband Domain," Submitted to SPIE Symposium on Electronic Imaging 1996, San Jose.

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

[10] A. Eleftheriadis and D. Anastassiou, "Constrained and General Dynamic Rate Shaping of Compressed Digital Video," 2nd IEEE International Conference on Image Processing (ICIP-95), Washington, DC, October 1995 (to appear).

[11] J.B. Lee and B.G. Lee, "Transform Domain Filtering Based on Pipelined Structure," IEEE T. on Signal Processing, pp. 2061-4, Vol. 40, No. 8, Aug. 1992.

[12] R. W. Picard, "Content Access for Image/Video Coding: *The Fourth Criteri*on," MIT Media Lab. Perceptual Computing Section Technical Report, No. 295.

[13] H. Zhang A. Kankanhalli, S.W. Smoliar," Automatic Parsing of Full-Motion Video," ACM-Springer Multimedia Systems 1993, 1(1), pp. 10-28.

[14] F. Arman, A. Hsu, and M.-Y. Chiu, "Image Processing on Compressed Data for Large Video Databases," Proceedings of ACM Multimedia Conference, June 1993.

[15] Y.Y. Lee and J. Woods, "Video Post Production with Compressed Images," SMPTE. J. Vol. 103, pp. 76-84, Feb. 1994.

[16] N. Dimitrova and F. Golshani, "Rx for Semantic Video Database Retrieval," ACM Multimedia Conference, 1994, Oct., San Francisco.

[17] 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.

[18] S.-F. Chang, "New Algorithms for Processing Images in the Transform-Compressed Domain, " SPIE Symposium on Visual Communications and Image Processing, Taipei, May, 1995.