Published in Image and Vision Computing Journal, Vol. 17, Issue 7, pp. 513-529, May 1999

A Critical Evaluation of Image and Video Indexing Techniques in the Compressed Domain

M. K. Mandal, F. Idris and S. Panchanathan
Visual Computing and Communications Laboratory
Department of Electrical and Computer Engineering
University of Ottawa, Ottawa, Canada - K1N 6N5
E-mail: mandal@trix.genie.uottawa.ca
URL: http://www.vccl.uottawa.ca/~mandalm/index.html
Tel. - (613) 562-5800, Ext. 6206, Fax - (613) 562-5175

Abstract

Image and video indexing techniques are crucial in multimedia applications. A number of the indexing techniques that operate in the pixel domain have been reported in the literature. The advent of compression standards has led to the proliferation of indexing techniques in the compressed domain. In this paper, we present a critical review of the compressed domain indexing techniques proposed in the literature. These include transform domain techniques using Fourier transform, Cosine transform, Karhunen-Loeve transform, Subbands and Wavelets; and spatial domain techniques using Vector Quantization and Fractals. In addition, temporal indexing techniques using motion vectors are also discussed.

I. INTRODUCTION

Digital image and video indexing techniques are becoming increasingly important with the recent advances in very large scale integration technology (VLSI), broadband networks (ISDN, ATM), and image/video compression standards (JPEG/MPEG). The goal of image indexing is to develop techniques that provide the ability to store and retrieve images based on their contents [1]. Some of the potential applications of image and video indexing are: multimedia information systems [2], digital libraries [3], remote sensing and natural resources management [4], movie industry and video on demand [5]. Traditional databases use keywords as labels to quickly access large quantities of text data. However, the representation of visual data with text labels needs a large amount of manual processing and entails extra storage. A more serious problem is that the retrieval results might not be satisfactory since the query was based on features that may not reflect the visual content. Hence, there is a need for novel techniques for content based indexing of visual data.

Several content based image retrieval systems have been proposed in the literature [6]-[11]. A block schematic of a typical image archival and retrieval system is shown in Fig. 1. A multidimensional feature vector is generally computed for each image, and indexing is performed based on the similarities of the feature vectors. Since the interpretation/quantification of various features are fuzzy, emphasis is typically placed on the similarity rather than the exactness of the feature vectors. In indexing applications, a feature is selected based on the following performance criteria: i) its capacity to distinguish different images, ii) the maximum number of images a query could possibly retrieve, and iii) the amount of computation required to compute (or the amount of space required to store them) and compare the features.

Typically, visual indexing techniques are based on features such as histogram, color, texture, *etc*. (see Fig. 3a). Here, the image features are extracted directly from the image pixels. Recently, image and video compression standards such as JPEG [12], MPEG [13] and H.261 [14], have been proposed to reduce the bandwidth and storage requirements. Hence, images and videos are expected to be stored in compressed form. This has led to the proliferation of a number of compressed domain indexing techniques in the literature. Here, indexing is performed directly on the compressed data (see Fig. 2).

Recently, Aigrain *et al*. [15] have surveyed the approaches for different types of visual content analysis, representation and their application in indexing, retrieval, abstracting, relevance assessment, and interactive perception. However, the review on indexing techniques is brief. Idris *et al*. [16] have presented a review of image and video indexing techniques pointing out the advantages and disadvantages of each approach. The review is mainly based on pixel-domain techniques. Ahanger *et al*. [17] have reviewed present research trends in multimedia applications and the requirements of future data delivery systems which includes a review of few video segmentation techniques. The video segmentation techniques discussed mainly operate in pixel domain, although a brief discussion of DCT (discrete cosine transform) based techniques has also been presented. In this paper, we provide a critical review of existing compressed domain indexing techniques (see Fig. 3b). The main focus of this paper will be on techniques for deriving feature vectors in the compressed domain.

The organization of the paper is as follows: a brief review of pixel domain indexing techniques is presented in section 2. In section 3, a review of compressed domain indexing techniques is presented. A brief review of pixel domain video indexing techniques is presented in section 4. The review of video indexing techniques in compressed domain is presented in section 5, followed by the conclusions.

II. IMAGE INDEXING IN PIXEL DOMAIN

The pixel domain indexing of visual data are based on features such as texture, shape, sketch, histogram, color, moments, *etc*. For example, the *Query By Image Content* (QBIC) system developed by IBM [7] retrieves images based on color, texture, shape, and sketches. The *Content-based Retrieval Engine* (CORE) for Multimedia Information Systems proposed by Wu *et al*. [9] employs color and word similarity measures to retrieve images based on content and text annotation, respectively. We now briefly describe the state of the art approaches in image indexing.

Color: Color is one of the important features of an image. Typically, the color of an image is represented using the image histogram. The histogram of an image with colors in the range [*0, L-1*] is a discrete function $p(i) = n_i / n$, where i is the color of a pixel, n_i is the number of pixels in the image with color i, n is the total number of pixels in the image, and i = *0, 1, 2, ..., L-1*. In general, $p(i) = n_i / n$ gives an estimate of the probability of occurrence of color *i*.

In image retrieval using color histogram, the histogram of the query image is matched against the histograms of the images in the database. The matching process is carried out using a similarity metric. The common similarity metrics employed for evaluating color similarity are histogram intersection [18], and weighted distance between color histograms [19]. The complexity of the matching process can be reduced by quantizing the color space [18], use of the dominant features of a histogram [20], use of a lower dimensional histogram by representing the color histogram at different resolutions [21], and the use of a lower complexity metric [19]. The retrieval performance can be improved by taking into account the location of the colors in the color representation of an image [20]. However, this technique requires the use of efficient segmentation and representation of the sub-images.

Texture: An image can be considered as a mosaic of different texture regions, and the image features associated with these regions can be used for search and retrieval. The term *texture* generally refers to repetition of basic texture elements called texels [22]. A texel contains several pixels and can be periodic, quasi periodic or random in nature. Texture modeling and classification are broadly grouped into three main categories [23]; structural, statistical, and spectral.

Recently, several techniques for image indexing based on texture features have been reported. Picard *et al*. [24] have presented a technique based on Wold decomposition which provides a description of textures in terms of periodicity, directionality and randomness. A modified set of the Tamura features (Coarseness, Contrast and Directionality) [25] have been used in the QBIC project [7]. Zhang *et al*. [26] have proposed a technique based on a multiresolution autoregressive model, Tamura features, and gray level histogram. Rao *et al*. [27] have studied the relationships between categories of texture images and texture words. Retrieval by texture is useful when the user is interested in retrieving images which are similar to the query image. However, the use of texture features requires texture segmentation which remains a challenging and computationally intensive task. In addition, texture based techniques lack robust texture models and correlation with human perception.

Sketch: A sketch is an abstract image that contains the outline of objects. In this approach, the users may provide a rough sketch of the query image. Typically, a sketch is created by using edge detection, thinning and shrinking algorithms. The sketch of a query image is used as a key to retrieve the desired images from the database. The similarity of two images is measured by using the similarity of their sketches based on local and global correlation measures [23]. A technique for sketch based image retrieval has been proposed by Kato *et al*. [28] and is implemented in the QBIC [7]. The disadvantage of this approach is that it is orientation and scale dependent. Similar images with different orientation or scale will not be retrieved when compared with the query image. This problem can be eliminated by using sophisticated edge representation and matching algorithms.

Shape: Shape is an important criterion for matching objects based on their profile and physical structure. In image retrieval applications, shape features can be classified into global and local features. *Global features* are the properties derived from the entire shape such as roundness, circularity, central moments, and eccentricity [29]. *Local features* are those derived by partial processing of a shape including size and orientation of consecutive boundary segments [30], points of curvature, corners and turning angle [29]. Shape features are fundamental to systems such as medical image databases where the color and textures of objects are similar. However, retrieval by shape similarity is a difficult problem because of the lack of mathematically exact definition of shape similarity that accounts for the various semantic qualities that humans assign to shapes.

Spatial Relationships: In this technique, objects and their spatial relationships among objects in an image are used to represent the content of an image. First, objects in an image are segmented and recognized. The image is then converted into a symbolic picture that is encoded using two-dimensional (2-D) strings [31]-[34]. We note that 2-D string represents relationships among the objects in the image and is expressed using a set of operators (e.g., left, right, above, etc.). The problem of image retrieval thus becomes a problem of 2-D sequence matching. The basic algorithm for image indexing using spatial relationships was presented by Chang *et al*. [31]. Jungert *et al*. [32]. have extended the basic 2-D string to increase the range of relationships that can be expressed, especially among overlapping objects. Chang *et al*. [33] have presented a generalization of the 2-D string called 2-D G-String to reduce the number of partitions required for representing overlapping objects. Lee *et al*. [34] have proposed the 2-D B-string for image indexing without the need for object partitioning. We note that matching 2-D strings is based on a simple ranking scheme. However, the generation of a 2-D string is based on object segmentation and recognition which is compute intensive.

III. IMAGE INDEXING IN THE COMPRESSED DOMAIN

The large volumes of visual data necessitate the use of compression techniques. Hence, the visual data in future multimedia databases is expected to be stored in the compressed form. In order to obviate the need to decompress the image data and apply pixel-domain indexing techniques, it is efficient to index the image/video in the compressed form**.** Compressed domain image/video indexing techniques based on compression parameters have been reported in the literature. These techniques have a lower cost for computing and storing the indices. Compressed domain indexing (CDI) techniques can be broadly classified into two categories: transform domain techniques, and spatial domain techniques. The transform domain techniques are generally based on DFT (discrete Fourier transform), KLT (Karhunen-Loeve transform), DCT, and Subbands/Wavelets. Spatial domain techniques include vector quantization (VQ) and fractals. We now present a review of compressed domain image indexing techniques.

Discrete Fourier Transform

Fourier transform is very important in image and signal processing. DFT employs complex exponential basis functions and provides a good coding performance since it has good energy compaction property. DFT has several properties that are useful in indexing or pattern matching. Firstly, the magnitude of the DFT coefficients are translation invariant. Secondly, the spatial domain correlation can be efficiently computed using DFT coefficients. We now present selected Fourier-domain indexing techniques.

Stone *et al*. [35] have proposed and evaluated an image retrieval algorithm in Fourier domain. The algorithm has two thresholds that allow the user to independently adjust the closeness of a match. One threshold controls an

intensity match while the other controls a texture match. The thresholds are correlation values that can be computed efficiently using the Fourier coefficients and are particularly efficient when the Fourier coefficients are mostly zero.

Several texture measures have been evaluated by Augustejin *et al.* [36] for classification of satellite images. The measures are based on the magnitude of the Fourier spectra of an image. The statistical measures include i) maximum magnitude, ii) average magnitude, iii) energy of magnitude, and iv) variance of magnitude of Fourier coefficients. In addition, the authors have also studied the retrieval performance based on the radial and angular distribution of Fourier coefficients. We note that the radial distribution is sensitive to texture coarseness whereas the angular distribution is sensitive to directionality of textures. It was observed that the radial and angular measures provide a good classification performance when a few dominant frequencies are present. The statistical measures provide a satisfactory performance in the absence of dominant frequencies.

Celantano *et al.* [37] have evaluated the performance of angular distribution of Fourier coefficients in image indexing. Here, the images are first pre-processed with a lowpass filter and the FFT is calculated. The FFT spectra is then scanned by a revolving vector exploring 180^{o} range. The angular histogram is calculated by computing the sum of image components contribution for each angle. While calculating the sum, only the middle frequency range is considered as they represent visually important image components. The angular histogram is used as the feature vector for indexing. The feature vector is independent of translation in pixel domain while the rotation in pixel domain corresponds to a circular shift in the histogram.

Karhunen-Loeve Transform

Karhunen-Loeve transform (Principal Component Analysis), is based on the statistical properties of an image. Here, the basis functions are the eigenvectors of the autocorrelation matrix of the image. KLT provides maximum energy compaction and is statistically the optimum transform. Since the KLT basis functions are image adaptive, a good indexing performance is obtained by projecting the images in K-L space and comparing the KLT coefficients.

Pentland *et al.* [38] have proposed a KLT-based technique for face recognition. Here, a set of optimal basis images, *i.e.*, *eigenfaces*, is created based on a randomly chosen subset of face images. A query image is then projected onto the *eigenfaces*. Faces are recognized based on the Euclidean distance between the KLT coefficients of the target and query image. Since the KLT basis images are ordered with respect to the eigen values, the salient image characteristics can be well represented by using a first few (15-20) KLT coefficients.

The projection to K-L space extracts the *Most Expressive Features* (MEFs) of an image. However, an eigenfeature may represent aspects of the imaging process, such as illumination direction, which are unrelated to recognition. An increase in the number of eigenfeartures does not necessarily lead to an improved success rate. To address this issue, Swets *et al.* [39] have proposed a Discriminant Karhunen Loeve (DKL) projection where KLT is followed by a discriminant analysis to produce a set of *Most Discriminating Features* (MDFs). In DKL projection, between-class scatter is maximized, while the within-class scatter is minimized. The authors have reported an improvement of 10-30% using DKL technique (over KLT) on a typical database.

KLT has also been applied to reduce the dimensionality of features derived from a texture for classification. We note that several methods [40] exist for texture classification, such as spatial gray level dependence matrix (SGLDM), gray level run-length method (GLRLM), and power spectral method (PSM). Tang *et al.* [40] have showed that KLT is efficient in reducing the dimensionality of the feature vectors.

Although, KLT has the potential to provide good performance, it has been tested on small databases. Therefore, detailed investigation has to be performed to gain insights on how to generate feature vectors for a large database with widely varying characteristics. We note that KLT is generally not used in traditional image coding because of higher complexity. However, it is employed in analyzing and encoding multispectral images [41] and has therefore a potential for indexing in remote sensing applications.

Discrete Cosine Transform

DCT, a derivative of DFT, employs real sinusoidal basis functions [22] and has energy compaction efficiency close to the optimal KL transform for most natural images. As a result, all international image and video compression standards, such as JPEG, MPEG 1 and 2, H.261/H.263, employ DCT. We now provide a brief description of DCT-

based JPEG [12] baseline algorithm since all the above mentioned standards employ a similar algorithm for coding. In JPEG, compression is performed in three steps (see Fig. 4): DCT computation, quantization and variable-length coding. The original image is first partitioned into non-overlapping blocks of 8x8 pixels as shown in Fig. 4. The 2-D DCT of the block is then computed and quantized using a visually adapted quantization table suggested by JPEG. Each 8x8 block generates one DC coefficient and 63 AC or, high frequency coefficients. The quantized coefficients are reordered using zigzag scan pattern to form a 1-D sequence of quantized coefficients. The DC coefficients from each block is DPCM coded and all other coefficients, *i.e.*, the AC coefficients are compressed using a combination of Huffman and run-length coding. We note that the above mentioned DCT-based standards do not address the aspect of indexing. We now discuss some DCT-based indexing techniques that have appeared in the recent literature.

Smith *et al*. [42] have proposed a DCT based method where the image is divided into 4x4 blocks and the DCT is computed for each block resulting in 16 coefficients. The variance and the mean absolute values of each of these coefficients are calculated over the entire image. The texture of the entire image is then represented by this 32 component feature vector. The authors use a Fisher discriminant analysis (FDA) to reduce the dimensionality of the feature vector. We note that FDA generates a family of linear composites from the original feature vectors that provide for maximum average separation among training classes. The reduced dimension feature vector is used for indexing.

Reeves *et al*. [43] have proposed a DCT-based texture discrimination technique which is similar to that of Smith *et al* [42]. Here, the image is divided into 8x8 blocks. A feature vector is formed with the variance of the first 8 AC coefficients. The technique does not employ the mean absolute value of the DCT coefficients, as in [42]. The technique assumes that the first AC coefficients have the most discriminating features, and thus avoids discriminant analysis used in [42]. The run-time complexity of this technique is smaller than that of [42], since the length of the feature vector is small.

Shneier *et al*. [44] have proposed a technique for image retrieval using JPEG. This technique is based on the mutual relationship between the DCT coefficients of unconnected regions in both the query image and target image. Here, a set of $2K$ windows is selected, and is randomly paired, producing K pairs of windows. For each window the average of each DCT coefficient is computed resulting in a 64-dimensional feature vector (f). The feature vectors corresponding to a pair of windows are compared and each pair of components is assigned a bit (0 or 1) depending on their similarity. Thus, each pair of windows will be assigned 64 bits. The similarity of the query and target image is determined by the overall similarity of all the bits in all window pairs.

Many content-based indexing and retrieval methods are based on the discrimination of edge information. Abdel-malek *et al*. [45] have proposed a technique to detect oriented line features using DCT coefficients. The technique is based on the observation that predominantly horizontal, vertical, and diagonal features produce large values of DCT coefficients in vertical, horizontal, and diagonal directions, respectively. The authors report that a straight line of slope m in spatial domain generates a straight line with a slope of approximately $1/m$ in the DCT domain. The technique can be extended to search more complex features composed of straight-line segments.

A segmentation technique [46] using local variance of DCT coefficients was proposed by Ng *et al*. Here, 3x3 DCT is computed at each pixel location using the surrounding points. The local variance of each DCT coefficient is then computed using a 15x15 sliding window. Changes in the local variance are used to segment the image.

Shen *et al*. [47] have proposed techniques to detect regions of interest and edges in images from the high frequency JPEG DCT coefficients. The technique estimates *edge orientation*, *edge offset from center*, and *edge strength* from DCT coefficients of a 8x8 block. The orientations include horizontal, vertical, diagonal, vertical dominant, and horizontal dominant. Experimental result [47] shows that the DCT based edge detection provides a performance comparable to the Sobel edge detection operator.

Subbands/Wavelets

Recently, subband and discrete wavelet transforms (DWT) have become popular in image coding and indexing applications [48,49]. Here, an image is passed through a set of lowpass and highpass filters, recursively, and the filter outputs are decimated in order to maintain the same data rate. In DWT, the lowpass output is recursively filtered (see Fig. 5). Gabor transform is similar to wavelet transform, where the basis functions are Gaussian in nature and hence

Gabor Transform is optimal in time-frequency localization. Since, most of the energy in the subband domain is represented by a few lowpass coefficients, high compression ratio is achieved by discarding the high frequency coefficients. Subband coding is generally implemented using quadrature mirror filters (QMFs) in order to reduce the aliasing effects arising out of decimation. We note that the entire data is passed through the filters, and there is no blocking of data as in JPEG. Subband decomposition has several advantages in coding - i) multiresolution capability, ii) better adaption to nonstationary signals, iii) high decorrelation and energy compaction efficiency, iv) reduced blocking artifacts and mosquito noise, and v) better adaptation to the human visual system characteristics.

Chang *et al.* [50] have proposed a texture analysis scheme using irregular tree decomposition where the middle resolution subband coefficients are used for texture matching. In this scheme, a J dimensional feature vector is generated consisting of the energy of J most important subbands. Indexing is done by matching the feature vector of the query image with those of the target images in a database. For texture classification, superior performance can be obtained by training the algorithm. Here, for each class of textures, the most important subbands and their average energy are found by the training process. A query image can then be categorized into one of the texture classes, by matching the feature vector with those of the representative classes.

A texture discrimination technique has been proposed by Smith *et al.* [42], where the energy of the subbands are used to define the texture feature sets. The performance of DCT and subbands has been compared with that of pixel domain techniques. For an $N \times N$ DCT transform, N^2 bands are obtained using the DCT/Mandala transform. For a 3-level DWT, feature vectors with 10 terms are produced. The texture feature vector is reduced by using Fisher discriminant technique. The Mahalanobis distance in the transformed feature space is used to measure the similarity between two images. Brodatz texture set was used for this experiment. The classification performance is as follows: uniform subband (92%), wavelets (92%), 4x4 DCT/Mandala (85%), 4x4 pixel domain (34%). In summary, it is observed that wavelets provide a superior texture classification performance compared to other transforms in the given framework.

Chen *et al.* [51] have proposed a rotation and gray scale transform invariant texture recognition technique using wavelets and hidden Markov model (HMM). In the first stage, the gray scale transform invariant features are extracted from each subband. In the second stage, the sequence of subbands is modeled as a HMM, and one HMM is designed for each class of textures. The HMM is used to exploit the dependence among these subbands, and is able to capture the trend of changes caused by rotation. During recognition, the unknown texture is matched against all the models and the best match model identifies the texture class.

Mandal *et al.* have proposed to compare the histograms of directional subbands to find a match with the query image [52]. It has been shown that the histograms of wavelet bands of similar images, with limited camera operations, are similar. The images can be discriminated by the amount of horizontal, vertical, and diagonal information at different scales. Different images might have similar overall histograms, but they are unlikely to have similar band statistics. The complexity of direct comparison of the histograms of all the subbands is high. This complexity is reduced substantially by matching the distribution parameters of the subbands. The *pdf*s (or histograms) of highpass wavelet subbands can be modeled using generalized Gaussian density (GGD) function [53] which is expressed in terms of two parameters - s (standard deviation) and g (shape parameter). Hence, the dissimilarity between a target and query image can be expressed in terms of the difference of the band parameters, *i.e.*,

$$d(f,g) = \sum_{k=1}^{Q} B_k \left(g_{f_k} - g_{g_k} \right)^2 + \sum_{k=1}^{Q} A_k \left(s_{f_k} - s_{g_k} \right)^2 \qquad (1)$$

where Q is the number of wavelet bands used for comparison. A_k and B_k are the weights of the parameters and are estimated by trial and error procedure to achieve the best performance. The images that have minimum distance are retrieved from the database.

Most of the indexing algorithms presented in the literature assume that the illumination level of the images are similar. The indexing performance may substantially degrade if the above assumption is violated. Mandal *et al.* [54] have proposed a histogram-based technique in the wavelet domain that is robust to changes in illumination. In this technique, the change in the illumination level is estimated using scale invariant moments of the histogram. The

subband parameters - s and g of each subband of the target image are then changed appropriately to counter the effect of illumination change. Eq. (1) can then be used for matching images.

An indexing technique [55] using Gabor wavelets was proposed by Manjunath *et al*. Here, each image is decomposed into four scales and six orientations. A feature vector, of dimension 48, is then formed using the mean (m) and standard deviation (s) of each subband. The similarity of the query image and a target image is determined by the similarity of their feature vectors. In this technique, the number of orientations are more, i.e., six, compared to three orientations (horizontal, vertical and diagonal) in the wavelet domain. Hence, better directional discrimination is achieved with this technique. However, the Gabor wavelets are computationally expensive compared to dyadic wavelets.

Wang *et al*. [56] have proposed a correlation based pattern matching in the subband domain. They have derived an expression for correlating two spatial domain functions in terms of their subband coefficients. This is executed as follows: Consider two one-dimensional signals $x(n)$ and $w(n)$. The correlation of the signals can be represented in z-transform domain as $\tilde{W}(z)X(z)$. Let the lowpass and highpass analysis filters be $H_0(z)$ and $H_1(z)$, respectively. Suppose the subband decomposition of $x(n)$ and $w(n)$ are $\{y_0(n),\ y_1(n)\}$ and $\{z_0(n),\ z_1(n)\}$, respectively. The correlation of the signal can then be represented as:

$$\tilde{W}(z)X(z) = \sum_{i,j=0}^{1} F_{ij}(z)\tilde{Z}_i(z^2)Y_j(z^2) \qquad (2)$$

where $F_{ij}(z) = H_i(z)\tilde{H}_j(z)$, $i, j = 0,1$

Eq. (2) shows that the correlation of two signals equals the weighted sum of their correlation in subbands. It was conjectured in [56] that to get a reasonable estimate of the correlation peaks, computation can be done on a few subbands with high energy, resulting in a substantial reduction in complexity. In addition, subband synthesis is not required since the coefficients also provide spatial information. This is in contrast to Fourier domain techniques where inverse transform must be performed to find the correlation peaks.

Jacobs *et al*. [57] have proposed an indexing technique based on direct comparison of DWT coefficients. Here, all images are rescaled to 128×128 pixels followed by wavelet decomposition. The average color, the sign (positive and negative) and indices of M (the authors have used a value of 40-60) largest magnitude DWT coefficients of each image are calculated. The indices for all of the database images are then organized into a single data structure for fast image retrieval. A good indexing performance has been reported in the paper. However, the index is dependent on the location of DWT coefficients. Hence, the target images which are translated and rotated versions of the query image, may not be retrieved using this technique.

Wang *et al*. [58] have proposed a technique which is similar to that of Jacob *et al* [57]. Here, all images are rescaled to 128×128 pixels followed by a four stage wavelet decomposition. Let the four lowest resolution subimages, which are of size 8×8, be denoted by S_L (lowpass), S_H (horizontal band), S_V (vertical band), and S_D (diagonal band). Image matching is then performed using a three-step procedure. In the first stage, 20% of the images are retrieved based on the variance of S_L band. In the second stage, a fewer number of images will be selected based on the difference of S_L coefficients of query and target images. Finally, the images will be retrieved based on the difference of S_L, S_H, S_V and S_D coefficients of query and target images. For color images, this procedure is repeated on all three color channels. The complexity of this technique is small due to hierarchical matching. The authors have reported an improvement of performance over Jacob's technique [57]. However, as in Jacob's technique, the indexing performance is not robust to translation and rotation.

Qi *et al*. [59] have proposed a complex wavelet transform where the magnitude of the DWT coefficients are invariant under rotation. The mother wavelet is defined in the polar coordinates. An experiment on a set of English character images shows that the proposed technique performs better than complex Zernike moments (whose magnitude are also rotation invariant). Rashkovskiy *et al*. [60] have proposed a class of nonlinear wavelet transforms

which are invariant under scale, rotation and shift (SRS) transformations. This wavelet transform adjusts the mother wavelet for every input signal to provide SRS invariance. The wavelet parameters or the wavelet shape are iteratively computed to minimize an energy function for a specific application.

Froment *et al*. [61] have proposed a second generation image coding technique that separates edges from the texture information. Multiscale edges are detected from the local maxima of the wavelet transform modulus. An error image is computed by subtracting the reconstructed image from the original image, which mostly provides the texture information. The textures are coded with a standard orthogonal wavelet transform. The multiscale edges have the potential to provide good indexing performance.

Vector Quantization

In coding theory, it is well known that better performance can be achieved by coding vectors instead of scalars. A vector quantizer (VQ) is defined [62] as a mapping Q of K-dimensional Euclidean space into a finite subset Y of , *i.e*.,

$$Q : R^K \rightarrow Y$$

where $Y = \{x_i'; i = 1,2,\ldots N\}$ is the set of reproduction vectors, and is called a VQ codebook or VQ table. N is the number of vectors in Y. A VQ encoder maps (see Fig. 6) each input vector onto one of a finite set of codewords (codebook) using a nearest neighbor rule, and the labels (indices) of the codewords are used to represent the input image. Hence, VQ is naturally an indexing technique.

Two image indexing techniques [63, 64] using vector quantization have been proposed by Idris *et al*. In the first technique [63], the images are compressed using vector quantization and the labels are stored in the database. The histograms of the labels are used as feature vectors for indexing. For an image of size $X \times Y$ pixels, the computation of histogram in pixel domain has a complexity of $O(X * Y)$, whereas the computation of label histogram has a complexity of $O(X * Y / L)$ where L is the dimension of a codevector. Hence, the computation of histogram is less complex in the VQ domain. The second technique [64] has been proposed for adaptive VQ where a large codebook is used. In this case, a usage map of codewords is generated for each image and is stored along with the image. We note that the usage map reflects the content of the image. Hence, the usage map of the VQ encoded query image is compared with the usage map of the target images in the database for indexing. The runtime complexity of this technique is only $O(K)$ bit-wise operations, where K is the size of the codebook [64]. Although, both techniques provide good indexing performance, the former has been shown to outperform the latter.

Barbas *et al*. [65] have investigated the problem of efficient representations of large databases of radar returns in order to optimize storage and search time. The technique employs multiresolution wavelet representation working in synergy with a tree structured vector quantization, utilized in its clustering mode. The tree structure is induced by the multiresolution decomposition of the pulse. The technique has been shown to provide a good overall performance.

Vellaikal *et al*. [66] have applied VQ technique for content-based retrieval of remote sensed images. Here, various distortion measures have been evaluated to enhance the performance of the VQ codewords as *content descriptors*. Two types of query: *query-by-class* and *query-by-value*, were tested with the proposed technique. It has been found that the second query type provide excellent performance while the first query type provides satisfactory performance.

Fractals/Affine Transform

A fractal is a geometric form where irregular details recur at different scales and angles which can be described by a transformations (*e.g*. an affine transformation). Fractal image compression [67] is the inverse of fractal image generation, *i.e*. instead of generating an image from a given formula, fractal image compression searches for sets of fractals in a digitized image which describe and represent the entire image. Once the appropriate sets of fractals are determined, they are reduced to very compact fractal transform codes or formulas. In block fractal coding, an image is partitioned into a collection of nonoverlapping regions known as range blocks. For each range block, a domain block and an associated transformation are chosen so that the domain block best approximates the range. These

transformations are known as *fractal codes*. While the pixel data contained in the range and domain blocks are used to determine the code, they are not part of the code, resulting in a high compression ratio.

Zhang *et al*. [68] have proposed a texture-based image retrieval technique that determines image similarity based on the match of fractal codes. Here, each image is decomposed into block-based segments that are then assembled as a hierarchy based on inclusion relationships. Each segment is then fractally encoded. The fractal codes of a query image are used as a key and are matched with the fractal codes of the images in a database. Retrieval is performed by applying searching and matching algorithms to the hierarchy of images in the database.

Zhang *et al*. [69] have also compared the performance of wavelet and fractals in image retrieval. In wavelet domain, the mean absolute value and variance of different subbands are used as the image features. In fractal domain, the authors have proposed a joint fractal coding of two images (M_1 and M_2). Here, the best approximation for range blocks of image M_1 is searched both in image M_1 and M_2. The similarity of M_1 and M_2 is estimated from the ratio of the number of best domain blocks found in M_1 and M_2. Based on simulation results, the authors have concluded that wavelets are more effective for images which contain strong texture features while fractals performs relatively well for various types of images. However, we note that this conclusion is valid in the given framework and the relative performance may change if other techniques are employed.

Ida *et al*. [70] have proposed a segmentation technique using fractal codes. The hypothesis includes three assumptions: i) if a domain block is in a region S, its range block will also be in S, ii) if a domain block is outside S, its range block will also be outside S, and iii) if a domain block includes the boundary of S, its range block will also include S, and the pixel pattern in the range block will be similar to that in the domain block.

Hybrid Schemes

In image and video compression, hybrid schemes generally refer to a combination of two or more basic coding schemes [49, 71]. These hybrid schemes exploit the advantages of the associated compression techniques and provide superior coding performance. For example, a wavelet-VQ scheme has been proposed in [49], while a wavelet-fractal scheme has been proposed in [71]. A few indexing techniques have been proposed in the hybrid framework which are presented below.

Idris *et al*. [72] have proposed a wavelet-based indexing technique using vector quantization (VQ). Here, the images are first decomposed using wavelet transform. This is followed by the vector quantization of wavelet coefficients. The codebook labels corresponding to an image constitute a feature vector that is then used as an index to store and retrieve the images.

Swanson *et al*. [73] have proposed a VQ-based technique for content based retrieval in wavelet domain. Here, each image to be stored in the database (see Fig. 7) is divided into 8x8 blocks. A segmentation algorithm is then applied to define image regions and objects. Each segmented region in the image is covered with the smallest possible rectangular collection of the previously defined 8x8 blocks. The collection of blocks is denoted a superblock. The superblocks are encoded by a combination of wavelets and vector quantization. The remaining image regions, *i.e.*, the 8x8 blocks which are not elements of a superblock, are coded using the JPEG algorithm. The authors have also proposed a joint text-based coding and indexing technique by minimizing a weighted sum of the expected compressed file size and the expected query response time. Each file is coded into three sections: a file header consisting of query terms, a set of indices denoting the locations of these terms in the file, and the remainder of the file. Each file header is constructed by concatenating the codeword for each query term which appears in that file. The order of the concatenation and the codeword lengths are based on the probability distributions of the query terms. Although, this technique was proposed for text-based indexing, it can be extended for image retrieval.

A VQ-based face recognition technique has been proposed by Podilchuk *et al*. [74] in DCT domain. Here, a block-DCT is first performed on a set of images and code vectors are formed from the DCT coefficients. The codebook is generated by *k*-means clustering algorithm. The retrieval performance has been evaluated based on feature selection, codebook size, and feature dimensionality. Although, the technique seems to be promising, the coding performance of the combined technique has not been evaluated.

We note that it is difficult to compare the performance of various indexing techniques. The success and popularity of an indexing technique generally depends on the performance of the associated coding techniques. KLT, although statistically optimal, is computationally intensive. In addition, the basis images need to be stored, resulting in a poor coding performance. The block DCT in JPEG provides a good coding and indexing performance. However, the block structure was not originally intended for indexing. The wavelet-based techniques are promising for indexing applications because of i) inherent multiresolution capability, ii) simple edge and shape detection, and iii) readily available directional information. It has been seen in [42] that the wavelet transform outperforms the DCT/Mandala transform in image classification. The vector quantization is a natural indexing technique where indexes are used in coding. The same indexes can also be employed in image retrieval. Although, fractals have a potential to provide good coding performance, the fractal codes are highly nonlinear and image dependent. Hence, direct use of fractal codes may not provide good retrieval rates. We note that the complexity of both VQ and fractal coding are highly asymmetric, *i.e*. encoding is compute intensive, while decoding is fast.

In contrast to textual database systems, image and video databases are required to evaluate properties of the data specified in a query. For example, to retrieve all images similar to a query image based on color, the color attributes (e.g., color histogram) of the query image has to be calculated.

IV. VIDEO INDEXING IN PIXEL DOMAIN

A video sequence is a set of image frames ordered in time. Generally video indexing refers to indexing of individual video frames based on their contents and the associated camera operations involved in the imaging process. We note that the image indexing techniques described in section 2 and 3 can be applied individually to index each frame based on their content. However, the neighboring frames in a video sequence in general are highly correlated. Hence, for computational efficiency, the video sequence is segmented in a series of *shots*. A *shot* is defined as a sequence of frames generated during a continuous operation and representing a continuous action in time and space. A frame in each shot is declared as a representative frame. Indexing is performed by applying the image indexing technique on representative frames from each shot. Each shot in a video sequence consists of frames with different scenes. There are two ways by which two shots can be joined together - i) abrupt transition, and ii) gradual transition. In abrupt transition, two shots are simply concatenated while in the gradual transition, additional frames may be introduced using editing operations such as fade in, fade out or dissolve. A good video segmentation technique should be able to detect shots with both types of transition.

The apparent motion in a video sequence can be attributed to camera or object motion. Motion estimation/compensation plays an important role in video compression. The objective is to reduce the bit rate by taking advantage of the temporal redundancies between adjacent frames in a video sequence. Typically, this is accomplished by estimating the displacement (motion vectors) of uniformly sized blocks between two consecutive frames. In general, motion vectors exhibit relatively continuos changes within a single camera shot, while this continuity will be disrupted between frames across different shots.

Detecting camera motion is becoming important with potential applications in low bit-rate video coding and video editing. We note that there are seven basic camera operations (see Fig. 8) - panning, tracking, tilting, booming, zooming, and dollying [75]. Since, both object motion and camera motion are reflected in the observed motion vectors of a block coding scheme, it is generally difficult to estimate the camera motion. However, several models have recently been proposed to improve the estimation. A review of camera motion estimation is outside the scope of this paper. Interested readers may refer to [16]. In this section, a brief review of video segmentation techniques in pixel domain is presented. A detailed review of video segmentation techniques in the compressed domain will be presented in section 5.

Pixel Intensity Matching: In this method, pixel intensities of the two neighboring frames are compared. For example, to detect a scene change between m-*th* and $(m+1)$-*th* frame, the distance between the two frames is calculated in L^k metric. If the distance exceeds a predetermined threshold [76], a scene change is declared at m-*th* frame. For color video sequences, the distance is calculated for all the three color channels. A scene change is declared if the overall change exceeds a threshold. In pixel intensity matching, it is difficult to distinguish a large change in a small

area and a small change in a large area. Hence, this method is sensitive to motion, and camera operations which might result in false detection.

Histogram Comparison: In this method, two consecutive frames are compared based on their histograms. There are two variations of this technique [77, 78]. *Difference of histograms* (DOH) measures the difference of histograms of the two frames in L^k metric. *Histogram of difference frame* (HOD) is the histogram of the pixel to pixel difference frame and measures the change between two frames f_m and f_n. The degree of change between f_m and f_n is large if there are more pixels distributed away from the origin. The DOH technique is insensitive to local object motion, however, it is sensitive to global camera operations such as panning and zooming and scene changes. The HOD technique is more sensitive to local object motion compared to DOH technique [79]. Histogram-based techniques fail when i) the histograms across different shots are similar, and ii) the histograms within a shot is different due to changes in lighting condition, such as flashes and flickering objects.

Block-Based Techniques: In this technique, each frame is partitioned into a set of k blocks. The similarity of the consecutive frames is estimated by comparing the corresponding blocks individually [79]. In *Block Histogram Difference* (BHD) technique, the blocks are compared with respect to histogram, whereas in *Block Variance Difference* (BVD) technique the blocks are compared with respect to variance. If the dissimilarity exceeds a threshold, a scene change is declared. The block-based technique emphasizes the local attributes (compared to the pixel matching and global histogram comparison that emphasizes the global attributes) and reduce the effect of camera flashes and other noises. The increased tolerance to slow camera and object movements results in a reduction in over-detected camera breaks. However, cuts may be misdetected between two frames that have similar pixel values, but different density functions.

Twin Comparison: The previous segmentation techniques are based on thresholding. With a single threshold, it is difficult to detect the two types of scene changes, namely *abrupt* and *gradual*. If the threshold is small, the cuts will be over-detected. On the other hand, if the threshold is large, gradual cuts will be undetected. A two-pass dual threshold algorithm, known as twin comparison algorithm, has been proposed in [77] to address this problem. In the first pass, a high threshold (T_h) is employed to detect abrupt cuts. In the second pass, a lower threshold (T_l) is used and any frame that has the difference more than this threshold is declared as a potential start of the transition. Once the start frame is identified, it is compared with the subsequent frames based on the cumulative difference. When this value increases to the level of the higher threshold (T_h), camera break is declared at that frame. If the value falls between the consecutive frames then the potential frame is dropped and the search starts all over.

V. VIDEO INDEXING IN COMPRESSED DOMAIN

In this section, we present a review of video segmentation techniques in DFT, DCT, KLT, DWT, VQ domains and hybrid approach (any combination of the three approaches). We note that motion vectors, not available for image indexing, is an important feature for video segmentation. Hence, a review of motion vector based video segmentation will also be presented.

DCT Coefficients

We recall that the international standards for image and video compression (JPEG, MPEG, H.261, and H.263) are based on DCT [12]-[14]. The transform coefficients in the frequency domain are related to the pixel domain. Therefore, the DCT coefficients can be used for scene change detection in compressed video sequences.

Before discussing the indexing techniques, we provide a brief description of MPEG [13] algorithm. In MPEG (see Fig. 9), a block-based motion compensation scheme is employed to remove the temporal redundancy. Because of the conflicting requirements of random access and high compression ratio, the MPEG standard suggests that frames be divided in three categories: I, P and B frames. The organization of the three frame types in a sequence is very flexible. Fig. 10 illustrates the relationship among the three different frame types in a group of pictures (GOP). Intra coded frames (I-frames) are coded without reference to other frames and employ a coding scheme similar to JPEG baseline scheme. Predictive coded frames (P-frames) are coded more efficiently using motion compensated prediction from a

past I or P-frames and are generally used as a reference for further prediction. Bi-directionally predictive coded frames (B-frames) provide the highest degree of compression but require both the past and future reference frames for motion compensation. We note that, B-frames are never used as a reference for prediction.

Zhang *et al*. [80] have presented a pair-wise comparison technique for the intracoded (I-frame) where the corresponding DCT coefficients in the two frames f_m and f_n are matched. This is similar to the pixel intensity matching technique (see section 4) in the uncompressed domain. Here, the pair wise normalized absolute difference $D(f_m, f_n, l)$ of the l block in two frames f_m and f_n is determined using

$$D(f_m, f_n, l) = \frac{1}{64}\sum_{k=1}^{64}\frac{\left|c(f_m, l, k) - c(f_n, l, k)\right|}{\max(c(f_m, l, k), c(f_n, l, k))} \tag{3}$$

where $c(f_m, l, k)$ is the kth coefficient of block l in f_m. If the difference $D(f_m, f_n, l)$ is larger than a threshold, the block l is considered to be changed. If the number of changed blocks exceeds a certain threshold, a scene change is declared in the video sequence from frame f_m to frame f_n.

Arman *et al*. [81] have proposed a technique based on the correlation of corresponding DCT coefficients of two neighboring frames. For each compressed frame f_m, B blocks are first chosen *apriori* from R connected regions in f_m. A set of randomly distributed coefficients $\{c_x, c_y, c_z, ...\}$ is selected from each block where c_x is the x th coefficient. A vector $Vf_m = \{c_1, c_2, c_3, ...\}$ is formed by concatenating the sets of coefficients selected from the individual blocks in R. The vector Vf_m represents f_m in the transform domain. The normalized inner product is used as a metric to judge the similarity of frame f_m to frame f_n.

$$\gamma = 1 - \frac{Vf_m \bullet Vf_n}{\left|Vf_m\right|\left|Vf_n\right|} \tag{4}$$

A scene transition is detected if Ψ is greater than a threshold. In case of false positives, which result from camera and object motion, f_m and f_n are decompressed and their color histograms are compared to detect camera breaks. We note that, Zhang's technique [80] is computationally less intensive compared to Arman's technique, although the former is more sensitive to gradual changes.

We note that the previous two algorithms are applied on video sequences compressed using motion JPEG. In the case of MPEG video, only I-frames are compressed with DCT coefficients and hence the previous two techniques cannot be directly applied to the B- and P-frames. In addition, the techniques based on I-frames may result in false positives. To overcome these problems, Yeo *et al*. [82] have proposed a unified approach for scene change detection in motion JPEG and MPEG. This algorithm is based on the use of only the DC coefficients.

To start with, a DC frame f_m^{DC} is constructed for every frame in the sequence. The DC coefficients in JPEG and I-frames in MPEG are obtained directly from each block. For P and B-frames in MPEG video, the DC coefficients are estimated. The sum of the difference magnitude of the DC frames f_m^{DC} and f_n^{DC} is used a measure of similarity between two frames, *i.e.*,

$$D(f_m^{DC}, f_n^{DC}) = \sum_{i=1}^{X/8}\sum_{j=1}^{Y/8}\left|f_m^{DC}(i,j) - f_n^{DC}(i,j)\right| \tag{5}$$

where $f_m^{DC}(i,j)$ is the DC coefficient of block (i,j). A scene change from f_m to f_n is declared if: (i) $D(f_m^{DC}, f_n^{DC})$ is the maximum within a symmetric sliding window and (ii) $D(f_m^{DC}, f_n^{DC})$ is *2-3* times the second largest maximum in the window. Although this technique is fast, cuts may be misdetected between two frames which have similar pixel values, but different density functions. A metric for gradual transition has also been proposed [82] based on temporal

subsampling where one in every 20 frames is tested rather than successive frames. This technique is sensitive to camera flashes and variations in scene that typically occur before scene changes.

Vector Quantization

Idris *et al*. [83] have proposed a vector quantization technique for video indexing. This is basically an extension of image indexing technique discussed in section 4 to video indexing. We note that the histograms of the labels of a frame f_m is a K dimensional vector $\{H(f_m,i);\ i=1,2,\ldots K\}$ where $H(f_m,i)$ is the number of labels i in the compressed frame and K is the number of codewords in the codebook. The difference between two frames f_m to f_n is measured using the c^2 metric:

$$d(f_m,f_n)=\sum_{i=1}^{K}\frac{(H(f_m,i)-H(f_n,i))^2}{(H(f_m,i)+H(f_n,i))^2} \qquad (6)$$

A large value of $d(f_m,f_n)$ indicates that f_m and f_n belong to different scenes. An abrupt change is declared if the difference between two successive frames exceeds a threshold. A gradual transition transition is detected if the difference between the current frame and the first frame of the current shot is greater than a threshold.

Subband Decomposition

Lee *et al*. [79] have proposed a histogram based indexing technique in the subband domain. Here, the histograms of lowpass subbands of two frames are compared hierarchically from lower resolution to higher resolution. The authors have compared the performance of DOH, HOD, BVD, and BHD (see section 4) techniques in the multiresolution framework. They have employed twin comparison method to detect both abrupt and gradual changes. A block diagram of this technique, with two level subband decomposition, is shown in Fig. 11. Let the histogram of level 0, 1, and 2 of a subband decomposed frame f be h_0^f, h_1^f, and h_2^f, respectively. In the first pass, a transition between frame f and g is coarsely estimated by comparing h_2^f and h_2^g. The estimation can be further refined by comparing histograms of lower levels. We note that the size of the level-2 subband is one-sixteenth the size of the original image and hence, this technique has a low computational complexity.

Motion Vectors

Motion analysis is an important step in video processing. A video stream is composed of video elements constrained by the spatiotemporal piecewise continuity of visual cues. The normally coherent visual motion becomes suddenly discontinuous in the event of scene changes or new activities. Hence, motion discontinuities may be used to mark the change of a scene, the occurrence of occlusion, or the inception of a new activity.

The spatiotemporal (ST) surfaces can be used to represent the shape and motion of a moving planar object. Hence, it is possible to classify image motion qualitatively if the types of adjacent surfaces patches are known. Hsu *et al*. [84] have proposed techniques to characterize motion activities by considering the Gaussian mean and curvature of the spatiotemporal surfaces. Clustering and split-and-merge approach are then taken to segment the video.

Shahraray *et al*. [85] have proposed a technique based on motion-controlled temporal filtering of the disparity between consecutive frames to detect abrupt and gradual scene changes. A block matching process is performed for each block in the first image to find the *best* fitting region in the second image. A nonlinear statistical filter is then used to generate a global match value. Gradual transition is detected by identification of sustained low level increases in matched values.

In MPEG, B and P-frames contain the DCT coefficients of the error signal and motion vectors. Liu *et al*. [86] have presented a technique based on the error signal and the number of motion vectors. A scene cut between a current P frame f_n^P and the corresponding past reference frame f_n^R increases the error energy. Hence, the error energy provides a measure of similarity between f_n^P and the motion compensated frame f_n^R.

$$S_6(f_m^p, f_n^R) = \left(\sum_{i=1}^{F_p} E_i \right) / F_p \qquad (7)$$

where E_i is the error energy of macroblock i and F_p is the number of forward predicted macroblocks. For the detection of scene changes based on B-frames, the difference between the number of forward predicted macroblocks F_p and backward predicted B_p is used. A scene change between a B-frame and its past reference B-frame will decrease F_p and increase B_p. A scene change is declared if the difference between F_p and B_p changes from positive to negative.

Zhang *et al*. [80] have proposed a technique for cut detection using motion vectors in MPEG. This approach is based on the number of motion vectors M. In P-frames, M is the number of motion vectors. In B-frame, M is the smaller of the counts of the forward and backward non-zero motion. Then $M < T$ will be an effective indicator of a camera boundary before or after the B and P-frame, where T is a threshold value close to zero. However, this method yields false detection when there is no motion. This is improved by applying the normalized inner product metric to the two I-frames on the sides of the B-frame where a break has been detected.

Meng *et al*. [87] have presented a segmentation algorithm based on motion information and the DC coefficients of the luminance component. To start with, the DC coefficients in the P-frames are reconstructed. The variance of the DC coefficients $|\Delta\sigma^2|$ for the I and P-frames is then computed. Three ratios are computed, namely: i) R_p - the ratio of intracoded blocks and motion predicted blocks for P frames, ii) R_b - the ratio of backward and forward motion vectors, iii) R_f - the inverse of R_b. A two-pass algorithm is applied. In the first pass, suspected scene change frames are marked. A P-frame and B-frame are suspected frames if R_p and R_b, peaks, respectively. An I- frame is a suspected frame if $|\Delta\sigma^2|$ peaks and R_f of the B-frames in front of them peaks. In the second pass, all suspected frames that fall in a dissolve region are unmarked. All the marked frames are then examined. If the difference between the current marked frame and the last scene change exceeds a threshold, then the current marked frame is a true scene change.

VI. CONCLUSION

The demand for multimedia data services necessitates the development of techniques to store, navigate and retrieve visual data. The use of existing text indexing techniques for image and video indexing is inefficient and complex. Moreover, this approach is not generic, and hence is not useful in a wide variety of applications. Consequently, content-based indexing techniques should be employed to search for desired images and video in a database. This paper reviews and summarizes compressed domain indexing techniques proposed in the recent literature. The main contribution of each algorithm has been presented in brief. The main focus of the review is how the image feature vectors are generated using the transform coefficients, VQ labels, or fractal codes. The use of motion vectors in video indexing was also reviewed. In addition, we also discussed how the pixel domain techniques (e.g. texture, histogram, *etc*.) were imported in compressed domain and their relative complexity.

We note that the techniques reviewed in this paper are associated with coding techniques that were developed to provide high compression ratio. To obtain a superior overall performance, integrated coding and indexing techniques should be developed. The emerging second-generation image and video coding techniques are expected to provide a better joint coding and indexing performance. These techniques are generally based on segmentation or model based schemes. The authors feel that the feature extraction techniques reviewed in this paper will provide important clues to design efficient indexing techniques in the future standards.

REFERENCES

1. B. Furht, S. W. Smoliar and H. Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, 1995.
2. M. O'Docherty and C. Daskalakis, "Multimedia information systems: the management and semantic retrieval of all electronic datatypes," *The Computer Journal*, Vol. 34, No. 3, pp. 225-238, 1991.
3. Digital Libraries, Special issue of *Communications of the ACM*, Vol. 38, No. 4, April 1995.

4. M. Ehlers, G. Edwards and Y. Bedard, "Integration of remote sensing with geographic information systems: a necessary evolution," *Photogrammetric Engineering and Remote Sensing*, Vol. 55, No. 11, pp. 1619-1627, 1989.
5. T. D. C. Little, G. Ahanger, R. J. Folz, J. F. Gibbon, F. W. Reeve, D. H. Schelleng and D. Venkatesh, "A digital video-on-demand service supporting content-based queries," *Proc. of 1st International Conference on Multimedia*, pp. 427-436, Aug. 1993.
6. V. N. Gudivada and V. V. Raghavan, "Content based image retrieval systems," *IEEE Computer*, Vol. 28, No. 9, pp. 18-22, Sep 1995.
7. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by image and video content: the QBIC system," *IEEE Computer*, pp. 23-32, September 1995.
8. J. R. Bach, C. Fuller, A. Gupta, A. Hamapur, B. Horowitz, R. Humphrey, R. Jain and C. F. Shu, "The Virage image search engine: an open framework for image management," *Proc. of SPIE*, Vol. 2670, pp. 76-87, 1996.
9. J. K. Wu, A. D. Narasimhalu, B. M. Mehtre, C. P. Lam and Y. J. Gao, "CORE: a content-based retrieval engine for multimedia information systems," *Multimedia Systems*, Vol. 3, pp. 25-41, 1995.
10. F. Arman, A. Hsu and M.-Y. Chiu, "Image processing on compressed data for large video databases," *Proc. of ACM International Conference on Multimedia*, pp. 267-272, Anaheim, CA, USA, 1993.
11. R. C. Jain, S. N. J. Murthy and P. L. J. Chen, "Similarity measures for image databases," *SPIE Proceedings: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 58-65, Feb 1995.
12. G. K. Wallace, "The JPEG Still Picture Compression Standard," *Communication of the ACM*, Vol. 34, No. 4, pp. 31-45, April 1991.
13. D. L. Gall, "MPEG: A video compression standard for multimedia applications," *Communications of the ACM*, Vol. 34, No. 4, pp. 59-63, April 1991.
14. M. Liou, "Overview of the p×64 kbits/s video coding standard," *Communications of the ACM*, Vol. 34, No. 4, pp. 46-58, April 1991.
15. P. Aigrain, H. Zhang and D. Petkovic, "Content-based representation and retrieval of visual media: a state of the art review," *Multimedia Tools and Applications 3*, pp. 179-202, 1996.
16. F. Idris and S. Panchanathan, "Review of image and video indexing techniques," *Journal of Visual Communication and Image Representation*, June 1997.
17. G. Ahanger and T. D. C. Little, "Survey of technologies for parsing and indexing digital video," *Journal of Visual communications and Image Representation*, Vol. 7, No. 1, pp. 28-43, March 1996.
18. M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp. 11-32, 1991.
19. J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance function," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, pp. 729-736, July 1995.
20. M. Stricker and M. Orengo, "Similarity of Color Images", *Proc. of SPIE: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 381-392, February 1995.
21. A. Vellaikal and C.-C. J. Kuo, "Content-Based Retrieval of Color and Multispectral Images Using Joint Spatial-Spectral Indexing", *Proc. of SPIE: Digital Image Storage and Archiving Systems*, Vol. 2606, pp. 232-243, October 1995.
22. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englwood Cliffs, New Jersy, 1989.
23. M. Tuceryan and A. K. Jain, "Texture analysis," *Handbook of Pattern Recognition and Computer Vision*, Editors: C. H. Chen *et al*., pp. 235-276, World Scientific, 1993.
24. R. W. Picard and F. Liu, "A New Wold Ordering for Image Similarity", *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, Vol. V, pp. 129-132, April 1994.
25. H. Tamura and N. Nokoya, "Image Database Systems: A Survey", *Pattern Recognition*, Vol. 17, No. 1, pp. 29-43, 1984.
26. H. Zhang and D. Zhong, "A Scheme for Visual Feature Based Image Indexing", *Proc. of SPIE: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 36-46, Feb 1995.
27. A. R. Rao, N. Bhushan, and G. L. Lohse, "The relationship between texture terms and texture images: A study in human texture perception," *Proc. of SPIE: Storage and Retrieval for Still Image and Video Databases IV*, Vol. 2670, pp. 206-214, Feb 1996.

28. K. Hirata and T. Kato, "Rough sketch-based image information retrieval," *NEC Research and Development*, Vol. 34, No. 2, pp. 263-273, April 1993.
29. D. Tegolo, "Shape analysis for image retrieval," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases II*, Vol. 2185, pp. 59-69, Feb 1994.
30. J. P. Eakins, K. Shields, and J. Boardman, "ARTISAN - a shape retrieval system based on boundary family indexing", *Proc. of SPIE: Storage and Retrieval for Image and Video Databases IV*, Vol. 2670, pp. 17-28, Feb 1996.
31. S.-K. Chang, Q.-Y. Shi and C.-W. Yan, "Iconic Indexing by 2-D Strings", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 3, pp. 413-428, May 1987.
32. E. Jungert, "Extended Symbolic Projection as a Knowledge Structure for Image Database Systems", *4th BPRA Conference on Pattern Recognition*, pp. 343-351, March 1988.
33. S. K. Chang, C. M. Lee and C. R. Dow, "Two dimensional string matching algorithm for conceptual pictorial queries", *SPIE Proceedings: Image Storage and Retrieval Systems*, Vol. 1662, pp. 47-58, Feb 1992.
34. S.-Y. Lee and F.-J. Hsu, "Spatial Reasoning and Similarity Retrieval of Images using 2-D C-String Knowledge Representation", *Pattern Recognition*, Vol. 25, No. 3, pp. 305-318, 1992.
35. H. S. Stone, C. S. Li, "Image matching by means of intensity and texture matching in the Fourier domain," *Proc. of SPIE*, Vol. 2670, pp. 337-349, 1996.
36. M. Augusteijn, L E. Clemens and K. A. Shaw, "Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 33, No. 3, pp. 616-626, May 1995.
37. A. Celentano and V. D. Lecce, "A FFT based technique for image signature generation," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases V*, Vol. 3022, pp.457-466, Feb 1997.
38. A. Pentland, R. W. Picard and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases II*, Vol. 2185, pp. 34-47, Feb 1994.
39. D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 831-836, Aug 1996.
40. X. Tang and W. K. Stewart, "Texture classification using principle-component analysis techniques," *Proc. of SPIE*, Vol. 2315, pp. 22-35, 1994.
41. J. A. Saghri, A. G. Tescher and J. T. Reagan, "Practical transform coding of multispectral imagery," *IEEE Signal Processing Magazine*, Vol. 12, No. 1, pp. 33-43, Jan 1995.
42. J. R. Smith and S. F. Chang, "Transform features for texture classification and discrimination in large image databases," *Proc. of IEEE Intl. Conf. on Image Processing*, Vol. 3, pp. 407-411, 1994.
43. R. Reeves, K. Kubik and W. Osberger, "Texture characterization of compressed aerial images using DCT coefficients," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases V*, Vol. 3022, pp. 398-407, Feb 1997.
44. M. Shneier and M. A. Mottaleb, "Exploiting the JPEG compression scheme for image retrieval," *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 849-853, August 1996.
45. A. A. Abdel-Malek and J. E. Hershey, "Feature cueing in the discrete cosine domain," *Journal of Electronic Imaging*, Vol. 3, pp. 71-80, Jan 1994.
46. I. Ng, T. Tan and J. Kittler, "On local linear transform and Gabor filter representation of texture," *Proc. of the 11th IAPR Intl. Conf. on Pattern Recognition*, pp. 627-631, 1992.
47. B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," *Proc. of SPIE*, Vol. 2670, pp. 404-414, 1996.
48. S. G. Mallat, "A theory for multiresolution signal representation: the wavelet decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, July 1989.
49. M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, Vol. 1, No. 2, pp. 205-220, April 1992.
50. T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. on Image Processing*, Vol. 2, No. 4, pp. 429-441, Oct 1993.
51. J. L. Chen and A. Kundu, "Rotation and gray scale invariant texture identification using wavelet decomposition and hidden Markov model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, pp. 208-214, Feb 1994.

52. M. K. Mandal, T. Aboulnasr and S. Panchanathan, "Image indexing using moments and wavelets," *IEEE Trans. on Consumer Electronics*, Vol. 42, No. 3, pp. 557-565, Aug 1996.
53. K. A. Birney and T. R. Fischer, "On the modeling of DCT and subband image data for compression," *IEEE Trans. on Image Processing*, Vol. 4, No. 2, pp. 186-193, Feb 1995.
54. M. K. Mandal, S. Panchanathan, and T. Aboulnasr, "Image indexing using translation and scale-invariant moments and wavelets," *Proc. of SPIE: Storage and Retrieval for Image and Video Databases V*, Vol. 3022, pp. 380-389, Feb 1997.
55. B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 837-841, Aug 1996.
56. H. Wang and S. F. Chang, "Adaptive image matching in the subband domain," *Proc. of SPIE: VCIP*, Vol. 2727, pp. 885-896, 1996.
57. C. E. Jacobs, A. Finkelstein and D. H. Salesin, "Fast multiresolution image querying," *Proc. of ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, pp. 277-286, Los Angeles, Aug 1995.
58. J. Z. Wang, G. Wiederhold, O. Firschein and S. X. Wei, "Wavelet-based image indexing techniques with partial sketch retrieval capability," *Proc. of the Forum on Research and Technology Advances in Digital Libraries*, pp. 13-24, Washington DC, May 1997.
59. F. Qi, D. Shen and L. Quan, "Wavelet transform based rotation invariant feature extraction in object recognition," *Proc. of Intl. Symp. on Information Theory & its Applications*, pp. 221-224, Nov 1994.
60. P. Rashkovskiy and L. Sadovnik, "Scale, rotation and shift invariant wavelet transform," *Proc. of SPIE : Optical Pattern Recognition V*, Vol. 2237, pp. 390-401, 1994.
61. J. Froment and S. Mallat, "Second generation compact image coding with wavelets," in *Wavelets: A Tutorial in Theory and Applications*, Ed: C. K. Chui, Academic Press, Inc., 1992.
62. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1991.
63. F. Idris and S. Panchanathan, "Image indexing using vector quantization," *SPIE Proceedings: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 373-380, Feb 1995.
64. F. Idris and S. Panchanathan, "Storage and retrieval of compressed images," *IEEE Trans. on Consumer Electronics*, Vol. 41, pp. 937-941, Aug 1995.
65. J. S. Barbas and S. I. Wolk, "Efficient organization of large ship radar databases using wavelets and structured vector quantization," *Proc. of Asilomer Conference on Signals, Systems and Computers*, Vol. 1, pp. 491-498, 1993.
66. A. Vellaikal, C. C. J. Kuo and S. Dao, "Content-based retrieval of remote-sensed images using vector quantization," *Proc. of SPIE*, Vol. 2488, pp. 178-189, 1995.
67. M. F. Barnsley and L. P. Hurd, *Fractal Image Compression*, Wellesley, MA: AK Peters Ltd., 1993.
68. A. Zhang, B. Cheng and R. S. Acharya, "Approach to query-by-texture in image database systems," *Proc. of SPIE: Digital Image Storage and Archiving Systems*, Vol. 2606, pp. 338-349, 1995.
69. A. Zhang, B. Cheng, R. S. Acharya, R. P. Menon, "Comparison of wavelet transforms and fractal coding in texture-based image retrieval," *Proc. of SPIE:Visual Data Exploration and Analysis III*, Vol. 2656, pp. 116-125, 1996.
70. T. Ida and Y. Sambonsugi, "Image segmentation using fractal coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 5, No. 6, pp. 567-570, Dec 1995.
71. J. Li, "Hybrid wavelet-fractal image compression based on a rate-distortion criterion," *SPIE Proceedings: Visual Communications and Image Processing*, Vol. 3024, pp. 1014-1025, Feb 1997.
72. F. Idris and S. Panchanathan, "Image indexing using wavelet vector quantization," *SPIE Proceedings: Digital Image Storage Archiving Systems*, Vol. 2606, pp. 269-275, Oct 1995.
73. M. D. Swanson, S. Hosur and A. H. Tewfik, "Image coding for content-based retrieval," *Proc. of SPIE: VCIP*, Vol. 2727, pp. 4-15, 1996.
74. C. Podilchuk and X. Zhang, "Face recognition using DCT-based feature vectors," *Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 2144-2147, 1996.
75. A. Akutsu, Y. Tonomura, H. Hashimoto and Y. Ohba, "Video indexing using motion vectors," *Proc. of SPIE: VCIP*, pp. 1522-1530, 1992.
76. R. Kasturi and R. Jain, "Dynamic Vision," *Computer Vision: Principles*, Eds. R. Kasturi, R. Jain., pp. 469-480, IEEE Computer Society Press, Washington, 1991.

77. H. J. Zhang, A. Kankanhalli and S. W. Smoliar and S. Y. Tan, "Automatic Partitioning of Full Motion Video," *Multimedia Systems*, Vol. 1, pp. 10-28, 1993.
78. A. Nagasaka and Y. Tanaka, "Automatic Video Indexing and Full-Video Search for Object Appearance," *IFIP: Visual Database Systems II*, pp. 113-127, 1992.
79. J. Lee and B. W. Dickinson, "Multiresolution video indexing for subband coded video databases," *SPIE Proceedings: Image and Video Processing II*, Vol. 2185, pp. 162-173, 1994.
80. H. Zhang, C. Y. Low and S. W. Smoliar, "Video Parsing and Browsing Using Compressed Data," *Multimedia Tools and Applications*, Vol. 1, No. 1, pp. 89-111, 1995.
81. F. Arman, A. Hsu, and M. Y. Chiu, "Image processing on compressed data for large video databases," *Proc. of ACM Multimedia*, pp. 267-272, 1993.
82. B. L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 5, No. 6, pp. 533-544, Dec 1995.
83. F. Idris and S. Panchanathan, "Indexing of compressed video sequences," *Proc. of SPIE*, Vol. 2670, pp. 247-253, 1996.
84. P. R. Hsu and H. Harashima, "Detecting scene changes and activities in video databases," *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Vol. 5, pp. 33-36, 1994.
85. B. Shahraray, "Scene change detection and content based sampling of video sequences," *Proc. of SPIE: Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 2-13, Feb 1995.
86. H. C. H. Liu and G. L. Zick, "Scene decomposition of MPEG compressed video," *Proc. of SPIE: Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 26-37, Feb 1995.
87. J. Meng, Y. Juan and S. F. Chang, "Scene change detection in a MPEG compressed video sequence," *Proc. of SPIE: Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 14-25, Feb 1995.

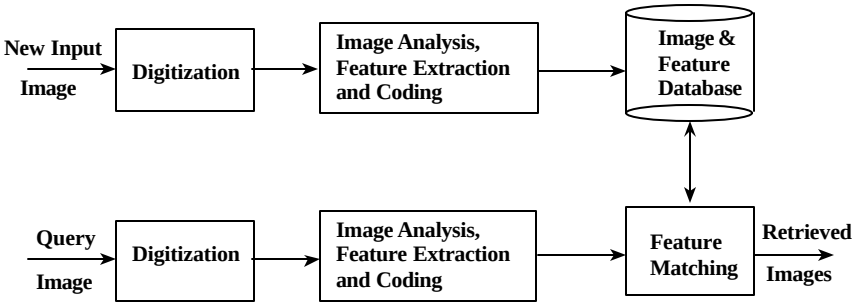


Figure 1: Schematic of an image archival and retrieval system [52]

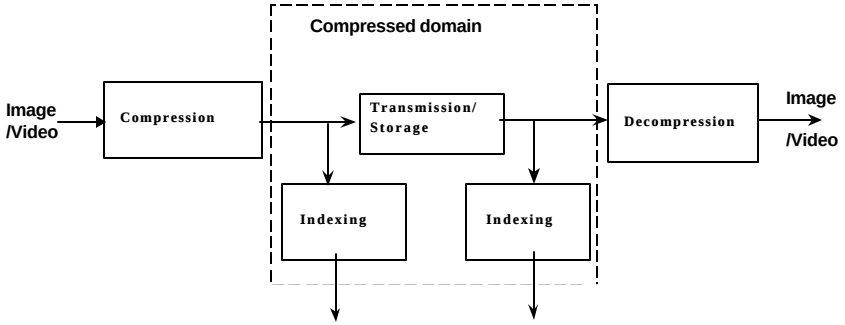
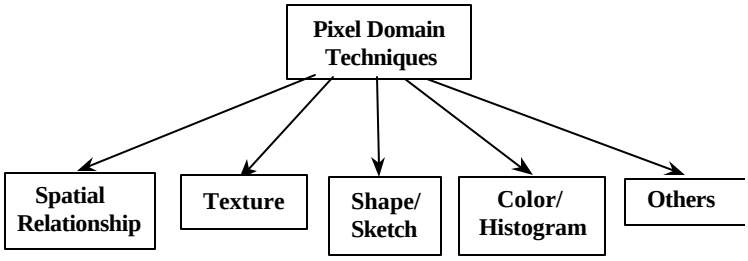
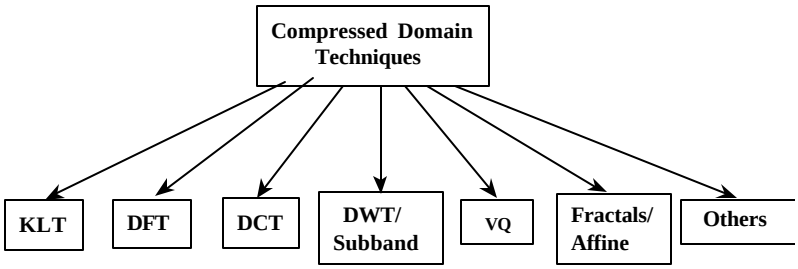


Figure 2: Block Diagram of a Compressed Domain Indexing System



(a)



(b)

Figure 3: Various Methods in Content Based Image Indexing: a) Pixel domain and b) Compressed domain

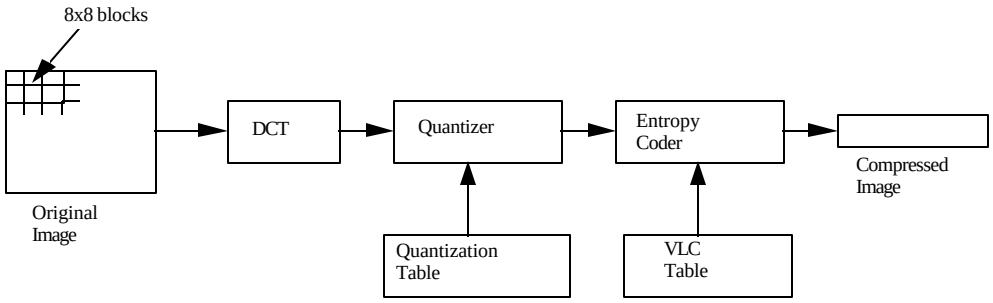


Figure 4: Baseline JPEG encoder

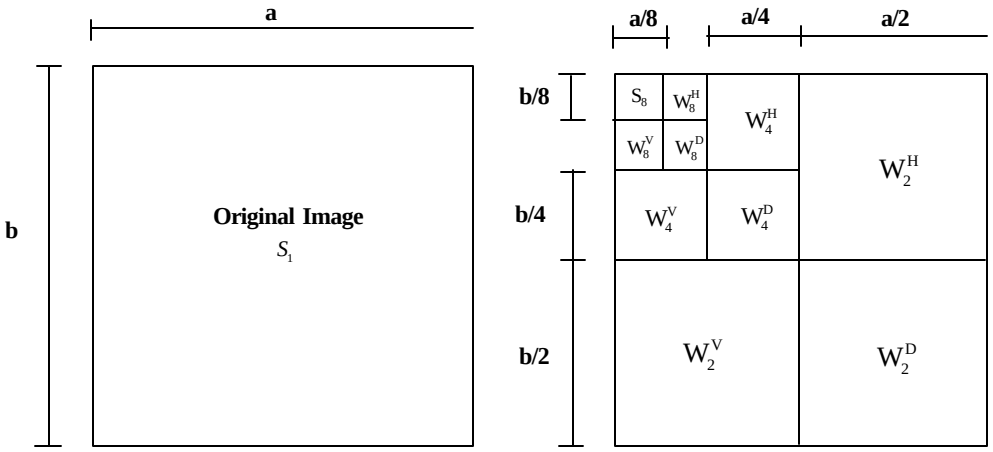


Figure 5: Wavelet transformed image

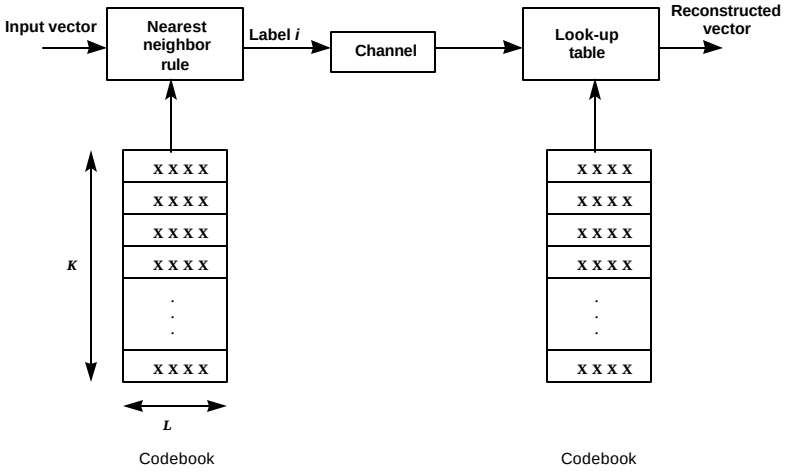


Figure 6: Block diagram of a vector quantization technique [63]

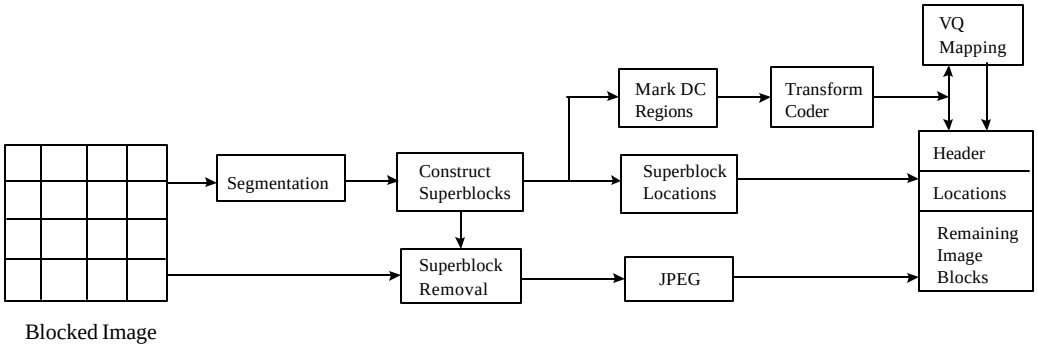


Figure 7: Block diagram of a hybrid image coding and indexing system [73]

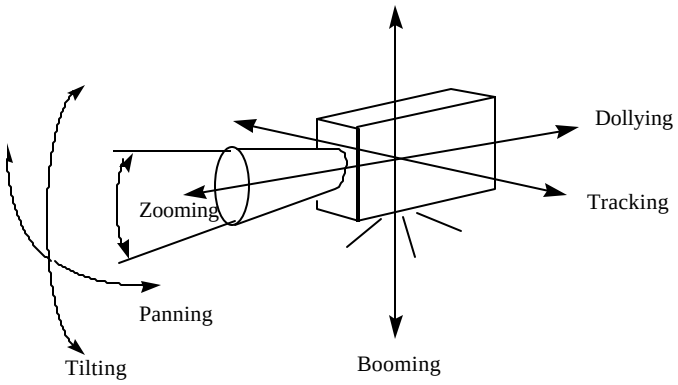


Figure 8: Basic camera operations [75]

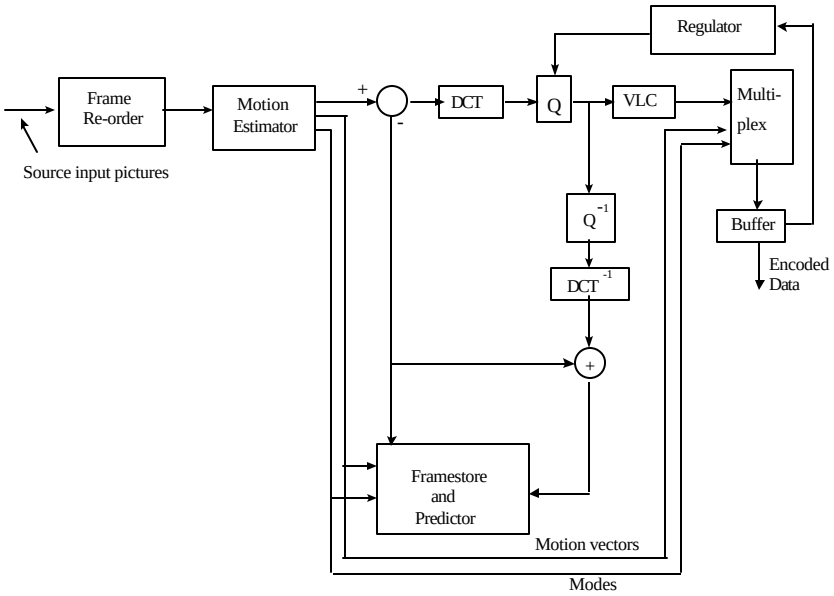


Figure 9: Block diagram of MPEG video encoder [13]

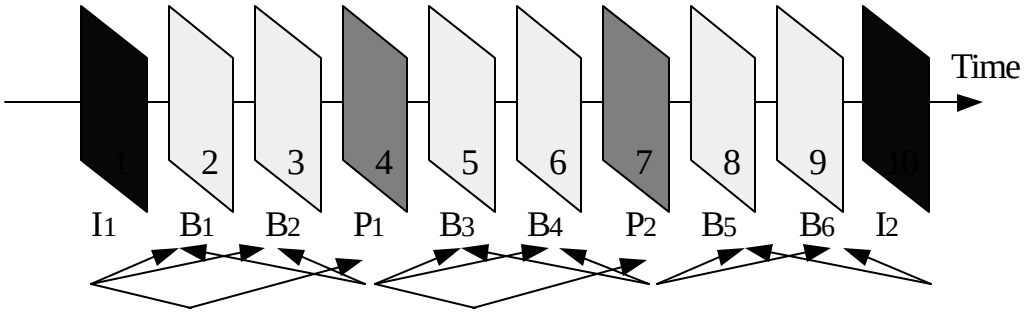


Figure 10: Example of a group of pictures (GOP) used in MPEG

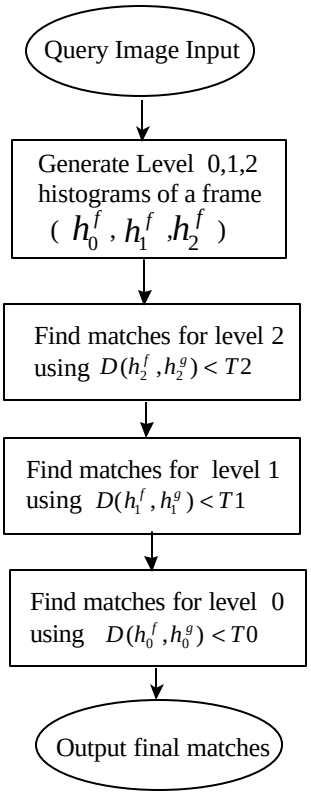


Figure 11: Flowchart of Multiresolution Video Segmentation [79]