Image and Video Compression: A Survey

Roger J. Clarke

Department of Computing and Electrical Engineering, Heriot-Watt University, Riccarton, Edinburgh EH14 4 AS, Scotland

ABSTRACT: At the present time, we stand upon the threshold of a revolution in the means available to us for the widespread dissemination of information in visual form through the rapidly increasing use of international standards for image and video compression. Yet, such standards, as observed by the casual user, are only the tip of the coding iceberg. Something like half a century of scientific and technological development has contributed to a vast body of knowledge concerning techniques for coding still and moving pictures, and the present article presents a survey of developments which have taken place since the first (predictive) coding algorithms were implemented in the 1950s. Initially, we briefly review the characteristics of the human eye which influence how we approach the design of coding algorithms; then we examine the still picture techniques of major interest—predictive and transform coding, vector quantization, and subband and wavelet multiresolution approaches. Recognizing that other forms of algorithm have also been of interest during this period, we next consider such techniques as quadtree decomposition and segmentation before looking at the problems arising from the presence of motion and its compensation in the coding of video signals. In the next section, various approaches to the coding of image sequences are reviewed, and we concentrate upon the now universally used hybrid motion compensated transform algorithm before examining more advanced techniques such as model and object-based coding. Of course, the key to widespread acceptance of any technological development is the establishment of standards, and all major proposals—JPEG, MPEG-I, -II, and -IV, H.261, and H.263, are considered with emphasis on the way in which the coding algorithm is implemented rather than on protocol and syntax considerations. Finally, comments are offered in respect of the future viability of coding standards, of less well-researched algorithms, and the overall position of image and video compression techniques in the rapidly developing field of visual information provision. © 1999 John Wiley & Sons, Inc. Int J Imaging Syst Technol, 10, 20–32, 1999

I. INTRODUCTION

As this century draws to a close, it is virtually impossible to imagine what life must have been like for the average man in the street 100 years ago. No cars meant either cycling to work or going by rail or on foot, no aircraft left international journeys to be undertaken by sea, no domestic appliances meant that almost all jobs around the house had to be done manually, and no communication meant, well, no communication. Telephone systems were in their infancy, there were no broadcast or television services, and this left newspapers as the only widespread information provider (and only one way at that). Person-to-person contact was carried out either face-to-face or by letter. Probably the greatest influence (although there are many contenders) on changes in social attitudes has been our growing ability to engage in almost instant broadcast and person-to-person communication over ever greater distances (in some cases it might even be argued that there now exists more communication provision than we need or is, strictly speaking, good for us). Undoubtedly, this change has been fostered by the widespread supplanting of analogue by digital technology over the past 3 decades or so (although, paradoxically, the final link in the chain, the telephone line, radio link, or whatever, may still well be analogue in nature), for this has allowed us to do three things much more easily than before: (a) carry out signal-processing operations very rapidly; (b) build very complex large-scale systems; and, most important, (c) store data easily. Where would telecommunications technology be, for example, if it were still as difficult to store information as it was, say, 50 years ago?

So where does image coding fit into all this? Throughout history, pictures have always had a high-profile role to play in communication. In the days when the majority of people could not read whatever written words were available to them anyway, images allowed an immediacy of impact and directness of contact achievable in no other way. Later, representation of moving pictures, as provided by the film and, subsequently, television, enhanced this capability to an enormous degree. It is impossible to appreciate the impact of moving color picture presentation (something we take for granted) on anyone who has not grown up with the idea. In the case of television, however, it was quickly realized that communicating a video image was vastly more expensive in terms of necessary channel capacity than was speech or music transmission; and even from early days, methods of reducing this requirement were sought. Given the constraints operating at the time, the development of interlace, for example, has to be seen as an elegant practical solution to the problem (despite the drawbacks it presents today for digital processing of conventional video material). Again, early studies showed that the video signal could advantageously be split into different frequency bands and these sent separately and yet needing, overall, less capacity than the original signal. With the development of digital services, a growing problem soon appeared in that it was no longer a matter of dealing with a few highly specialized application areas, but rather a wide-ranging spectrum encompassing person-to-person (videophone) communication at one end through videoconference and conventional television to high-definition television at the other. Of course, supply and demand are inextricably linked in this field, as elsewhere. Given the capability of sending images and video efficiently, more people want to make use of the facility, and at the same time, they generate new application areas, and so on. Widespread use over many different fields of application

 CCC 0899–9457/99/010020-13

also depends upon the development of, and agreement on, standards, which is the point at which we stand at the moment. In a broad sense, we can now assume that the first phase of the development of image coding has been completed. Established, well-regarded techniques are in place in several standards for still picture, video communication, broadcast use, etc., and there is some uncertainty about where we should be going next. There are also other well-researched and efficient techniques which did not make it into the standards, but which nevertheless may still be of use in particular areas of application. On the other hand, it is hard to escape the realization that, notwithstanding past successes, some fresh ideas are now necessary to impart a new direction to coding development and also free us from some of the problems which still bedevil what we have come to call conventional approaches. Again, the enormous bandwidth offered to us by optic fiber may remove the need for image-coding algorithms to be used in fixed service applications, leaving mobile systems as the major user of such techniques in an effort to conserve finite, and thus precious, spectrum space. There are thus many factors, not all of a purely technical nature, which will determine the future course of image-coding research.

In this article, we initially review the status of well-researched techniques, both those now incorporated into standards and others of similar standing, and then consider how the standards work of the last 10 years has employed such algorithms for the processing of a wide variety of image data. Thereafter, we consider the possibilities for new directions in image coding.

II. THE HUMAN VISUAL RESPONSE

It is appropriate to commence a survey of image-coding techniques with a brief review of what is usually, after all, the end user, so to speak, of the system output—the human visual system (HVS). It has to be said that, although the overall properties of the HVS naturally influence the possible degree of data reduction, it is unfortunate that over the years, more has not been done to establish in a detailed sense what is exactly necessary to present the eye with just that degree of information which it needs and no more. One reason for this is, of course, that the response of the eye contains a subjective element which it is notoriously difficult to quantify, and this as an area where much more work would be welcome. As far as we are concerned at present, the major characteristics of the HVS are (a) its response in the frequency domain, (b) its response to luminance amplitude variations and (c) the effect of object motion (Clarke, 1995). The (spatial) frequency domain response is characterized by a peak at a frequency of about 4–6 cycles per degree (as subtended by a sinusoidal luminance amplitude pattern presented to the eye by the display). This has two consequences. First, we can use the lower sensitivity of the eye at high spatial frequencies to be less careful about how we process frequency components in that region—reducing the bit allocation appropriately and, correspondingly, taking more care over those components which occur at or near the peak frequency. Modification of the algorithm can be done fairly easily for frequency domain coding approaches. The second consequence, however, is inseparable from the first and brings problems in its wake. Since the peak in response corresponds to a certain spatial frequency band in terms of the angle subtended at the eye's location, it follows that this relation must be maintained for the effect to work. Thus, we cannot observe the screen from any arbitrary distance that we choose—for every set of coding parameters and screen size, there is a correct viewing distance for maximum benefit to be obtained, but this constraint is normally totally ignored in practice. As far as amplitude variations are concerned, the logarithmic nature of the HVS response means that, where appropriate, we can use nonuniform quantization to allow more (absolute) error for large amplitude signals (in predictive coding, for example), and it is also possible to make use of the masking effect of the response—that a large luminance discontinuity reduces the visibility of nearby small degradations, in quantizer design. Making use of the temporal response of the HVS in the coding of image sequences is more difficult, not least because object size, contrast, etc., are involved in a non-separable way with the perception of motion. Basically, the response is similar to that in the frequency domain—a peak at a frequency of a few Hertz (temporal frequency) followed by a significant fall-off at higher frequencies, but the overall effect of this characteristic is considerably complicated by the fact that the eye tends to track objects of interest as they move, and thus tries to hold them stationary in the field of view (Girod, 1992). It remains a fact that, after many years, we still do not have a firm grasp on how HVS characteristics can really be used best to produce improvements in coding efficiency, not to mention how larger-scale subjective effects of the viewer/display relationship—fatigue or long-term changes in image quality, for example—influence quality judgments.

Figure 1. The first image frame from a typical sequence ("Claire") used for the development of compression algorithms.

III. BASIC CODING TECHNIQUES

In this and the next section, basic algorithms for coding still images (intraframe coding) are described. Typically, the image will be around 512 × 512 in extent, quantized to 8-bit (256 levels) amplitude resolution and either monochrome or color. In the latter case, it is usual to move from a three-color plane (RGB) representation to a television-style YUV or YIQ formulation. It is found that the additional load on the coding system due to the color information is relatively small—the luminance signal carries all of the detail resolution information and the U and V or I and Q data can be subsampled and coded using the same algorithm as the luminance term, but only needing something like 20% of the rate. Figure 1 shows the first frame of the sequence "CLAIRE": 256 × 256 in extent with 8-bit amplitude resolution.

A. Predictive Coding. It is usually the case that useful images comprise recognizable objects—cars, buildings, people, etc.—portrayed against similarly recognizable areas of reasonable extent.

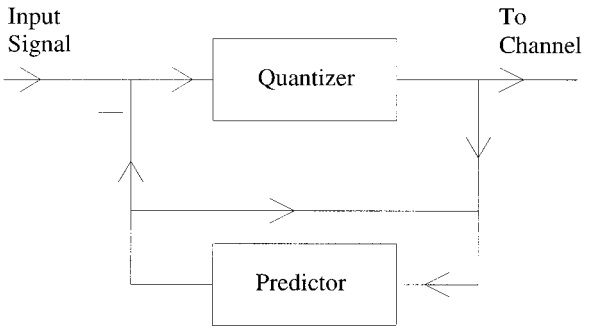

Figure 2. Predictive coding.

Only for regions of high detail or texture is this unlikely to be true, and these tend to occur over only a small part of the image or video sequence. This being so, we can make use of the high degree of correlation existing within the picture to reduce the number of bits necessary for reproduction, and the algorithm which does this in the most obvious fashion is predictive coding. Deriving from work on general signal prediction in the 1940s, this algorithm was the first image-coding technique to be extensively researched starting in the 1950s, and in its basic form it still finds place (albeit in the time dimension) in the standards algorithms of today. It is very simple and capable of reasonable rate reduction with excellent quality when applied to still pictures (O'Neal. 1996; Musmann, 1979). From the values of previously scanned picture elements at both transmitter and receiver, predictions are made of the next sample which the coder is to process. This prediction is subtracted from the actual value of the next sample and the error signal (difference) quantized and transmitted, to be added to the prediction made at the decoder to form the reconstructed signal. The procedure is shown in Figure 2. For a good prediction the error will be small, but just as important is the fact that its probability distribution is very well behaved compared with that of the original picture, and of Laplacian or even Gamma distribution form—very highly peaked around zero with relatively few large values, which latter will be a consequence of the edge detail in the picture. Coding this signal with maybe three bits per element and using nonuniform quantization and variable word-length coding allows good quality to be achieved for images containing television-type detail (Musmann et al., 1985). Of course many refinements are possible; both predictor and quantizer may be made adaptive to account for localized detail (needing transmission of overhead information to the receiver), possibly through the use of a least mean squares updating algorithm (Alexander and Rajala, 1984) or, more recently, through the use of higher-order statistics (Tekalp et al., 1990). Alternatively, neural approaches may be used to optimize predictor structure (Dony and Haykin, 1995). Setting the basic element-by-element algorithm in a block context in this way and maybe using arithmetic coding for the output signal allows rates down to around one bit per element to be achieved.

B. Block Transform Coding. Unfortunately, despite the many advantages of predictive coding, it is not capable of achieving the very low transmission rates demanded by some of today's applications. For this, a move away from single-element processing is necessary and we must allocate bits to blocks instead. The predominant example of this type of approach is transform coding, by far the most widely researched technique for image coding over the past 30 years (Clarke, 1985). In this case, operation is in a frequency-like domain (true frequency should the Fourier transform be used), and use is made of the poor high-frequency response of the eye to quantize coarsely, or even delete entirely, small high-order terms to achieve data rate reduction. Since small (8×8 or 16×16) image blocks are two-dimensionally transformed in this algorithm, undesirable windowing artifacts can occur with the Fourier transform, and the optimum, data-independent transform now universally used for this purpose is the discrete cosine transform (DCT), which, as a result of its formulation, reduces such problems (Ahmed et al., 1974). It is worth pointing out here that the crucial principle in the operation of transform coding is just the same as that of predictive coding, and this is the strong change in the distribution of the signal produced by the transform. In predictive coding most error terms are small and only a few are large, and the resulting distribution can be efficiently quantized with many fewer levels than those in the original data. In a similar way, for data with reasonable interelement correlation, the frequency spectrum will be strongly low-pass, and this is what allows us to concentrate coding capacity on the few important lower-order terms. In addition, these terms themselves have sharply peaked amplitude distributions, and this brings added benefit when they are quantized and variable length coded. In the absence of quantization, the operation is itself totally reversible, and it is the change in the distribution of the signal mentioned above (which might be looked on as a preprocessing first stage) which is significant in allowing efficient quantization. At the decoder the quantized (i.e., approximate) coefficient set, with terms deleted at the coder being replaced by zeroes, is inverse transformed to generate the output picture. For the transform operation to achieve useful efficiency, it was soon realized that it would be necessary to make it adaptive. A mass of experimentation exhaustively reported in the literature led to classified adaptive transform coding (Chen and Smith, 1977) in which image blocks are sorted into four categories according to activity of detail, maybe using block variance as a criterion, and then transformed, the coefficients scaled, and then quantized with a minimum mean square error quantizer. This scheme can easily deal with color information and represents the state which transform coding had reached by the late 1970s, with data rates running at maybe 0.75–1.0 bit/element. In the early 1980s, however, a modified design with significant advantages was reported (Chen and Pratt, 1984). Here, all transform coefficients apart from the zero-order term (often called the DC term, this coefficient is responsible for the reproduction of average block luminance and so needs to be retained accurately to avoid otherwise easily visible luminance discontinuities from block to block) are thresholded, scaled, and uniformly quantized. These 'AC' terms are then Huffman coded in amplitude and location along a zigzag path running across the coefficient array from top left (DC term) to bottom right (the highest frequencies). In this way, operation at around 0.5 bit/element with good quality is possible. Transform coding carried out in this way is the basis of the JPEG still picture coding standard (Clarke, 1995) (see below). It is worth mentioning here that a major drawback of the block transform scheme at very low data rates (below 0.5 bit/element) is the likely appearance of block-structured artifacts in the reproduced picture. To demonstrate this, Figure 3 shows the image of Figure 1 transform coded at 0.2 bit/element. Typical artifacts are present: The regular 8×8 block structure is very visible and annoying, and, as always with frequency domain techniques, when coding bits are scarce, high-frequency detail suffers and fuzzy object edges result. Several methods of mitigating the visibility of block structure by smoothing the block-to-block discon-

Figure 3. The image of Figure 1 transform coded at 0.2 bit/element.

tinuity have been reported, one of the most significant being the idea of overlapping the blocks to be transformed (Malvar, 1992).

C. Vector Quantization. Transform coding makes use of the interelement correlation within a picture to concentrate coding capacity on the dominant low-frequency terms produced in the corresponding low-'frequency' block representation. Another technique which makes use of similarities within the data, albeit at a higher level, is vector quantization (Gray, 1984). In scalar quantization, data amplitudes are reconstructed as the nearest predetermined value to that actually occurring within any given (one-dimensional) space between two adjacent decision levels. Given the usual level of interelement correlation within pictures, it is evident that we could jointly quantize pairs of neighboring picture elements as points in a two-dimensional decision space and, apart from quantization inaccuracies, gain efficiency by so doing; pairs of similar values are much more likely to occur than those which are widely different (these latter representing edges). Vector quantization extends this idea to a larger region (a usual approach is to take a 4×4 picture block, considered as a vector of length 16) to produce results which rival transform coding in terms of the quality of reproduction at rates around 0.5 bit/element, particularly if adaptive schemes are used [Panchanathan and Goldberg, 1991). In vector quantization, use is made of the fact that many picture blocks will be very similar (background, interiors of large objects, etc.) in terms of luminance, color, and so on, or maybe contain strong detail of the same orientation. Such blocks will all be displayed as the same representative block chosen from a codebook of typical blocks (vectors) via some appropriate distance measure. The system has the advantage for certain applications that all the processing power is required at the transmitter, the receiver/decoder being trivially simple—one transmitted index word per block is all that is needed to access the codebook (look-up table) entry for display, maybe with some simple scaling operation. As a simple example, if we have 1024 representative entries, a 10-bit index is needed. If this identifies a 4×4 block, then the data rate is about 2/3 bit/element. In practice, some further sophistication is needed in the algorithm to cope with the artifacts which would be produced by such a basic scheme. There are many methods of generating the codebook, one tried and tested example of which was reported in 1980 (Linde et al., 1980). Given a first try reproduction codebook, all vectors from a suitable training sequence are allocated the closest entry according to some distance measure (mean square, or mean absolute, energy of the error vector, for example). Optimization proceeds by determining the new best code word for each of the partitions of training vectors so produced and then iterating the process. An initial codebook may be produced by first finding the one optimum code word for the whole of the training sequence, splitting it into two close but different vectors, and then proceeding as above. This codebook generation process is intensive both in time and computation, as is the other basic operation needed for coding an input vector: full search of the codebook for the nearest reproduction vector to determine the appropriate index to be transmitted. Most research on vector quantization since its introduction for image coding in the early 1980s has concentrated on these two problems, and a multiplicity of methods is now available for their (partial) solution: applying the splitting process preferentially to those nodes giving greatest decrease in distortion for the smallest increase overall in the number of codebook entries, and maybe using multiple splits as well. Separating out block mean value and standard deviation (corresponding to activity level) for separate transmission can also be helpful (Murakami et al., 1982) (Figure 4), as can classification of codebooks according to the presence of strong directional detail (Gersho and Ramamurthi, 1982). Neural optimization techniques can also be employed (Dony and Haykin, 1995; Lee and Petersen, 1990). Likewise, fast search methods have been intensively researched, with all manner of partial, tree, and approximate searches contributing to the speed-up of the process. It is also possible to use a regular lattice structure for the codebook (Chen, 1984). This has the advantage that no actual codebook need be stored and processing is very rapid—especially beneficial in video applications (see later). Over the years, vector quantization has evolved into an efficient coding technique which may either be used on its own or as a postprocessing step to code the output of a previous algorithm—arrays of transform coefficients, for example.

D. Subband and Wavelet Coding. One of the earliest approaches suggested for the reduction in bandwidth or channel capacity for the transmission of image data was frequency division, in which the total bandwidth is split, at the simplest level, into low- and high-frequency segments. The low-frequency band has reduced resolution and so needs far fewer bits for transmission; the upper-frequency band will generally have few significant components and can likewise be easily coded. Over the past 10 years or so, this basic

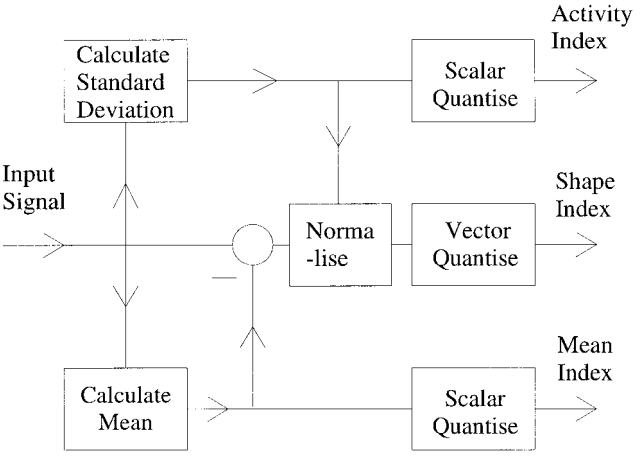

Figure 4. Normalized vector quantization.

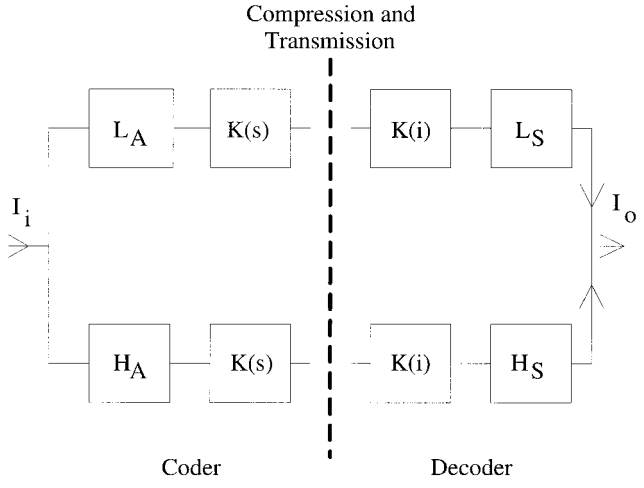

Figure 5. Basic filter structure for subband coding. I_i = input signal; I_o = output signal; L_A and H_A = low- and high-pass analysis filters; L_S and H_S = low- and high-pass synthesis filters; K(s) = subsampling at the coder; K(i) = interpolation at the decoder.

idea has developed into a powerful and flexible image coding technique: subband coding (Woods, 1991). Here, the image spectrum is split basically into two components as above, but now both horizontally and vertically, which are then subsampled by the same factor to give four two-dimensional frequency bands containing combinations such as low horizontal and low vertical frequencies, low horizontal and high vertical frequencies, and so on, the total number of samples being unchanged. The basic one-dimensional scheme is shown in Figure 5. As with predictive and transform coding, no data reduction is achieved by the first part of the algorithm, and it is the efficient coding of the various subbands using predictive coding, or vector quantization, for example, which is responsible for this process. Such latter algorithms can be much more efficiently matched to the individual subband characteristics than to the image as a whole, and this allows good results in the 0.5 bit/element region to be achieved (Lookabaugh and Perkins, 1990). Usually, the band split will be more extensive than the simple 2×2 split described above. The reapplication of this step to the outputs of the initial split will result in a 16-band (4×4) structure. Alternatively, it may be better to split lower-frequency bands more finely still and leave the highest horizontal and vertical frequency band unsplit—it will rarely contain any detail of real significance. It might be noted here that there are close connections between subband and transform coding. Sequential multilevel filtering using a simple high/low split of the kind mentioned above can be shown to produce the same result as a transform operator; indeed, transform coding may be considered to be a form of subband coding in which each subband contains only one coefficient.

Lately there has been much interest in the analysis of signals whose properties change with time. For this purpose, the Fourier transform is not suitable, since its baseline (in theory at least) is infinite, and the necessary window function added for practical implementation always introduces undesirable artifacts into the analysis. Out of this interest has come intensive research into wavelets (Chui, 1992). Such functions have restricted support and can allow flexible and efficient time (or space)/frequency processing, with good resolution at low frequencies via long windows, and good spatial resolution at high frequencies obtained by using short, wide-bandwidth ones. As far as image coding is concerned, wavelet processing is not unlike subband coding in basic outline, save that it is usually carried out in a multiresolution context. Here, an initial 2×2 split and subsample operation is carried out as previously described, following which only the lowest-frequency subband is resplit. At each level, this results in one low-resolution image and three so-called detail images. We can thus generate a hierarchy of images at various levels of resolution which will allow image reconstruction for a wide variety of applications from the one data stream. The smallest and lowest-resolution image might be used as the initial output of a database search, for example. If higher resolution is needed, this signal is upsampled (interpolated) and the three detail signals at the next level added, and so on, until the final full-resolution version is obtained. As usual, predictive coding or vector quantization can be used to code the various individual bands involved, and the scheme may be significantly improved by the inclusion of tree-structured algorithms for tracking significant coefficients through the various wavelet levels (Schapiro, 1993; Said and Pearlman, 1996).

Wavelet coding represents a sophisticated version of multiresolution decomposition whereby one algorithm may have a variety of image qualities at the output. It is appropriate to mention here the original impetus for the idea, which has been with us for some 20 years now. In this realization, lower-resolution images were produced from higher-level ones by Gaussian filtering and subsampling and then expanded (interpolated) again to be used as a prediction for the upper-level image. The prediction error then corresponds to the detail signal and, since it has a Laplacian probability distribution, and, moreover, appears as a set of image signal levels, one above the other, the idea of the Laplacian pyramid emerged (Burt and Adelson, 1983). Multiresolution decomposition is of great significance, given the variety and scope of present-day digital image services, and the wavelet approach is more elegant and flexible than many other methods (using the DCT, for example) which have been proposed for this purpose.

IV. OTHER TECHNIQUES

Broadly speaking, the techniques described above form the mainstream approach to still image coding. Over the years, however, a multiplicity of other approaches has been developed, some of which, at least, hold continuing interest for the advancement of the subject. These seem to be mainly spatial domain algorithms—maybe the frequency domain has now yielded up all its secrets in this connection? One technique which has appeared and reappeared in a number of different guises is that of quadtree image description (Samet, 1984). In its basic form, the quadtree is an image decomposition operation which, for coding applications, has the merits of strong adaptivity and simple block addressing. Starting with large image blocks (32×32, for example), some test of uniformity (luminance, color, or texture, for example) is made. If passed, the whole block is coded as such; if not, a subdivision is carried out and the process is repeated, down maybe to the level of 2×2 blocks (which latter might then be simply vector quantized). An alternative approach is bottom-up coding by successive merging of larger and larger blocks in a similar way (Strobach, 1991). Instead of searching for some simple uniformity property over successive blocks, the sophistication of this step may be increased to testing against an interpolation made from the four corner points of any block, or even against a simple polynomial fit, at each step bearing in mind the need for the totality of data to be transmitted (corner points, polynomial coefficients, etc.) to be significantly less than that which would otherwise

be necessary for simple element-by-element block representation. Again, quadtree decomposition may advantageously be combined with transform coding or vector quantization with overall benefit.

For the first half of the past decade, a remarkable degree of comment and speculation surrounded the suggestion made in 1988 that the use of so-called fractals was "a better way to compress images" (Barnsley and Sloan, 1988). Now that the debate has died down, this approach emerges as one having a performance roughly equivalent to that of the older techniques already discussed; indeed, it is best viewed as a relative of vector quantization, though without an explicitly defined codebook (Jacquin, 1992). Pairs of blocks are sought within the image from one of which, via an affine transform (rotation, translation, and scale change), the other may be derived. The coefficients of the transform then constitute the coded signal from which the output image may be iteratively reconstructed. The self-similarity upon which the method depends can exist within a single image frame or between the various image planes of the kind of multiresolution structure described previously.

For more than 10 years now, increasing disquiet has been expressed in some quarters of the image-coding fraternity that use of the algorithms so far described might not be the appropriate route to really efficient coding in the long term (Kunt et al., 1985). The majority use regular subdivision techniques to process small square blocks of the image without regard to the object detail contained therein, and at extremely low rates it is all too easy for the block structure itself to appear as an artifact in the reconstructed image. There is, in any case, a further argument against simplistic division of an image in this way. Given increasing specialization in digital image/video services, it will become more important to code specific objects, rather than simply whole scenes, for efficient transmission. In many situations, the background may just be irrelevant to the procedure, especially when coding capacity is at a premium. There is therefore nowadays increasing interest in coding actual objects, for which they must first be extracted from the image by some segmentation technique. Although it must be admitted that we are still in the early stages of being able to do, even with difficulty, what the eye does effortlessly, good results in simplified situations can be achieved. Typical application areas relate more to video rather than to still picture coding (see later); nevertheless, work in the latter area has been fairly extensively reported. Thus, a head-and-shoulders image may be adaptively segmented (using smaller thresholds in areas of greater importance—eye and mouth detail, for example) and the region outlines coded using some sort of differential chain code (Soryani and Clarke, 1992). Internal areas can then be coded simply as uniform regions or, more accurately, with low-order polynomial fits. Naturally, approximations are evident in the output at low rates, but they take a different form compared with those of frequency domain approaches. In the latter, omitting high-frequency detail leaves a characteristic out-of-focus or fuzzy image. In segmented coding, the edge detail remains sharp (which the eye may well find preferable) even at low rates. The cost is the unnatural contouring which may appear. Regions which have a gradual shading profile across them appear as constant in luminance or color, or have noticeable steps connecting them to neighboring areas. At equivalent rates, however, the image coded in this way can retain much more of the important structure inherent in the original than can one processed using transform coding. Figure 6 shows the image of Figure 1 coded in this way at the same rate as Figure 3. A comparison of the two is illuminating.

It may be worthwhile here commenting briefly upon the relative performance of the major methods discussed above. To those unacquainted with image coding, it might appear a reasonable assumption that, given the wide variety of approaches—clustering, transformation, frequency subdivision, quadtree decomposition, etc.—a similar diversity of rates and output qualities will result. Although it is true that different techniques produce different kinds of degradation in the reproduced image (poor edge detail for frequency methods, block artifacts for block-structured algorithms, contouring in the case of segmentation, etc.), when pushed to the limit, their performance, if not identical, turns out to be very similar. Thus, well-designed algorithms of all kinds (save for predictive coding, which processes individual image elements) will give, for a reasonably detailed input image, essentially artifact-free reproduction at around 0.5 bit/element (thus, at the resolution which the printing process provides, versions of Figure 1 coded at this rate using transform, wavelet/subband, or vector quantization methods would be indistinguishable from the original), excellent quality at twice this rate and (as we have seen) noticeably degraded performance in the region of 0.25 bit/element. At present, in this lower region, multiresolution wavelet schemes with maybe vector quantization of wavelet planes seem to offer most hope of further improvement. The prospect of excellent reproduced quality at, say, one tenth of the above values seems more than remote.

Figure 6. The image of Figure 1 coded with a segmentation scheme at 0.2 bit/element.

V. MOTION

If there is one thing above all that the prevalence of television throughout the world as a provider of information and entertainment demonstrates, it is the overwhelming preference of the human observer for moving images. As in the case of still pictures, there has long been a concomitant interest in ways of processing these at as low a rate, given quality of reproduction constraints, as possible. It is only with the comparatively recent development of large-scale, on-chip storage, however, combined with the ready availability of ultra-high-speed hardware, that it has become practicable to implement such schemes. One way, of course, is simply to process image sequences on a frame-by-frame basis, i.e., without regard for any interrelation between them. Just as it is logical, though, to consider objects rather than arbitrary square blocks in the case of still images, so too, these objects are not only present, but also move within image sequences, and so the estimation of motion and its compensation have assumed increasing importance in image coding over the past 20 years or so.

Early work on motion estimation is represented by algorithms involving both the space and the frequency domain (Limb and Murphy, 1975; Haskell, 1974). In the former, the ratio of frame-to-frame differences, over the moving area, to the sum of element to element differences in the present frame was used to give an object speed measure, whereas in the latter the Fourier shift theorem can give an indication of motion via its phase shift term. Work started in earnest, however, in the late 1970s with the development of a recursive steepest descent algorithm which minimized the inter-frame difference signal, in an algorithm which could also be modified to account for problems with changing illumination (Netravali and Robbins, 1979; Stuller et al., 1980). Intensive development of this algorithm by various workers continued for the next decade or so, but problems with reliable determination of changing areas and choice of a suitable initial estimate for the recursion meant that alternative schemes came into prominence and, as will be seen, were eventually incorporated into standard algorithms. The technique most widely used at present is based upon the taking of a block of elements in the present frame and simply searching for a similar block in the previous frame which minimizes some function of the frame-to-frame difference over the area—mean square or mean absolute error (Jain and Jain, 1981). Prior to search, it can be advantageous to test the initial error against a threshold; if it is small enough motion compensation is not needed, anyway. The relative locations of blocks in present and previous frames are then characterized as a motion vector which must be transmitted to the decoder. This simple correlation-like technique is computationally intensive (search over a ± 7-element displacement in both x and y directions requires the error to be calculated at over 200 locations); and although it is now possible at real-time rates and is indeed the preferred approach, as if fully searched it guarantees to find a true minimum in the error function, the literature contains a long history of reduced-search approaches—using a search route covering only a selection of locations, minimizing x and y error terms sequentially, and so on (Kappagantula and Rao, 1985). An added advantage is that, having all possible error values, displacement can be determined via interpolation to an accuracy better than a single element. A refinement which can aid fast determination of the location corresponding to the minimum frame-to-frame error is hierarchical block matching (Wang and Clarke, 1990). Here, an image pyramid consisting of a sequence of planes, each formed by averaging over a small region of the previous one, is used top to bottom: A rapid search at the lowest resolution level forms the initial estimate for the next, and so on. This approach is also useful in dealing with large frame-to-frame differences.

Motion estimation and compensation form part of the standard present-day approach to low-rate video transmission, but their applicability is wider that simply optimizing frame-to-frame prediction. They can also be used for interpolation, when maybe every other frame in a sequence is dropped to achieve a minimum bit rate (Thoma and Bierling, 1989). In this situation, simple static interpolation produces unacceptable motion artifacts when used to reconstruct the missing frames, and motion compensation enables the movement of the object(s) to be accounted for in this operation (note that in this case, vectors representing true motion are necessary, not simply those which indicate a minimum in the frame difference signal). Areas covered up and uncovered by the moving object are accounted for by forward and backward extrapolation, respectively. Another application of motion compensation is in noise filtering, where, if the motion estimate is good, strong low-pass filtering may be applied along the object path to reduce noise (Dubois, 1992).

Naturally, the assumption that square block translation represents true object motion is only approximate, and experiments have been done in an attempt to allow more refined tracking of object rotation, scale change, and general motion throughout a video sequence (Wu and Kittler, 1990). There is also continuing work on effective frequency domain algorithms for motion compensation (Young and Kingsbury, 1993). The area in general is one where we are yet some way from the goal of being able to track object motion reliably through a sequence of frames and employ the knowledge so gained in reducing yet further the data rate for video transmission.

VI. IMAGE SEQUENCE CODING

From the early days of television, ways have been sought to reduce the bandwidth (or channel capacity) necessary for its transmission. Both the increasing availability of high-speed digital technology and the development of new video services have intensified this search to the point where, over the past few years, coding standards have been introduced to cope with delivery of the latter, while leaving very much of a question mark over where research in the area should now be heading. First of all, though, what of the algorithms available?

A. Three-Dimensional Coding. Having coded single-image frames with a variety of two-dimensional techniques, the logical extension of this approach is to code sets of frames making up a sequence using three-dimensional extensions of the same methods. Predictive coding can be successfully implemented using adaptive prediction from present and previous frames (Zetterberg et al., 1982). Unfortunately, it has the same problem as that which besets its two-dimensional counterpart: It cannot produce the very low rates required for many of today's applications. Our other major space-domain technique—vector quantization—can likewise be implemented in three dimensions, coding maybe $4 \times 4 \times 4$ volumes (Huguet and Torres, 1990); few examples have been reported, however. In the frequency domain, we have transform, subband, and wavelet coding, which can all be employed in three dimensions (Akiyama et al., 1990). In general they have not generated much interest, possibly because the driving force in image coding over the past 20-odd years has been transform coding, which is particularly unsuited to the three-dimensional operation since it needs a block length of at least eight elements to make efficient use of data correlation. In the temporal domain, eight frames represents an unacceptable delay for interpersonal operation (videophone and videoconference), and this has led to emphasis being placed upon hybrid techniques.

B. Hybrid Coding.

1. Hybrid Transform Coding. Over the past 15 years or so hybrid transform coding has developed to become the predominant method of coding image sequences. Developed from an original intraframe algorithm (Habibi, 1974), it has the object of using simple predictive coding (plus motion compensation) in the temporal domain (interframe coding) together with two-dimensional intraframe transform coding spatially to achieve a combination of low rates and acceptable quality. Implementation is possible in two equivalent ways: We could intraframe transform code successive frames and then predict coefficients from frame to frame, or predict spatially first and then transform code the motion-compensated frame differences (Ericsson, 1985). Since the preferred method of motion compensation is block matching, performed in the spatial

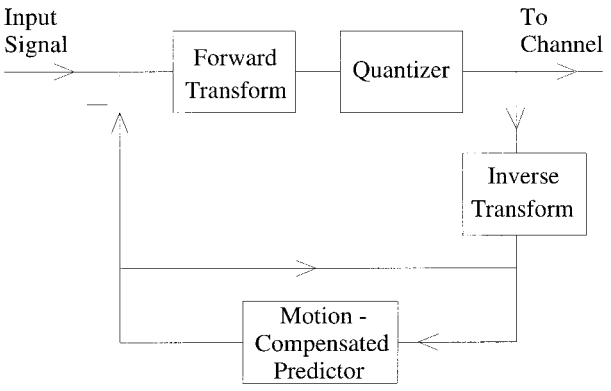

Figure 7. Hybrid coding.

domain, the latter approach is taken and the basic structure is shown in Figure 7. The output transform coefficient arrays are zigzag scanned (see Section IIIB) and coded with Huffman or arithmetic coding. This is the basic format of the algorithm used in the standards to be described in Section VII. It does, however, suffer from a theoretical if not practical problem, and that is the basic conflict between what each part of the algorithm is trying to do. The job of the predictive operation is to remove as much structure as possible from the error signal. We hope, therefore, that this signal will contain a minimum of redundancy and correlation and, if the prediction fitted the image characteristics properly, it would be, in signal-processing terms, "white," i.e., with a flat frequency spectrum. A good motion-compensation scheme can help greatly in this respect. So what is left for the intraframe transform coding to do? Well, not very much, and there is a significant amount of evidence published demonstrating that following the motion-compensated prediction process with the transform gives little if any further benefit (Chen and Pang, 1992). The reasons for including the transform in practice seem to be twofold. First, even when motion compensated, the practical prediction operation can be quite inefficient. Also, block-based motion compensation is only approximate, and examination of the error sequence frequently shows up areas where significant structure exists. Second, the transform itself is very powerful, especially when combined with the scheme mentioned earlier for subsequent coding of coefficients in terms of their amplitude and distance along the scan path from the previous significant value. In any case, after years of development, a well-defined scheme has been established which now lies at the heart of all the standards at present in place for video coding. Figure 8 shows a frame from the "Claire" sequence coded at 0.07 bit/element (which would allow transmission at 15 frames/s at a rate of approximately 64 kb/s). Note that the usual transform coding artifacts are apparent, and also that the quality of the sequence, when run in real time, is much better that the single, isolated frame would indicate.

2. Other Hybrid Schemes. Given the question mark which hangs over the motion-compensated prediction/transform coding approach, it is not surprising that many alternative schemes have been devised for low-rate processing of image sequences, although none, of course, has the prominence accorded to the transform approach via its incorporation into the standard algorithm. Vector quantization, for example, may be used to process the motion-compensated interframe predictive image sequence (Watanabe and Suzuki, 1989). In most cases, since fast processing is essential in a video application, use is made of some form of test to ensure that only those blocks which still contain significant structure are vector quantized. Again, it is often the case that before vector quantization, the signal is normalized as mentioned in Section IIIC. Alternatively, the lattice codebook structure may be used. Here, the codebook is not derived from a training sequence, but is simply a predetermined regular lattice in the vector space known to both transmitter and receiver, which can be searched very quickly indeed (Sampson and Ghanbari, 1992). Naturally, the efficiency of such an approach is lower than that of a more conventional design, but surprisingly, often by not very much.

Subband and wavelet approaches can also be used to code the error sequence (Westerink et al., 1990). In these cases, however, there is more latitude in the arrangement of the various operations, since the output of the frequency processing step is still a spatial image, albeit only containing certain spatial frequencies. Motion compensation can therefore be carried out after the subband filtering step; indeed, different subbands may have their own motion compensation schemes. More flexibility can be introduced by the use of wavelet decomposition, since this produces images at various resolution levels which can be used to implement multiresolution motion compensation at the same time (Yao and Clarke, 1993). Naturally, in all such schemes some further form of coding must follow the frequency decomposition. Predictive coding can be used for this, but vector quantization is a more popular alternative.

Straightforward space domain operations can also be applied to the motion-compensated prediction. Segmentation can be used, of course, and this is an area capable of further development. Even simple schemes can give reasonable results for scenes which are not too complex, especially if adaptive techniques are applied to important features (head and shoulders in videophone images) (Soryani and Clarke, 1992). Figure 9 shows a frame from the "Claire" sequence, this time coded using a hybrid version of the method used for Figure 6. The rate is the same as that used for Figure 8. Other spatial techniques can similarly be used: quadtree decomposition starting with blocks of 8×8 or 16×16 elements, or the reverse—successive merging operations starting with 2×2 blocks. These can also be combined with region approximation methods to improve performance with some increase in complexity. Figure 10 shows the result of coding the same frame with an adaptive quadtree division scheme combined with simple bilinear interpolation over the various block sizes. This time, the rate is 0.09 bit/element. Note that al-

Figure 8. A frame from the "Claire" sequence hybrid transform coded at 0.07 bit/element.

Figure 9. As in Figure 8, but coded with a hybrid version of segmented coding at 0.07 bit/element.

though this is a block structured scheme, blocks are of different sizes and so are much less visible than in Figure 8. Edge detail is better preserved as well.

C. Model and Object-Based Coding. The 1980s saw several significant developments in image coding: the change in structure of conventional adaptive transform coding, the introduction of vector quantization, and, later, the start of a move toward standardization. Of more fundamental significance was the feeling that the subject was not moving along quite the correct path and that more conventional techniques had perhaps reached their performance limits. One result of this was a radical suggestion to use a quite different approach to the very low rate coding of image data (Forchheimer and Kronander, 1989; Li et al., 1994). In the case of a well-defined object to be imaged and this image transmitted (the object receiving by far the most attention in this context being the head-and-shoulders image of a videophone sequence), the scheme works by establishing a three-dimensional model of the object, maybe based upon a polygonal wire-frame structure. A similar structure is also used at the decoder, together with computer graphics shading techniques, to represent the object data. Feature analysis methods are then used at the coder to determine the movement of significant points on the object (eye direction, disposition of the mouth, etc.) which are then transmitted to the decoder to control the displacement of corresponding points on the decoder model. Remarkable visual results can be achieved in this way at transmission rates in the region of a few kilobytes per second for constrained classes of object and degrees of motion. There are severe problems, however, of which the major one seems to be the reliable tracking of original feature alteration (analysis). Reconstruction (synthesis) seems to be less of a problem. This kind of image coding (where we already have available an image model) is sometimes called semantic-based image coding. An alternative, called object-based coding, differs from the above approach in that it does not have a ready-made object model which may be modified by the signals derived from the analysis stage of coding (Musmann et al., 1989). In this more general approach, objects in the scene are specified by three parameters—shape, color, and motion—which are transmitted to the decoder and also used at the coder to resynthesize a model which can be tested against the input. Good results can be achieved by this approach, and these can be made better still by the use of a combination of model-based and more traditional (waveform) coding methods, the latter being used to code regions which are identified as being those where modeling has failed. However, robust and consistent application of such techniques to arbitrary image detail must be seen as being a long way off.

VII. STANDARDS

Undoubtedly, the one advance which has brought image coding over the past few years to general notice at the forefront of digital image processing (after decades spent in obscurity in relatively few academic and industrial research laboratories around the world) is the establishment of standards which by any criterion has been a monumental achievement. In view of the enormous volume of literature which now exists detailing the various specifications, most of which are concerned, in any case, with matters of format and syntax, it is more relevant here to concentrate upon the way in which algorithms already described are employed in combination to code image data, from still pictures to video at rates between tens of kilobytes per second and those relevant to HDTV.

A. JPEG. The basic Joint Photographic Expert's Group standard for coding both monochrome and colour still images was introduced in 1991 as a result of collaborative activity between working and study groups of the ISO and CCITT (now ITU-T) and was intended to provide reconstruction of conventional ITU-R recommendation 601 format 4:2:2 pictures (720 luminance and 360 chrominance active samples per line, with 576 lines) at reasonable quality in the 0.25–0.5 bit/element region through to no perceptible degradation at about 1.5 bit/element (Clarke, 1995; Rao and Hwang, 1996). Image data (8-bit resolution in the baseline version) has its mean value removed and is then subjected to an 8×8 discrete cosine transform, followed by scaling and uniform quantization according to predefined quantization tables. The DC coefficient is predicted block to block and the 63 other (AC) coefficients coded using the run length/level method along a zigzag scan path as described previously, for which Huffman coding tables are provided. Transmission of coded blocks left to right and top to bottom across the image represents the standard method of processing. However, this is by no means the most efficient method, and a progressive mode may be

Figure 10. As in Figure 8, but coded using adaptive quadtree decomposition and interpolation at 0.09 bit/element.

invoked in which successive approximations are sent, i.e., a rough representation of the image as a whole is produced first, followed by a gradual increase in resolution. This can be done in one of two ways. Coefficients may be sent either after subdivision into spectral bands, i.e., all low-order terms first followed by higher-frequency components (spectral selection) or by sending all most significant bits first, followed by those of lesser significance (successive approximation). A further possibility is available (albeit with relatively low levels of data reduction). This lossless mode employs simple one- or two-dimensional predictive coding, together with entropy coding of the error signal—there is no further approximation of the error signal by scaling/quantization as in the case of the transform coding mode. As its name implies, this mode allows exact reproduction of the input image with maybe 2 or 3:1 bit rate reduction. A further refinement may be made; we have seen how a hierarchical operation can usefully provide images at a variety of scales (multiresolution). Here, such an algorithm is provided by allowing the input image to be successively subsampled by a factor of two horizontally and vertically. The low-resolution output can then be interpolated (upsampled) and used as a prediction at the next higher level, the difference error image being separately coded by the algorithm itself. Other extensions are also available—the ability to change quantization parameters on a block-by-block basis, the definition of an appropriate image interchange format for different applications, and so on.

B. MPEG-I. JPEG provides a good basic example of the harmonious combination of the various image coding techniques which have been previously described: prediction, transformation (including run-length coefficient scan), variable-length coding, and the possibility of hierarchical operation. All other standards are considerably more complex, since processing of image sequences is involved. MPEG-I, intended for (noninterlaced) video and audio coding at up to about 1.5 Mb/s (typically for CD-ROM applications), with Source Input Format—360 elements/line and 240 lines (NTSC) or 288 lines (PAL)—was agreed on as an ISO standard in 1992 (Rao and Hwang, 1996). Here, we consider the video coding only. The need to implement various editing techniques—forward and reverse video at normal and fast speeds, random access, and audio/visual synchronization—implies that the overall structure of MPEG-I is complex. The basic coding algorithm is still, however, the combination of motion-compensated interframe prediction and intraframe transform coding. The 8×8 blocks are combined into so-called macroblocks consisting of four 8×8 luminance blocks (over which resulting 16×16 block motion compensation is carried out) plus two chrominance blocks. Four kinds of picture are now specified: (a) I pictures, intraframe coded, which can provide random access points; (b) P pictures, predicted from previous I or P pictures; (c) B (bidirectional) pictures, predicted from both past and/or future I or P pictures (for these data, reordering may be necessary); and (d) D pictures, which allow fast-forward mode with restricted quality by coding only each block DC coefficient. There are several different options for macroblocks in P and B pictures depending on whether motion compensation is used, the quantizer scale has been altered, and so on. The only new element is the introduction of B pictures, which, however, are never used to form a prediction.

C. MPEG-II. In 1994, an expanded, higher-rate extension of MPEG-I was standardized by the ISO to support a range of full-motion, interlaced video and audio coding applications over an extended range of transmission rates—MPEG-II (Rao and Hwang, 1996). MPEG-III, which was initially directed at digital television applications, was brought into this standard, which, as a result of collaboration with the ITU, is also ITU-T Recommendation H.262. Basically, the algorithm is again that of MPEG-I with additions to cope with inputs that may have either field or frame formats, and with extra emphasis on scalability (where part of the data stream can be neglected and decoding at a lower quality level can still proceed), provided (a) spatially in the multiresolution hierarchical manner already described; (b) for signal-to-noise (SNR) ratio (at the same resolution but with different quality), where an enhancement layer is used to refine the accuracy of the DCT coefficients transmitted in the base layer; and (c) temporally, where the enhancement layer carries the prediction error produced by using the base layer data as a prediction for the input signal. The areas of application of MPEG-II are so diverse that it is not possible for any single set of parameters to be generally applicable. This has resulted in the introduction of so-called profiles (five, from simple to high) and levels (four, from low to high). These allow for picture sizes from 352×288 to 1920×1152, and rates from 4 to 100 Mb/s to be allocated. Thus, for example, MP@ML (Main Profile at Main Level) implies a picture size of 720×576, 30 frames/s and a rate of 15 Mb/s (appropriate for general use), whereas HP@HL (High Profile at High Level) allows for HDTV applications—1920×1152, 60 frames/s, 100 Mb/s in the enhancement layer, and so on.

D. H.261/H.263. Turning now to lower rates of transmission, ITU-T Recommendation H.261 was standardized in 1990 for use at rates of $p \times 64$ kb/s, where p lies between 1 and 30 (Clarke, 1995). Input luminance format is, recognizing the lower transmission rates involved, noninterlaced so-called Common Intermediate Format (CIF), 352×288 or one quarter of this—(QCIF), 176×144. The block/macroblock structure is as in MPEG-I, with 33 macroblocks making up a group of blocks (GOB) $[(11 \times 16) \times (3 \times 16)]$, 176×48 elements and 12 GOB (2×6) comprising a picture. The transform part of the algorithm again operates on 8×8 blocks and intraframe operation can be selected (similar to coding I pictures in MPEG) if a large prediction error indicates high activity or rapid motion. Given the fact that operation in an error-prone environment may be required, H.261 makes provision for error correction via the inclusion of an error-correcting code. More recently, attention has focused upon transmission at rates below 64 kb/s [one application being use on the Public Switched Telephone Network (PSTN)], and an extended version of the standard, H.263 (Wiegand et al., 1996), has been introduced for this purpose. It forms part of a more general ITU recommendation (H.324) for a videophone/multimedia terminal using the PSTN (Schaphorst, 1996) and embodies refinements which, overall, allow H.261 quality at approximately half the bit rate. In this case input formats are QCIF and sub-QCIF (128×96) (larger options are also available), and the major advances all involve the prediction/motion compensation operations, apart from an optional arithmetic coding capability. In H.261, the predictive loop included a noise-smoothing filter, omitted here since motion vectors are calculated by interpolation to one half element accuracy, which operation has a low-pass characteristic in any case. There is the possibility of generating four motion vectors per macroblock (one per 8×8 block) and also of incorporating overlapping block-matching motion estimation. Here, each predictive term is the weighted average of three predicted values, derived via the use of three motion vectors: that of the current luminance block and those of the nearest two adjacent blocks. In addition, motion vectors may point outside the picture, in which case the last available (edge)

element is used instead. A further inclusion is that of a PB frame (MPEG terminology). This consists of one P picture predicted from the last decoded P picture and one B picture predicted from the last decoded P picture and also the P picture currently being decoded. This allows an effective doubling of the frame rate with only a small increase in transmission cost. As far as error control is concerned, no explicit error correction coding is included, and techniques need to be evolved to cater for the error-prone environments in which H.263 is likely to be used. Possibilities are error concealment (copying a complete corrupted frame from the previous, good frame) or the use of error tracking based upon a feedback channel signal (Girod et al., 1996).

E. MPEG-IV. Current interest in MPEG standardization centers upon MPEG-IV, the activity surrounding which was first proposed in late 1992, with the intention of this having been developed to the status of an International Standard by late 1998 (Pereira and Koenen, 1996). Initially, the project had the object of providing in the long term a very low-bit-rate video-coding capability. We have already seen that various views had been expressed upon the technical aspects of this prospect—mainstream approaches as exemplified in the H.261/3 structure were reaching a limit as far as the low-rate/acceptable quality trade-off was concerned (H.263 was under development at this time), and segmented/model-based techniques were felt to be as yet untried and of insufficient generality to form a working standard. Given the fact, then, that a significant increase in performance, in a purely low-rate sense, could not be guaranteed within a realistic time scale, in mid-1994 the decision was taken to make an important alteration in the proposal's terms of reference. This involved a large-scale widening of the scope of the plan, still to incorporate high compression, but also to support new means of dealing with image *content*, in an audiovisual/multimedia context, and to move away from simply processing frame-based video. In a sense, then, this represented a move similar to that made in MPEG-II (to incorporate profiles and levels), where in contrast to the relatively narrow areas of operation of, say, H.261 or MPEG-I, toolkits are (or will be) available to cope with an enormous range of input and applications, from surveillance and traffic monitoring, teleshopping, computer graphics, and databases to home entertainment, games, audiovisual services, simulation and distance educational provision, and so on. The proposals emphasize the central importance of objects (which may be real or synthetic) initially coded at their own appropriate resolution levels and then made generally available through the incorporation of scalability (as previously discussed) at a variety of resolutions (Sikora, 1997). The unifying mechanism allowing manipulation of such a disparate collection of processes and representations is the MPEG-IV Syntactic Descriptive Language (MSDL), which will define interfaces between tools, be used to transmit decoding rules to the receiver, and so on. The basic processing operation is defined in terms of so-called video object planes (VOPs), each of which defines a significant object in the scene and which can be made individually available for further processing. Defining all the object detail separately in this way and then similarly decoding and reassembling all the VOPs allows the complete input scene to be reconstructed. Information on the shape, motion, and texture of each video object so defined is coded as a video object layer (VOL). Present definitions allow separate shape (binary or gray-scale) coding of the various VOPs, and motion/texture information by an algorithm which will by now be familiar to readers: motion-compensated predictive/transform coding. Where blocks cross shape boundaries, the object detail is padded out to complete the block for transformation. It is envisaged that eventually MPEG-IV will support both MPEG-I and MPEG-II functionalities, and to this end it recognizes the conventional rectangular video frame as a special case of a VOP. We thus have the promise of an eventual total integration of coding capabilities covering all manner of applications, input formats, and transmission rates, and implemented by selection of the appropriate set of tools from the MPEG-IV toolbox. Indeed, just identifying a specific item of information from the enormous volume of material suitable for MPEG-IV input is likely to become such a tortuous operation that work on this requirement has just started—so-called (for the moment) MPEG-VII.

VIII. GENERAL COMMENTS AND CONCLUSIONS

No one needs reminding nowadays that as we approach the millennium, we are living in a time of phenomenal change in our ability to access information. Undoubtedly, this has been due to the development of the computer and large-scale high-speed integrated circuits, and these have contributed to major advances in communication via speech and the written word. Overwhelmingly, however, it has been the image that has been in the forefront of this revolution as even a cursory examination of our image-processing capabilities half a century ago and now will reveal. Oddly enough, we may be in the position we are now just because pictures, especially in moving form, contain so much information that they present an interesting and relevant challenge to the technology existing at any particular point in time as far as data reduction developments are concerned.

From the early days of predictive coding, via extremely powerful transform-coding algorithms and the development of variable word-length coding and motion compensation, all of which have given us the standards we have today, necessary transmission rates have fallen from megabytes to kilobytes per second; in addition, a plethora of alternative techniques has grown up, all of which, even if not standardized or, in some cases, very successful, have taught us more about the relation between the human observer and the visual representation of scenes. And priorities? In 1985, it was suggested that conventional algorithms had reached their limit and that we should be coding something other than square image blocks. Therefore, here we are 13 years later with all of our standards still based on just that approach, but with quite a bit more compression and flexibility to show, whereas in parallel, object- and model-based approaches creep painfully toward more generic and robust application. Shall we now say that the old methods have really reached the end of the road? We can always argue, of course, that more research is needed, and so it is, especially in bringing the HVS into the system—how do we really perceive images, still or moving? how can we perform segmentation even with a tiny fraction of the ease with which the eye does it? etc? Oddly enough, what was once considered to be the major stumbling block, processing speed (or the lack of it) seems to have disappeared from the equation. No longer do we have to accept compromises in algorithm design because the hardware cannot cope, and the days of an overnight job at the computer center simply to transform code a single image frame now seem like just a bad dream.

So where do we go from here? For fixed services, there seem to be two distinct opinions. One says that given the enormous bandwidth of optical fiber, the provision of new image services can be handled with only moderate levels of compression, or even none at all (this view ignores the fact of life that, given a new source of transmission capacity, human ingenuity will soon find ways not only of filling it up, but of finding reasons for requiring still more). The alternative argues that

even so, there will always be a place for satellite and conventional broadcast distribution, in which case compression algorithms will play a vital part in providing choice within an ever-increasing volume of program material. Whichever of these holds sway eventually, we can be safe in the knowledge that it is difficult to attach a fiber-optic link to a moving vehicle, and finite radio channel space together with an ever-increasing demand for video communication in this context (public service applications, for example) is a guarantee that image and video compression techniques will become and remain common in this area. Again, the standardization activity of the past 10 years has to be seen by any criterion as a monumental achievement in drawing together scientific, technological, and economic and commercial considerations. Yet, in one sense, all arguments about standards and rationalization may not matter at all—the diversity of applications and techniques for image and video coding may mean that I as a service provider can arrange for you as a consumer to download my decoding software prior to transmission of the data, and we can all use whatever algorithm is dictated as economic by the actual application. Therefore, it may well be that there is room for everyone after all.

REFERENCES

N. Ahmed, T. Natarajan, and K.R. Rao, Discrete cosine transform, IEEE Trans Comput C-23,(1974), 90–93.

T. Akiyama, T. Takahashi, and K. Takahashi, Adaptive three-dimensional transform coding for moving pictures, Proc Picture Coding Symp, Cambridge, MA, 26–28 March, 1990, Paper 8.2.

S.T. Alexander and S.A. Rajala, Optimal gain derivation for the LMS algorithm using a visual fidelity criterion, IEEE Trans Acoust Speech Signal Process ASSP-32 (1984), 434–437.

A.D. Barnsley and A.D. Sloan, A better way to compress images, BYTE, January (1988) 215–223.

P.J. Burt and E.H. Adelson, The Laplacian pyramid as a compact image code, IEEE Trans Commun COM-31 (1983), 532–540.

C.F. Chen and K.K. Pang, Hybrid coders with motion compensation, Multidimen Syst Signal Process 3 (1992) 241–266.

T.C. Chen, A lattice vector quantization using a geometric decomposition, IEEE Trans Commun COM-38 (1984), 704–714.

W.H. Chen and W.K. Pratt, Scene adaptive coder, IEEE Trans Commun COM-32 (1984), 225–232.

W.H. Chen and C.H. Smith, Adaptive coding of monochrome and color images, IEEE Trans Commun COM-25 (1977), 1285–1292.

C.K. Chui, Wavelets: A tutorial in theory and applications, Academic Press, San Diego, 1992.

R.J. Clarke, Transform coding of images, Academic Press, San Diego, 1985.

R.J. Clarke, Digital coding of still images and video, Academic Press, San Diego, 1995.

R.D. Dony and S. Haykin, Neural network approaches to image compression, IEEE Proc 83 (1995), 288–303.

E. Dubois, Motion-compensated filtering of time-varying images, Multidimen Syst Signal Process 3 (1992), 211–239.

S. Ericsson, Fixed and adaptive predictors for hybrid predictive/transform coding, IEEE Trans Commun COM-33 (1985), 1291–1302.

R. Forchheimer and T. Kronander, Image coding—from waveforms to animation, IEEE Trans Acoust Speech Signal Process ASSP-37 (1989), 2008–2023.

A. Gersho and B. Ramamurthi, Image coding using vector quantization, ICASSP Proc, 1982, pp. 428–431.

B. Girod, Psychovisual aspects of image communication, Signal Process 28 (1992), 239–251.

B. Girod, N. Faerber, and E. Steinbach, Standards based video communications at very low bit rates, EUSIPCO Proc, 1996, pp. 427–430.

R.M. Gray, Vector quantization, IEEE ASSP Mag, April (1984), pp. 4–29.

A. Habibi, Hybrid coding of pictorial data, IEEE Trans Commun COM-22 (1974), 614–624.

B.G. Haskell, Frame-to-frame coding of television pictures using two-dimensional Fourier transforms, IEEE Trans Inf Theory IT-20 (1974), 119–120.

J. Huguet and L. Torres, Vector quantization in image sequence coding, EUSIPCO Proc, 1990, pp. 1079–1082.

A.E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformation, IEEE Trans Image Proc 1 (1992), 11–30.

J.R. Jain and A.K. Jain, Displacement measurement and its application to interframe image coding, IEEE Trans Commun COM-29 (1981), 1799–1808.

S. Kappagantula and K.R. Rao, Motion compensated interframe image prediction, IEEE Trans Commun COM-33 (1985) 1011–1015.

M. Kunt, A. Ikonomopoulos, and M. Kocher, Second generation image coding techniques, IEEE Proc 73 (1985), 549–574.

T.C. Lee and A.M. Peterson, Adaptive vector quantization using a self-development neural network, IEEE J Select Areas Commun 8 (1990), 1458–1471.

H. Li, A. Lundmark, and R. Forchheimer, Image sequence coding at very low bitrates: A review, IEEE Trans Image Proc IP3 (1994), 589–609.

J.O. Limb and J.A. Murphy. Measuring the speed of moving objects from television signals, IEEE Trans Commun COM-23 (1975), 474–478.

Y. Linde, A. Buzo, and R.M. Gray, An algorithm for vector quantizer design, IEEE Trans Commun COM-28 (1980), 84–95.

T.D. Lookabaugh and M.G. Perkins, Application of the Princen-Bradley filter bank to speech and image compression, IEEE Trans Acoust Speech Signal Process ASSP-38 (1990), 128–136.

H.S. Malvar, Signal processing with lapped transforms, Artech House, London, 1992.

T. Murakami, K. Asai, and E. Yamazaki, Vector quantizer of video signals, Electron Lett 18 (1982) 1005–1006.

H.G. Musmann, "Predictive image coding," Image transmission techniques (Advances in electronics and electron physics, Suppl 12), W.K. Pratt (Editor), Academic Press, New York, 1979, pp. 73–112.

H.G. Musmann, M. Hotter, and J. Ostermann. Object-oriented analysis-synthesis of moving images, Signal Process Image Commun 1 (1989), 117–138.

H.G. Mussman, P. Pirsch, and H.-J. Grallert, Advances in picture coding, IEEE Proc 73 (1985), 523–548.

A.N. Netravali and J.D. Robbins, Motion-compensated television coding: Part 1, Bell System Techn 58 (1979), 631–670.

J.B. O'Neal, Jr., Predictive quantizing systems (differential pulse code modulation) for the transmission of television signals, Bell Syst Techn 45 (1966), 689–721.

S. Panchanathan and M. Goldberg, Min-max algorithm for image adaptive vector quantization, IEE Proc Commun Speech Vision 138 (1991), 53–60.

F. Pereira and R. Koenen. Very low bit rate audio-visual applications, Signal Process Image Commun 9, 1996, 55–77.

K.R. Rao and J.J. Hwang, Techniques and standards for image, video and audio coding, Prentice Hall, Upper Saddle River, New Jersey, 1996.

A. Said and W.A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans Circuits Syst Video Technol 6 (1996), 243–250.

H. Samet, The quadtree and related hierarchical structures, Comput surveys 16 (1984), 187–260.

D.G. Sampson and M. Ghanbari, Interframe coding of images using lattice vector quantization, Proc 4th IEE Int Conf Image Process, Maastricht, Netherlands, 7–9 April, 1992, pp. 131–134.

R.A. Schaphorst, Status of H.324—the videoconference standard for the public switched telephone network and mobile radio, Opt Eng 35 (1996), 109–112.

J.M. Schapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans Signal Process SP-41 (1993), 3445–3462.

T. Sikora, The MPEG-4 video standard verification model, IEEE Trans Ckts Syst Video Technol 7 (1997), 19–31.

M. Soryani and R.J. Clarke, Segmented coding of digital image sequences, Proc IEE I Commun Speech Vision 139 (1992) 212–218.

P. Strobach. Quadtree-structured recursive plane decomposition coding of images, IEEE Trans Signal Process SP-39 (1991), 1380–1397.

J.A. Stuller, A.N. Netravali, and J.D. Robbins, Interframe television coding using gain and displacement compensation, Bell Syst Techn J 58 (1980), 1227–1240.

A.M. Tekalp, M.K. Ozkan, and A.T. Erdem, Image modelling using higher-order statistics with application to predictive image coding, ICASSP Proc, 1990, pp. 1893–1896.

R. Thoma and M. Bierling, Motion compensation interpolation considering covered and uncovered background, Signal Process Image Commun 1 (1989), 191–212.

Q. Wang and R.J. Clarke, Motion compensated sequence coding using image pyramids, Electron Lett 26 (1990), 575–576.

H. Watanabe and Y. Suzuki 64 kbit/s video coding algorithm using adaptive gain/shape vector quantization, Signal Process Image Commun 1 (1989), 87–102.

P.H. Westerink, J. Biemond, and G. Muller, Subband coding of image sequences at low bit rates, Signal Process Image Commun 2 (1990), 441–448.

T. Wiegand, M. Lightstone, D. Mukherjee, T.G. Campbell, and S.K. Mitra, Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard, IEEE Trans Ckts Syst Video Technol 6 (1996), 182–190.

J.W. Woods (Editor), Subband image coding, Kluwer, Dordrecht, 1991.

S. Wu and J. Kittler, A differential method of simultaneous estimation of rotation, change of scale and translation, Signal Process Image Commun 2 (1990), 69–80.

S. Yao and R.J. Clarke, Motion-compensated wavelet coding of colour images using adaptive vector quantization, Proc Conf Image Process, Theory and Applications, San Remo, Italy, 14–16 June, 1993, pp. 99–102.

R.W. Young and N.G. Kingsbury, Frequency domain motion estimation using a complex lapped transform, IEEE Trans Image Proc 2 (1993), 2–17.

L.H. Zetterberg, S. Ericsson, and H. Brusewitz, Interframe DPCM with adaptive quantization and entropy coding, IEEE Trans Commun COM-30 (1982), 1888–1899.