Chapter XXVI
Pattern Based Video Coding

Manoranjan Paul
Monash University, Australia

Manzur Murshed
Monash University, Australia

ABSTRACT

People's demands are escalating with technology advances. Now, people are not happy with only text or voice messages, they like to see video as well. Video transmission through limited bandwidth, for example, an existing telephone line, requires an efficient video coding technique. Unfortunately, existing video coding standards have some limitations due to this demand. Recently, a pattern-based video coding technique has established its potentiality to improve the coding compared to the recent standard H.264 in the range of low bit rates. This chapter describes this technique with its background, features, recent developments, and future trends.

INTRODUCTION

Video conferencing, video telephony, teleteaching, telemedicine, surveillance, and monitoring systems are some of the video coding applications that have attracted considerable interest in recent years. The burgeoning Internet has increased the need for transmitting (nonreal-time) and/or streaming (real-time) video over a wide variety of different transmission channels connecting devices of varying storage and processing capacity. Stored movies or animations can now be downloaded and many reality-type interactive applications are also available via Web-cams.

The video itself is a series of still images (or frames) taken at some specific frame rate. Considering that the frame rate has to be fast enough to exploit the persistence of vision in creating the illusion of smooth motion as well as natural colour, the digital information content of a video can pose a significant challenge in the areas of efficient digital storage and transmission. For example, a 30 *frames* *per second* (fps) video with 24-bit per pixel true colour frames of moderate resolution (352×288 pixels) generates more than 70 mega bits per second. In order to cater to devices with limited power and storage with stringent transmission bandwidth requirements, these raw digital video data need to be compressed in the order of 10 to 10,000 times depending on the applications.

One way for video coding technology to facilitate the amount of video data compression needed is by eliminating redundant and visually insignificant data. Intraframe spatial redundancy is usually eliminated by run length encoding, while interframe temporal redundancy is eliminated by skipping a block of data. Insignificant data are usually eliminated by applying spatial subsampling by dropping some intermediate pixels from each frame, temporal subsampling by dropping some intermediate frames, and quantisation. The efficiency of these elimination processes is usually significantly improved by using block-based *motion estimation* (ME) and *motion compensation* (MC), and transforming pixel-level information to energy (frequency) domain.

Another problem which also effects the video data compression is the limited transmission bandwidth. For example, the low cost common networks like *public switched telephone network* (PSTN), *integrated services digital networks* (ISDN), and many computer networks normally allow for only several *kilo bits per second* (kbps) transmission. Even wireless transmission systems using cellular phones or *personal digital assistants* (PDA) operate under similar bandwidth restrictions. *Very low bit rate* (VLBR) video coding mandates bit rates between 8 and 64 kbps to facilitate video communications over these kinds of transmission media. Therefore, an efficient encoder is indispensable to enable the transmission of video. In this chapter we like to emphasise those video coding schemes which enable limited battery power and processing capacity devices, such as mobile phones and PDAs, to encode live video data in real-time and achieve significant improvement in coding efficiency so that the encoded steams could be transmitted cost-effectively at a much lower bit rate.

BACKGROUND

Reducing the bit rate by maintaining acceptable image quality has been a continuing effort for researchers over a long period of time. Block-based very low bit rate video coding addresses this trend at the extreme level where sacrificing quality to meet a more stringent bit rate is inevitable. A graceful degradation of quality while attaining the highest possible quality for the operating bit rate remains a challenge for the research community. There are two ways to reduce the video data. One is a trivial and simple way which can be applied with any other modern video coding technology with sacrificing video quality; another is a standard way which must be used for any professional or commercial purpose.

Simple Way of Compression

The simple ways to reduce the bit rate for a video sequence in generic coding paradigm are by extending the group of picture, down sampling the image size, skipping frames, nonmotion compensated blocks, residual-error-compensation, and by increasing quantisation values. These bit reduction techniques will be presented in the rest of the section.

- During the video coding, a video sequence is divided into a group of picture (GOP) 0. A GOP consists of one intracoded frame (I-frame), one or more predicted coded frame (P- frame), and one or more bidirectional coded frames (B- frame). This classification depends on what reference frames are used for encoding. No reference frame is used for I-frame, previous I- or P-frames are used as reference frames for P-frame, and previous and next I- or P-frames are used as reference frame for B-frames. The length of a group can only be increased by introducing more P- and/or B-frames. Although the image quality of a frame coded as an I-frame is the best, considerably fewer bits would be required if the frame were coded as a P- or B-frame. This leads to a natural trade-off between compression efficiency and image quality in that a small group exhibits better image quality with less compression; by contrast, a large GOP exhibits more compression with poorer quality. In low bit rate video coding a large GOP is preferable

- Down sampling of an image size reduces the bit rate quite significantly. A large size image format, naturally, needs more bits compared to a small size one. In low bit rate video coding applications, a small size video is preferable as it requires a small number of bits.
- Besides spatial resolution subsampling of an image, temporal subsampling can also be used to reduce the bit rate for transmitting video through limited bandwidth channels. Temporal subsampling means dropping certain intermediate frames in a video sequence. For very low bit rate applications, instead of 30, 15 or 10 or 7.5 frames may be transmitted per second. Reducing the temporal frequency by two however does not halve the bit rate, because temporal decimation introduces longer motion vectors and higher residual errors and as a consequence, the bit requirement increases.
- Sometimes transmission bits can be reduced by skipping the residual-error-compensation information if it has no significant components.
- A large quantisation value reduces the magnitude of transform coefficients so that only relatively small bits are needed to represent them. This ensures better bit compression.
- Fractional motion estimation and compensation is the another way to reduce the effective bit rates and improve the image quality.

Standard Coding

The above mentioned procedures are the basic ways to reduce the video data. Besides this there are encoder technologies which enable the video data reduction significantly. Most video encoders comprise of the basic functionalities of prediction, transformation, quantisation, and entropy coding. There still exist considerable variations in the structure of the encoder and decoder (CODEC) arrangement. An arbitrary input frame is firstly subdivided into *macroblocks* (MBs), which generally correspond to a group of 16×16 nonoverlapping pixels in the original image. Each MB is then coded in either intra- or intermode determined by the block predictor with the help of the previously coded frames (i.e., frame memory). In intramode, no motion vector is generated, and thus the original (sometimes there are differences between the original and the neighbouring MB in the same frame) MB is transformed by *discrete cosine transformation* (DCT) 0. The DCT coefficients are then quantised (Q), reordered (zigzag scanned), and entropy-encoded using any efficient *variable length coding* (VLC) technique. Sometimes a *coding control* mechanism is used to control the bit rate by adjusting the quantisation value. In intermode, an MB is formed by MC prediction from one or more reference frames using ME; however, the prediction for each MB may be formed from one or two previous or forward frames (in time order) that have already been encoded and reconstructed. The prediction MB is subtracted from the current MB to produce a residual error MB which is then transformed using DCT and quantised to give a set of coefficients which are reordered (zigzag scanned) and entropy-encoded, using any VLC algorithm. The entropy-encoded coefficients, together with the side information required to decode the MB (such as the MB prediction mode, motion vector, and quantisation step size), form the compressed bit stream. Inverse quantisation and *inverse DCT* (IDCT) are used to form the reference frames which are stored in the *frame memory* for the encoder.

The decoder path uses the quantised MB coefficients in order to reconstruct a frame for encoding further MBs. At the decoder, the incoming compressed bit stream is disassembled and the data elements are entropy-decoded and reordered to produce quantised coefficients. These are rescaled and inverse-transformed to form the residual error of MB. The decoder then creates a prediction MB from the reference frame by incorporating the motion vector to its original MB (in current frame) position. Predicted MB is added to residual error to create the decoded MB.

While for special applications, some functional elements are modified or additional blocks are included, the basic structure of a video CODEC remains the same. Examples of some of the additional components include a preprocessing filter (to reduce the noise introduced in capturing images from low-quality sources, or camera shake) and a postprocessing filter (to reduce the blocking and/or ringing effects) 0. These additional components enhance the performance in certain cases at the expense of increased hardware complexity.

H.264

The most recently advanced video coding standard, H.264/AVC 0, has been recently finalised to support a wide range of applications by including new flexible features to merge the concepts of MPEG-X and H.26X. The applications of H.264/AVC include: i) broadcast over cable, satellite, and cable modem; ii) interactive or serial storage on optical or magnetic storage; iii) conversational services over ISDN, LAN, and wireless mobile networks; and iv) video-on-demand.

A profile is defined as a set of coding functions and specifications that is required by an encoder or decoder which complies with that profile. H.264 supports three profiles: *baseline*, *main*, and *extended*. The baseline profile is used in low bit rate video coding applications such as telemedicine, video telephony, videoconferencing, and wireless communication. The baseline profile of H.264 supports two types of frame, namely intraframes and predicted frames.

The overall steps of H.264 video coding can be divided into several steps, such as motion estimation and compensation 0, transform, quantisation, and entropy coding. Intermacroblock prediction creates a predicted macroblock of the current block from one or more previously encoded video pictures through motion estimation. H.264 considers each MB as either skipped or nonskipped MB. The MB with no motion or little motion is considered as skipped MB and no motion vectors or residual errors are needed as it will be copied from the reference frame directly. Each nonskipped MB (16×16) may be divided four ways, and motion estimation and compensation are carried out either as one 16×16 block, two 16×8, two 8×16, or four 8×8 blocks. If the 8×8 mode is selected, each of the four 8×8 sub-MBs within the MB may be further divided four ways, either as one 8×8 sub-MB, two 8×4 sub-MBs, two 4×8 sub-MBs, or four 4×4 sub-MBs. The motion estimation mode is selected based on the minimum value of the Langragian cost function. The Langrangian cost function is defined by the total bits needed to encode the MB and the sum of square differences between the original MB and the reconstructed MB multiplied by the Langrangian multiplier 00. After motion estimation, DCT is applied on the difference between the original MB and best matched MB (based on the motion) in reference frame. Unlike H.263, H.264 used 4×4 integer transformation 0 instead of 8×8 noninteger DCT transformation. Transform coefficients are quantised for more compression. H.264 used 52 levels of quantisation where H.263 used only 31 levels. Transform coefficients are entropy-coded using a context-based adaptive variable length coding (CAVLC) 0 but not context-adaptive-based arithmetic coding (CABAC) 0. CABAC improves coding efficiency by about 10% over that obtained from the CAVLC, but it requires a larger circuit scale and more power. In applications like mobile phones and digital cameras where minimising power consumption is a prime consideration, the baseline profile is used. All other syntax elements are coded using fixed-length or Exp-Golomb 0 variable length codes.

Up to the H.263 standard, all intra-MBs are transformed by DCT and then entropy-coded, in the case of H.264, however, intramode prediction is introduced which forms a prediction block from the differences between the current block and the previously encoded and reconstructed blocks. Although this strategy involves a huge amount of computational time, it improves the performance in terms of bit rate. There are nine selection modes for an intrablock of size 4×4 and four selection modes for an intrablock of size 16×16.

Existing VLBR Schemes

Since 1980, when the VLBR concept first emerged, vector quantisation 00000 has been a serious competitor among the VLBR video coding techniques. Vector quantisation (VQ) is a nonstandard video coding technique, but is very effective for data compression as it seeks to exploit the correlation between components within a vector. Optimum coding efficiency can be achievable if the vector dimension is infinite so the correlation between all components is exploited. VQ is comprised of i) vector formation, ii) training set generation, iii) codebook generation, and iv) quantisation. Vector formation is the decomposition of images into a set of 2-D vectors like the MB, which can be considered as a set of 2-D vectors. Coding efficiency of VQ is highly dependent on choosing the best training set, which is selected from either the image, or statistically similar images. Codebook generation is the most important process in VQ, since coding efficiency will be optimal when the interrelations between the codewords in a codebook are minimised. Different criteria can be applied such that the input vector source is classified into a predefined number of regions by the minimum distance rule between intracodewords and the maximum distance rule between intercodewords. Quantisation selects the most appropriate codeword in the codebook for each input vector using some prescribed metric such as mean square error. An exhaustive search process over the entire codebook provides the optimal result, but is time-consuming. There are alternative search algorithms such as tree-search, which although suboptimal, are much faster.

To achieve high video data compression, quantisation becomes the pivotal element in the CODEC process. An ideal VQ approach based on a combination of variable vector size, multistages, dynamic codebook updating using locality, and parallel computing structures together with small codebook size, could theoretically prove a very strong competitor for any contemporary digital video coding standard. Pragmatically, however, it is not feasible to incorporate all the aforementioned properties because many have individual trade-offs. One major problem with VQ is that it does not reconstruct edge vectors efficiently as the codebook is unable to reproduce all possible patterns. VQ with a dynamically updated codebook based upon locality provides a good approximation of subimages but often requires a large number of bits due to high codebook transmission frequency to the decoder. Generally a VQ coding system requires preprocessing for vector and codebook formation as well as the codebook transmission overhead. Codebook searching time also takes a significant amount of time and these limitations ultimately restrict the range of applications of VQ.

Content-based coding for VLBR is a fundamental component of the MPEG-4 0 video-coding standard, although the concept is not exactly new. *Model-based coding* (MBC), for example, was first introduced in 1981 by Wallis, Pratt, and Plotkin 0 and represents a special kind of object-based coding. Applications of MBC 0–0, however, have tended to be restricted to video telephony and conferencing, where only one or two objects are considered and some *a priori* knowledge about a scene's content exists. In contrast to the conventional digital video coding standards that are based on eliminating spatial and temporal redundancies in a sequence, MBC treats images as two-dimensional

(2-D) projections of a 3-D world involving *a priori* knowledge of a scene's contents. One or more moving objects in a video sequence is analysed using computer vision techniques to create a parametric model incorporating key information concerning the size, location, and motion of these objects. At the decoder, the model synthesises each object, by using computer-graphical methods, with automatic tracking techniques enabling the model to mimic the respective objects' movements. The parameters needed to animate the model are then coded and transmitted to the receiver, which reconstructs the model. For low quality images, the animation data are sufficient to give a good approximation to the original image sequence, but for higher image quality, an additional residual error signal is required that typically comprises the coded frame differences between the original video sequence and the animated model. The bit rate performance of MBC is very good because only the model parameters are transmitted, and this has attracted attention as it provides high-quality images at VLBR applications. As a consequence, the MBC has been viewed as a potential competitor for MPEG-4 and H.264, though major practical problems remain to be solved, namely the difficulty in modelling unknown objects and the inevitable presence of analysis errors.

In the context of low bit rate, the video coding wavelet theory has demonstrated an ability to not only provide high coding efficiency, but also spatial and quality scalability features. Grossman and Morlet first introduced the wavelet transform in 1984 0 by mapping a time or spatial function into a two-dimensional function. The main advantages of the discrete wavelet transformation (DWT)-based video coding 00 are: i) DWT has high decorrelation and energy compaction efficiency; ii) the wavelet basis functions match well with the human visual system (HVS) characteristics; iii) blocking artefacts and perceptual distortion are far less visible in wavelet filters due to the spatially global decomposition, resulting in subjectively better reconstructed images; iv) the DWT allows multiple resolution analysis that supports high scalability since wavelet coefficient data structures are spatially self-similar across subbands; and v) the number of image pixels and DWT coefficients are the same, so there is no information is lost. On the other hand, due to the following limitation, it is included in recent video coding standards. The limitations are: i) DWT requires more memory and processing time because global decomposition requires the whole image to be considered as a large size block; ii) computational complexity is relatively high compared to discrete cosine transformation; iii) due to the large block size, efficient coding specially in VLBR is often impossible because it cannot differentiate active from static regions; and iv) there is no standard ways to incorporate the motion information using large block size.

Though the above mentioned techniques are good competitors for the standard video coding technique, they have their own limitation in real-time video coding. Moreover, they cannot work under the existing video coding standard frameworks.

MAIN FOCUS OF THE CHAPTER

Block-based H.261/3/4 000 and block-/content-based MPEG-4 0, standards have already the VLBR video coding option. The H.261 and H.263 standards are, however, unable to efficiently encode the boundary-adjoined part of a moving object within a 16×16 pixel *macroblock* (MB) during motion estimation, resulting in all 256 residual error values being transmitted for motion compensation regardless of whether there are moving objects or not. Efficient encoding of these blocks needs to eliminate the *intrablock temporal redundancy* (ITR), because they are almost static in the successive frames. None of the block-based standards, however, are able to exploit the ITR in the form of static background within the MB. To remove this inefficiency, the MPEG-4 video standard first introduced the concept of content-based coding by dividing video frames into separate segments (instead of MBs), comprising a background and one or more arbitrary-shaped moving objects that are coded separately. As this process requires expensive segmentation and shape coding, and is also ineffective for real-world video objects, it is not suitable for low processing devices using VLBR applications.

One solution in exploiting ITR is to subdivide the MB and then apply ME and MC to each subblock. With sufficient numbers of subblocks, the shape of a moving object can be more accurately represented. This solution has been implemented in the recent H.264 standard using the *variable block size* (VBS) mode. It, however, requires not only bits overhead for the motion vector and VBS mode for each partition, but also higher computational complexity in order to identify the best partitioning. Obviously, the smaller the subblocks, the higher these overheads will be and that could potentially offset all the coding efficiency resulting from better moving object shape approximation. As a result, VLBR coding using H.264 avoids smaller subblocks, and thus makes its VBS mode ineffective.

The MPEG-4 video standard exploits intrablock temporal redundancy by dividing video frames into separate segments comprising of a background and one or more moving objects. Paradoxically, it also depends on computationally expensive segmentation and shape coding and is not suitable for low processing devices, especially nonsynthetic real-world objects.

An alternative approach was proposed by Fukuhara, Asai, and Murakami 0 who used four MB-partitioning patterns each comprising of 128-pixels. ME and MC was carried out on all eight possible 128-pixel partitions of an MB and the pattern with the lowest prediction error was selected. While this approach gives better performance compared to H.263, the computational complexity of the motion-based processing is too high for real-time applications and having only four patterns means that it is insufficient to represent moving objects 0. As well, treating each MB, irrespective of its motion content, also resulted in a higher bit-rate being incurred for those MBs which contained only static background or had moving object(s), with little static background. In such cases, the motion vectors for both partitions were almost the same and so only one could be represented.

To address the limitations of Fukuhara et al.'s approach, Wong, Lam, and Siu 0 and Paul and Murshed 000 exploited the idea of partitioning the MBs via a simplified segmentation process that again avoided handling the exact shape of the moving objects, so that popular MB-based motion estimation techniques could be applied. This *pattern-based video coding* (PVC) algorithm focused on the moving regions of the MBs, through the use of a set of regular n ($n < 256$) pixel pattern templates, from a pattern codebook (PC). If in using some *similarity* measure, the *moving region* (MR) of an MB is well covered by a particular pattern, then the MB can be coded by considering only the n pixels of that pattern, with the remaining 256-n pixels being skipped as *static background*. Successful pattern matching can therefore, theoretically, have a maximum compression ratio of 256/n:1 for any MB. The actual achievable compression ratio will be lower due to object occlusion and the computing overheads for handling an additional MB type, pattern identification numbering, and pattern matching errors. This approach is radically different from H.264 subblocking as ME and MC are carried out only for the selected patterns, thus keeping the computational complexity in check.

FUTURE TRENDS

The future trend is to contribute to the acceptance of the concept of pattern based coding (PBC) as a potential additional mode for future VLBR video coding standards through a series of innovations in the form of pattern codebook construction, real time pattern selection, pattern matching criteria, coding scalability, and optimality issues so that the overall compression gain will be achieved in relation to the conventional block-based coding techniques as well as the existing PBC. The future research, thus, focuses on the following major areas:

To extend the regular-shaped pattern codebook for better shape approximation of the moving region; to improve moving region definition in a quantitative manner for adaptability; to enhance pattern matching criterion to reduce computational complexity and ensure better coding efficiency; and to develop a variable length pattern identification coding scheme to reduce side information overhead.

To develop a real time pattern selection algorithm by intuitively engaging a low-complexity low-accuracy pattern matching criterion to restrict the amount of high-complexity accurate pattern matching per MB within a user-selectable bounded subset of the pattern codebook, and to modify the calculation strategy of the accurate pattern matching criteria to reject patterns at intermediate steps resulting in further computational complexity improvement.

To develop a feasible-sized pattern set selection technique in order to limit pattern identification overhead by pruning the universal set of all possible patterns through some content-based analysis within the regularity constraints for maximising the efficiency of the moving regions representation using the selected set as the pattern codebook.

To extend the concept of PBC to the content-based coding paradigm by developing an efficient heuristic to generate arbitrary-shaped patterns from the video content, and to investigate whether a variable pattern size mode, similar to the VBS mode in H.264, can be introduced with PBC to address its scalability issue.

Pattern Codebook

In this approach, a number of regular/irregular shaped *n*-pixel patterns are used. Patterns are used as an intermediate processing tool to get the moving regions from a binary matrix created from the current and reference frames based on their relative pixel intensity changes. Each pattern is defined as a 16×16 block where the white region represents '1' (i.e., motion) and the black region represents '0' (i.e., background). These patterns are selected intuitively based on the following two features: i) as a moving region covers part of an object, the region must start from the edge of the boundary, and ii) the moving region must be a convex polygon so that it is simple and regular.

Using a small number of patterns cannot approximate the shape of the MR in MB for all kinds of moving objects, resulting in many active-region MBs (RMBs) potentially being neglected, as moving regions vary widely between objects. If the codebook size is extended, however, the number of RMBs will increase and the image quality will improve as the residual error is reduced, although, there will be a corresponding increase in the number of pattern identification bits for each RMB. Any improvement in managing the pattern identification bits will accommodate more patterns in code book. So far 8 to 32 patterns are used in pattern-based video coding. To extend the PC size certain features were assumed for each *n*-pixel patterns. Each would be *regular* (bounded by straight lines), *clustered* (the pixels would be connected), and *boundary-adjoined.* Since the MR of an MB is normally a part of a rigid object, assuming clustered and boundary-adjoined features for a pattern is quite justifiable. The regularity feature is added to limit the pattern codebook size.

When applied to 16×16 pixels MBs and *n*-pixel patterns, an astronomically high number of ${}^{256}C_n$ possible patterns can result. However, it is possible to select a feasible size of codebook from the universal set in the following extremes. At one extreme, arbitrary-shaped patterns can be generated through video content analysis. At the other extreme, the universal set can be pruned to obtain a set of regular-shaped patterns of a feasible size. A two-stage pruning mechanism where the universal set is first reduced to a size, say α, by using regularity constraints on the pattern shape, and then the set is further reduced using an iterative greedy approach to attain a subset of the λ ($< \alpha$) best-matched pattern set based on the video content. A *content-based pattern generation* algorithm may be developed which does not assume any specific feature among the generated patterns. Obviously, such generated patterns would not only exhibit close conformation to the moving-region-defining *clustered* and *boundary-adjoined* features but would also approximate the shape of the region more closely. This would then lead to improved rate-distortion performance of *pattern-based video coding*. This approach also allows for the introduction of a *variable pattern size* (VPZ) mode similar to the *variable block size* (VBS) mode of H.264. This is because the additional rate-distortion improvement resulting from arbitrary-shaped pattern generation can outweigh the coding overhead required to accommodate the pattern size identifiers.

Pattern identification code (PIC) is a vital issue especially when the PC size is large, as mentioned in the earlier section. Fixed length 3-bit PICs to distinguish eight patterns is the obvious choice. Using an efficient variable length coding scheme, for example, the Huffman and arithmetic coding schemes, can reduce the effective number of bits used per PIC. VLC for all the patterns are calculated using the Huffman coding from the average RMB frequency captured by each pattern over a large number of standard and nonstandard video sequences. In this way 04.62 bits are required instead of five bits for 32 patterns. Using the co-occurrence matrix of patterns can reduce PIC up to 0.85 bits per patterns. Further research is necessary to reduce PIC bits.

MB Classification

The MB classification is also a crucial to find the MBs which would be presented by pattern templates. In implementing the RMB category, an MB is considered a *candidate* RMB (CRMB) if at least one of the four 8×8 quadrants does not contain moving pixels. This classification may, in certain instances, reduce the number of RMBs by misclassifying a possible CRMB as an active MB (AMB), where only a few moving pixels exist in another quadrant. Conversely, it may also increase computational complexity by misclassifying an AMB as a CRMB where all but one quadrant has many moving pixels. Ultimately, a CRMB is classified as an RMB depending on a *similarity measure* based on the patterns in the codebook.

A new and more efficient parametric ($64 < \delta < 256$) MB classification may be used 0, where δ is the maximum total number of moving pixels in a MB to be a RMB, without regard to any 8×8 quadrant. The justification of the lower and upper limit is that the pattern size is 64 and total maximum moving pixels are 256. The experimental results show that the actual value of δ is within ($64 < \delta < T_S + 64$)under a similarity threshold. If δ is greater than $T_S + 64$, then no CRMBs will be classified as RMBs; on the other hand, if δ is less than 64, then some CRMBs will not be classified as RMBs although they are good candidates. This definition considers the number of '1's in an MB irrespective of their position in any specific quadrant. Moreover, this definition introduces a parameter, δ, which controls the number of RMBs. The corollary is that parameter δ directly contributes to both overall quality and compression.

Similarity Measure

Empirical results confirmed that between 6% and 29% of the total MBs are classified as RMBs in smooth motion video sequences for any PBC. The similarity metric, however, is applied much more often as the number of CRMBs will always be higher than the RMBs. Motion estimation, irrespective of a scene's complexity, typically comprises more than 60% of the processing overhead required to encode a predicted-frame with a software codec using the DCT, when a full search is used. Similarity metric calculation is also an expensive part of ME in any PBC algorithm. The corollary of this is that the computational efficiency of a similarity metric for a CRMB is critical to the overall complexity, since, for example, for a 32 PC size, the metric represents about 55% of the ME time of a RMB. Hence, any strategy that improves the computational efficiency of the metric concomitantly reduces the overall encoding complexity.

A similarity measure can be defined in two ways. Pattern-included approach considers both the mismatch areas between a pattern template and moving region. Pattern-excluded approach 0 considers only the mismatched area of moving regions. In both approaches a pattern is selected as the best-matched pattern for a given moving region if this measure is minimum. The advantage of the later approach is that it has better control in rate-distortion curves using the similarity threshold. Moreover, it reduces the computational time because it only requires calculating the mismatched area of moving regions instead of both. Obviously a large similarity threshold will select more numbers of RMBs and as a result reduce the bit rate with decreasing the image quality.

Relevant Measure

Measuring the similarity between a CRMB and all the patterns in the codebook on a piecewise-pixel basis can be very computationally expensive, especially when the PC size is large, which is always desirable for better coding efficiency. However, it can be easily observed that not all patterns are *relevant* for consideration when using the similarity measure. A gravitational centre proximity-based *pattern relevance* measure is proposed in 0 to dynamically create a smaller-sized *customised PC* (CPC) for each CRMB, by eliminating irrelevant patterns from the original codebook. This algorithm selects the best pattern for a CRMB from the CPC, using a piecewise-pixel similarity measure. The rationale in using both relevance and similarity metrics to select the best pattern for a CRMB is that it provides a facility to trade off between computational complexity and picture quality. In selecting the best pattern, the relevance metric uses only one point, say, *gravitational centre* (GC), to represent all moving pixels in a CRMB, whereas the similarity metric uses all pixels; so there will be an error between the two metrics. However, the relevance metric requires 18 times less add-equivalent operations compared with the similarity metric. The algorithm uses a novel mechanism to control the size of the CPC within predefined bounds to adapt the computational complexity of the pattern selection process, ensuring real time operation without compromising image quality. Furthermore, the computational overhead of the similarity metric is reduced significantly by performing the processing on a quadrant-by-quadrant basis with the option to terminate whenever the measure exceeds a predefined threshold value.

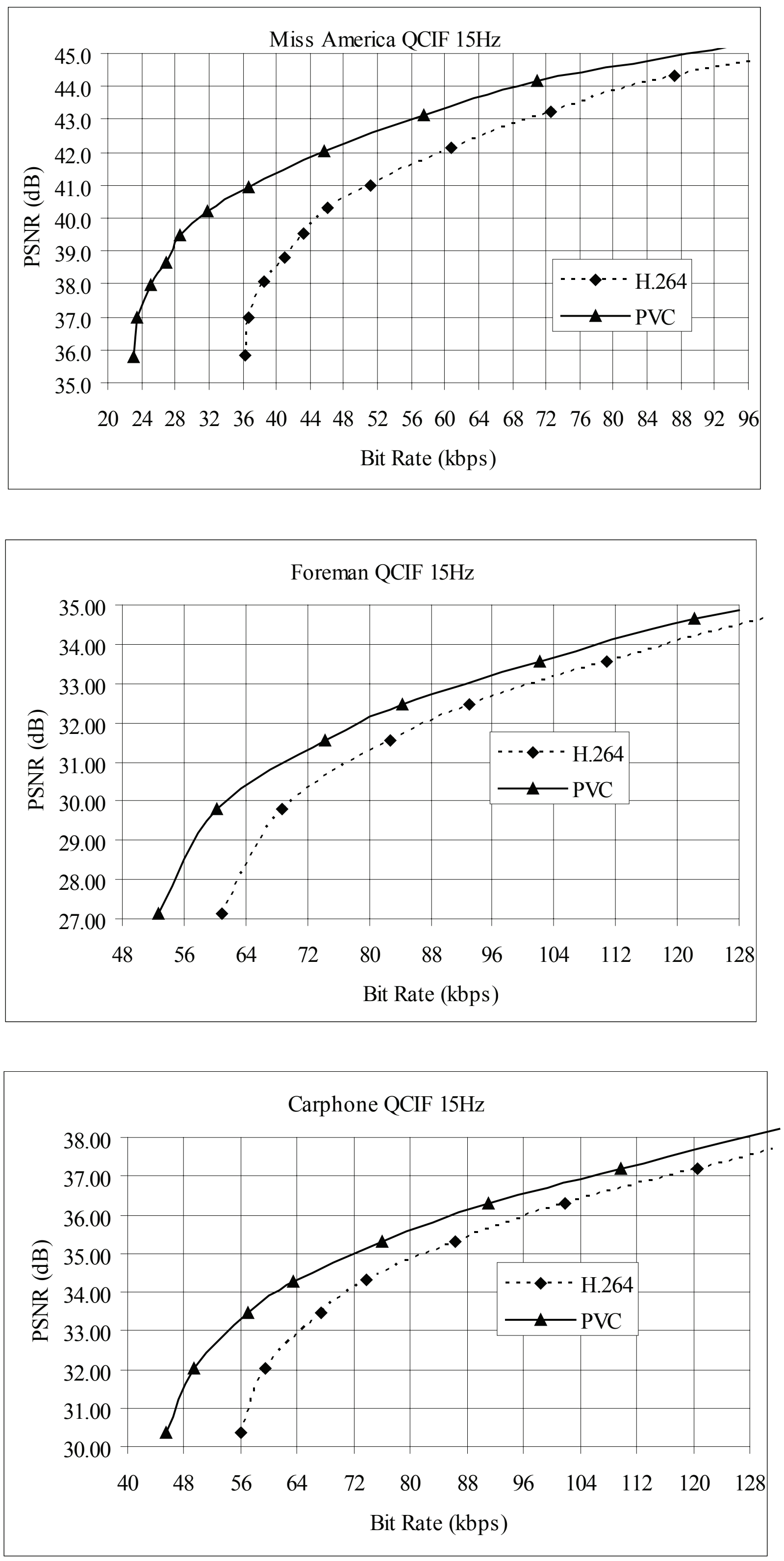
This arrangement ensures the algorithm always uses the complete codebook at some stage of the pattern selection process and still manages to keep the computational complexity within real-time constraints. This principal can be easily extended to arbitrarily sized pattern codebooks. The computational efficiency of the similarity measure is significantly improved by using a predefined threshold and computing the metric on a quadrant-by-quadrant basis.

Embedding Issues with Existing H.264

To analysis the performance of pattern-based video coding, we cab easily observe that the moving region covered by the best-matched pattern template provides relatively less bits with better image quality; on the contrary, an uncovered moving region provides poor image quality. At a very low bit rate, a large scale distortion occurs, thus image distortion due to the uncovered moving region is negligible compared to the high distortion in overall image. As a result, pattern-based video coding algorithm outperforms the H.264 standard for very low bit rate range. When the target bit rate is high, the distortion in an uncovered moving region is relatively high compared to overall image distortion. As a result the rate-distortion performance of pattern-based coding diminishes with bit rates compared to the H.264.

To address this problem Paul et al. 0 proposed two ways. It considers pattern-based coding as a mode, that is, selected MB will be processed using pattern and other variable block size modes, then it will pick

Figure 1. Rate-distortion performance using H.264 standard and after embedding PVC as an extra mode in H.264 for three standard video sequences



that mode which provides the best rate-distortion performance. The reason behind this approach is that when a pattern can not provide the best rate-distortion performance, the encoder automatically selects the best mode among the variable blocks so that it ensures the performance better or at least the same as the H.264 does. It considers two large pattern sets with larger variable block size modes and one small size pattern set with small modes to ensure the approximation of the variable size moving regions more accurately.

The experimental results (see Figure 1) confirmed that this new scheme improves as high as 1.5dB image quality compared to the H.264 standard at the same bit rates. There is still a scope to generate a best pattern set which can further improve the rate-distortion performance. Since pattern-based video coder as an extra mode has improved image quality significantly, less effort will be needed to include this in the existing framework of H.264; we are hopeful about the inclusion of patter-based video coder in future H.264 video coder versions.

CONCLUSION

Pattern-based video coding with efficient MB classification, optimal pattern codebook, novel similarity and relevant measurement, and successful inclusion in H.264 as a mode, has outperformed the existing H.264 video coding standard by 0.5dB to 2dB in low to mid range bit rates. Moreover, it also outperformed the H.264 in terms of computational complexity. Still it has some scopes to improve further. There is no optimal pattern set for all kind of video sequences. Arbitrary shape pattern templates may be a solution but it can not work in true real time. Further research is needed to make an arbitrary shaped pattern-based video coding in real time application. Pattern identification code is another interesting area. The bits for pattern identification code will help to retrieve the exact pattern in decoder but increase the bit rates. If we can retrieve the exact pattern in decoder from the side information without those bits, eventually this reduction of bits will improve the rate-distortion performance. Applications of pattern-based video coding in distributed video coding concepts and application of distributed video coding concepts in pattern-based video coding approaches may be two other potential areas for future research.

REFERENCES

Ahmed, N., Nararajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE Transaction on Computer,* 90-93.

Ardre, T., Antonini, M., & Barlaud, M. (2005). *Puzzle temporal lifting for wavelet-based video coding.* Paper presented at the IEEE International Conference on Image Processing (Vol. 3, pp. 213-216).

Aron, A., & Girod, B. (2003). Towards practical wyner ziv coding of video. In *Proceedings of the IEEE International Conference on Image Processing.*

Asif, A., & Kouras, M. (2006). Scalable video codec by noncausal prediction, cascaded vector quantization, and conditional replenishment. *IEEE Transaction on Multimedia*, *8*(1), 19-31.

Balter, R., Gioia, P., Morin, L., & Galpin, F. (2004). *Scalable and efficient coding of 3D model extracted from a video.* Paper presented at the International Symposium on 3D Data Processing, Visualization and Transmission (pp. 836-843).

Bashshur, R., & Lovett, J. (1977). Assessment of telemedicine: Results of the initial experience. *Aviation Space and Environmental Medicine*, *48*(1), 65-70.

Bjontegaard, G., & Lillevoid, K. (2002). *Context-adaptive VLC coding of coefficients.* Paper presented at the 3rd Meeting of the JVT document JVT-028, JVT of ISO MPEG & ITU VCEG, Rairfax, VA.

Docef, A., Kossentini, F., Nguuyen-Phi, K., & Ismail, I. R. (2002). The quantized DCT and its application to DCT-based video coding. *IEEE Transaction on Image Processing*, *11*(3), 177-187.

Fukuhara, T., Asai, K., & Murakami, T. (1997). Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories. *IEEE Transaction on Circuits and Systems for Video Technology*, *7*, 212-220.

Golomb, S. W. (1966). Run-length encoding. *IEEE Transactions on Information Theory*, *12*, 399-401.

Grossman, A., & Morlet, J. (1984). Decompositions of hard functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal*, *15*(4), 723-736.

Halsall, F. (2001). *Multimedia communications: Applications, networks, protocols and standards.* Addison-Wesley.

Hu, M., Worrall, S., Sadka, A. H., & Kondoz, A. M. (2002). Model design for scalable two-dimensional model-based video coding. *Electronics Letter*, *38*(24), 1513-1515.

Isechi, A., Oda, C., Shinkura, R., Akiba, S., Fujikawa, H., & Yamazaki, K. (2004). *Experiment of Internet-based tele-medicine in Amami Rural Islands.* Paper presented at the IEEE International Symposium on Applications and the Internet Workshops.

ISO/IEC 14496. MPEG-4. (1998). *Information technology-coding of audio-visual objects.*

ITU-T Rec. H.264/ISO/IEC 14496-10 AVC. Joint Video Team (JVT) of ISO MPEG and ITU-T VCEG, JVT-G050 (2003).

ITU-T Recommendation H.261. (1993). *Video CODEC for audiovisual services at p× 64kbits/s.*

ITU-T Recommendation H.263. (2000). *Video coding for low bit-rate communication* (version 3).

Jun, S., & Yu, S. (2001). Efficient method for early detection of all-zero DCT coefficients. *Electronics Letters*, *37*(3), 160-161.

Karayiannis, N. B., & Li, Y. (2001). A replenishment technique for low bit-rate video compression based on wavelets and vector quantization. *IEEE Transaction on Circuits and Systems for Video Technology*, *11*(5), 658-663.

Liu, B. (2005). Detection of region of interest in neurosurgical video used for telemedicine. In *Proceedings of IEEE International Conference of Information Acquisition*.

Lin, C-H., & Wu, J.-L. (1997). Content-based rate control scheme for very low bit-rate video coding. *IEEE Transaction on Consumer Electronics, 43*(2), 123-133.

Man, H., Queiroz, R. L., & Smith, M. J. T. (2002). Three-dimensional sub-band coding techniques for wireless video communications. *IEEE Transaction on Circuits and Systems for Video Technology*, *12*(6), 386-397.

Marpe, D., Schwarz, H., & Weigand, T. (2003). Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transaction on Circuits and Systems for Video Technology*, *13*(7), 620-636.

Pan, J. S., Lu, Z. M., & Sun, S. H. (2003). An efficient encoding algorithm for vector quantization based sub-vector technique. *IEEE Transaction on Image Processing*, *12*(3), 265-270.

Paul, M., Frater, M., & Arnold, J. (2006). *Efficient mode selection algorithm using image distortion for H.264 video encoder*. Paper presented at the IEEE International Conference on Signal Processing (Vol. 2, pp. 16-20).

Paul, M., & Murshed, M. (2005a). Advanced very low bit rate video coding using preferential pattern selection algorithms. *Journal of Research and Practice in Information Technology*, *37*(2), 89-99.

Paul, M., & Murshed, M. (2005b). An efficient similarity metric for pattern-based very low bit-rate video coding. *Journal of Internet Technology*, 6(3), 337-344.

Paul, M., & Murshed, M. (2007). *Efficient H.264/AVC video encoder where pattern is used as extra mode for wide range of video coding.* Paper presented at the International Multimedia Modeling Conference, Lecture Notes in Computer Science, Springer-Verlag, Heidelberg (Vol. 4352, pp. 353-362).

Paul, M., Murshed, M., & Dooley, L. (2002). *Impact of macroblock classification on low bit rate video coding focusing on moving region.* Paper presented at the International Conference on Computer and Information Technology (pp. 465-470).

Paul, M., Murshed, M., & Dooley, L. (2005). A real-time pattern selection algorithm for very low bit-rate video coding using relevance and similarity metrics. *IEEE Transaction on Circuits and Systems for Video Technology, 15*(6), 753-761.

Puri, R., & Ramchandran, K. (2002). PRISM: A new robust video coding based on distributed compression principles. In *Proc. of the Conf. on Comm., Control, and Computing*, Allenton, IL.

Richardson, I. E. G. (2002). *Video codec design*. John Willey & Sons, Ltd Press.

Richardson, I. E. G. (2003). *H.264 and MPEG-4 video compression.* Wiley Press.

Shen, G., Zeng, B., & Liou, M. L. (2003). Adaptive vector quantization with codebook updating based on locality and history. *IEEE Transaction on Image Processing, 12*(3), 283-293.

Siu, M., Chan, Y. H., & Siu, W. C. (2001). A robust model generation technique for model-based video coding. *IEEE Transaction Circuits Systems for Video Technology*, *11*(11), 1188-1192.

Slepian, J. D., & Wolf, J. K. (1973). Noiseless coding of correlated information sources. *IEEE Transaction on Information Theory*, *19*, 471-490.

Sorwar, G., Murshed, M., & Dooley, L. (2003). *A fully adaptive performance-scalable distance-dependent thresholding search algorithm for video coding.* Paper presented at the IEEE International Conference on Speech, Acoustics, and Signal Processing (Vol. 3, pp. 649-652).

Sun, M., Lie, Q., Xu, J., Liu, B., Hu, C., Kassam, A., et al. (2004). A multimedia system for remote neurosurgical monitoring. In *Proceedings of International Symposium on Intelligent Signal Processing and Communication System* (pp. 379-383).

Wallis, R. K. Pratt, W. K., & Plotkin, M. (1981). Video conferencing at 9600 bps. In *Proceedings of the Picture Coding Symposium* (pp. 104-105).

Wang, D., Zhang, L., & Vincent, A., (2006). New method for reducing GOP-boundary artifacts in wavelet-based video coding. *IEEE Transaction on Broadcasting*, *52*(3), 350-355.

Weigrand, T., Schwarz, H., Joch, A., & Kossentini, F. (2003). Rate-contrained coder control and comparison of video coding standards. *IEEE Transaction on Circuits and Systems for Video Technology*, *13*(7), 688-702.

×Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol, 13*(7), 560-576.

Wong, K.-W., Lam, K.-M., & Siu, W.-C. (2001). An efficient low bit-rate video-coding algorithm focusing on moving regions. *IEEE Transaction on Circuits and Systems for Video Technology*, *11*(10), 1128-1134.

Wyner, D., & Ziv, J. (1976). The rate-distortion functions for source coding with side information at the decoder. *IEEE Transaction on Information Theory*, *22*, 1-10.

Xu, J., Sclabassi, R. J., Liu, Q., Hu, C., Chaparro, L. F., & Sun, M. (2005). A region-based video coding method for remote monitoring of neurosurgery. In *Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference* (pp. 89-91).

Xuan, Z., Henghua, Y., & Songyu, Y. (1998). Method for detecting all-zero DCT coefficients ahead of discrete cosine transformation and quantization. *Electronics Letters*, *34*(19), 1839-1840.

Yokoyama, Y., Miyamoto, Y., & Ohta, M. (1995). Very low bit-rate video coding using arbitrarily shaped region-based motion compensation. *IEEE Transaction on Circuits and Systems for Video Technology*, *5*(6), 500-507.

Zach, S. (1996). *Telemedicine overview and summary.* Paper presented at the 19th Convention of Electrical and Electronics Engineers in Israel (pp. 409-412).

KEY TERMS

Context Adaptive Based Arithmetic Coding (CABAC): This variable length coding technique is used to compress the data using arithmetic variable length coding.

Context-Based Adaptive Variable Length Coding (CAVLC): This variable length coding technique is used to compress the data using relevant context side information.

Discrete Cosine Transformation (DCT): DCT compresses the spatial image data, such that the total energy in the image becomes concentrated into a relatively small number of components, that is, the pixel data are decorrelated so compression can be achieved.

Discrete Wavelet Transformation (DWT): DWT compresses the spatial image data, such that the total energy in the image becomes concentrated into a relatively small number of components, that is, the pixel data are decorrelated so compression can be achieved.

Group of Picture (GOP): A collection of frames which are processed at a time to reduce the temporal redundancy. For more compression and low image quality a relatively large GOP size is used and for low compression and better image quality a relatively small GOP is used in video coding.

Macroblock (MB): Nonoverlapping 16×16 pixels in an image which normally is used as a processing unit in video coding.

Motion Estimation and Motion Compensation (ME and MC): ME involves identifying the translational displacement vectors (popularly called *motion vectors*) of objects based on the changes between two successive frames in a video sequence. MC involves calculating the differential signal (residual error) between the intensity value of the pixels in the moving areas and their counterparts in the reference frame, translated by the estimated motion vector.

Variable Length Coding (VLC): A reversible procedure for entropy coding that assigns shorter bit strings to symbols expected to be more frequent and longer bit strings to symbols expected to be less frequent.

Vector Quantisation (VQ): A vector quantisation matrix typically has small values in top-left elements of the matrix and larger values in the bottom-right, retaining low frequency information at the expense of higher frequencies.

Very Low bit Rate Video Coding (VLBR): VLBR broadly encompasses video coding which mandates a temporal frequency of 10 frames per second (fps) or less, and bit rates between 8 and 64Kilo bit rate per second, facilitating video communications over mobile and fixed telephone channel transmissions as well as the Internet.