

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
5 February 2004 (05.02.2004)

PCT

(10) International Publication Number
WO 2004/012458 A2

(51) International Patent Classification⁷: **H04N 7/26**

(21) International Application Number:
PCT/SG2002/000168

(22) International Filing Date: 26 July 2002 (26.07.2002)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):
NANYANG TECHNOLOGICAL UNIVERSITY
[SG/SG]; Innovation and Technology Transfer Office, Blk 1,
Unit 213, 16 Nanyang Drive, Singapore 637722 (SG).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **YANG, Xiao, Kang**
[CN/SG]; Innovation and Technology Transfer Office, Blk 1,
Unit 213, 16 Nanyang Drive, Singapore 637722 (SG).
YU, Rong, Shan [CN/SG]; Innovation and Technology
Transfer Office, Blk 1, Unit 213, 16 Nanyang Drive,
Singapore 637722 (SG). **ZHU, Ce** [CN/SG]; Innova-
tion and Technology Transfer Off, Blk 1, Unit 213, 16

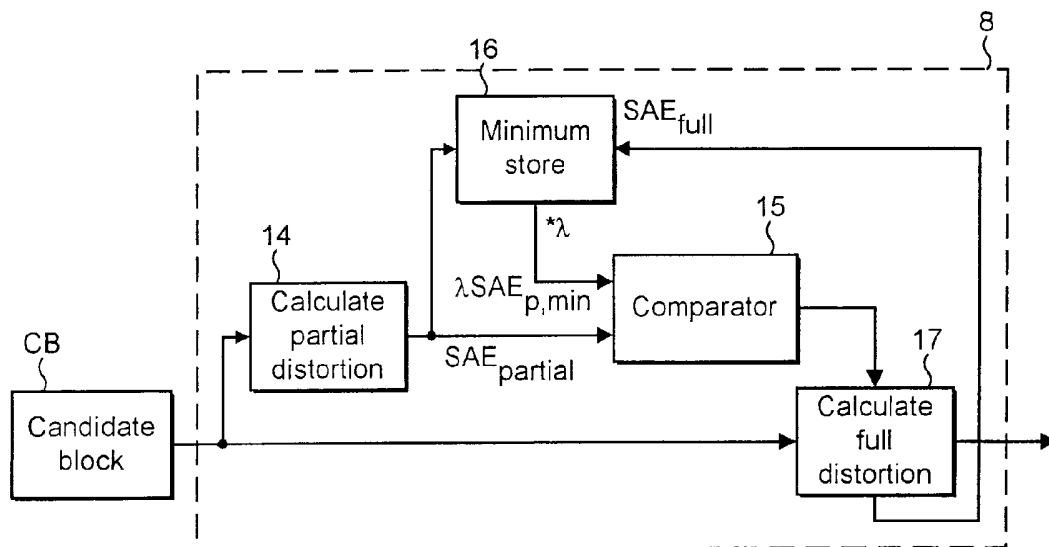
Nanyang Drive, Singapore 637722 (SG). **LIM, Keng, Pang** [SG/SG]; Innovation and Technology Transfer Office, Blk 1, Unit 213, 16 Nanyang Drive, Singapore 637722 (SG). **LI, Zheng, Guo** [CN/SG]; Innovation and Technology Transfer Office Blk 1, Unit 213 16 Nanyang Drive, Singapore 637722 (SG). **PAN, Feng** [SG/SG]; Innovation and Technology Transfer Office, Blk 1, Unit 213, 16 Nanyang Drive, Singapore 637722 (SG). **FENG, Ge, Nan** [CN/SG]; Innovation and Technology Transfer Office, Blk 1, Unit 213, 16 Nanyang Drive, Singapore 637722 (SG). **WU, Da, Jun** [CN/SG]; Innovation and Technology Transfer Office, Blk 1, Unit 213, 16 Nanyang Drive, Singapore 637722 (SG). **WU, Si** [CN/SG]; Innovation and Technology Transfer Office, Blk 1, Unit 213, 16 Nanyang Drive, Singapore 637722 (SG).

(74) Agent: **NAMAZIE, Farah**; Namazie & Co., Robinson Road Post Office, PO Box 1482, Singapore 902932 (SG).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,

[Continued on next page]

(54) Title: EFFICIENT VIDEO TRANSMISSION



(57) Abstract: A method and system for reducing the complexity of block motion estimation is described. According to the invention, the block distortion measure is determined by calculating a partial error result using a proportion of pixels in the block, comparing the partial error result with a threshold value and rejecting blocks where the partial error result is greater than the threshold value, and calculating a full error result for non-rejected blocks only. The threshold value is a function of the current minimum partial error result, and is preferably the current minimum partial error result multiplied by a threshold constant λ . Another aspect of the invention provides a video transport scheme which exploits the different importance of both GOP and re-synchronisation levels. According to this aspect, an unequal loss protection (ULP) philosophy is utilized. That is, the bit stream data is partitioned into several parts according to different importance, and the parts are unequally protected against packet loss using forward error correction codes.



LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Title of the InventionEFFICIENT VIDEO TRANSMISSIONField of the Invention

This invention relates to efficient video transmission.

Background to the Invention

In low bit-rate video coding, the technique of block motion estimation is widely adopted to improve the coding efficiency. The basic concept of block motion estimation can be described as follows. For each equal-sized pixel block in the current frame, we look for the block in the previously transmitted frame that is the closest to it, according to a predefined distortion criterion such as the Sum of Absolute Error (SAE) or Mean Square Error (MSE). This closest block is then used as a predictor for the present block. One straightforward algorithm to find the matching block is the Full Search (FS) algorithm, where all the candidate blocks inside a search window are checked. Although it gives optimal prediction performance, the computational complexity of FS algorithm is generally too high for a practical application.

Previously numerous search strategies have been proposed to reduce the computation complexity of the block motion estimation procedure. Most search strategies aim to reduce the computational complexity by checking only some of the points inside a search window. Typical examples of such approach include 3 Step Search (3SS) R. Li, B. Zeng and M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Tran. Circuits Syst. Video Technol.*, vol. 4, Aug., 1994, pp. 438-442, Four Step Search (FSS) L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Tran. Circuits Syst. Video Technol.*, vol. 6, pp. 313 – 317, June 1996, Diamond Search (DS) J. Y. Tham *et al*, "A novel unrestricted center-biased diamond search

algorithm for block motion estimation," *IEEE Tran. Circuits Syst. Video Technol.*, vol. 8, no. 4, Aug. 1998, pp. 369 – 377, and the Hexagon Search (HS) C. Zhu *et al*, "A novel hexagon-based search algorithm for fast block motion estimation," *Proc. ICASSP 2001*. Based on the assumption that all pixels in a block move by the same amount, another feasible approach to reduce the computation is to use only a fraction of the pixels in a block in calculating the block distortion measure (BDM). Bierling M. Bierling and R. Thoma, "Motion compensation field interpolation using a hierarchically structured displacement estimator," *Signal Processing*, vol. 11, no. 4, Dec. 1986, pp. 387 –404). M. Bierling, "Displacement estimation by hierarchical block matching", *SPIE, Visual Commun. Image Processing '99*, vol. 1001, 1988, pp. 942 – 951 introduced a hierarchical motion estimation technique in which an approximation of the motion field is obtained from lowpass-sub-sampled filtered images. T. Koga and *et al*, "Motion Compensated Interframe Coding for Video Conferencing," *Proc. Nat. Telecommun. Conf. 1981*, pp. 531 – 535 also proposed using 1:2 decimated images without low-pass filtering (see Figure 1a). Intuitively, too few pixels used in the procedure will eventually result in inaccurate estimation. For example, it is reported in B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors", *IEEE Tran. Circuits Syst. Video Technol.*, vol.3, No. 2, April 1993, pp. 148 – 157 that a 1:4 decimation ratio (see Figure 1b) will increase the error entropy by 0.1 bit/pixel and will be considered excessive. In that paper, Liu and Zaccarin proposed an efficient method based on pixel decimation that considered only one-fourth pixels of the total block. The technique is very effective for its quality performance. However, since all the positions in the search window should be considered, it cannot be applied to every search strategy. In Y. L. Chan and W. C. Siu, "A New Block Motion Vector Estimation Using Adaptive Pixel Decimation," *Proc. ICASSP 1995*, pp. 2257 – 2260, "New Adaptive Pixel Decimation for Block Motion Vector Estimation," *IEEE Tran. Circuits Syst. Video Technol.*, pp. 942 – 951, vol. 6, no. 1, pp. 113 – 118, 1996 and Y. L. Chan, W. L. Hui and W. C. Siu, "A Block Motion Vector Estimation Using Pattern Based Pixel Decimation," *Proc. IEEE International Symposium on Circuits and Systems 1997*, pp. 1153 – 1156 the authors proposed a sequence of

adaptive pixel decimation methods in which the patterns of selected pixels varied according to the value of the gradient of the luminance. Although those adaptive methods give slightly better results than Liu and Zaccarin, the test operations required by this adaptive approach prevent a global reduction of the computational complexity. Recently in C. Cheung and L. Po, "Normalized Partial Distortion Search Algorithm for Block Motion Estimation," *IEEE Tran. Circuits Syst. Video Technol.*, vol, 10, no. 3, pp. 417 – 422, 2000, a technique called Normalized Partial Distortion Search (NPDS) algorithm is proposed where a halfway-stop technique is introduced. In this halfway-stop technique, the distortion from partial pixels is compared with the normalized current minimum distortion obtained with full pixels of a block. Although the normalized process increases the probability of early rejection of non-possible candidate predictor block, it also increases the risk of false rejection. In addition, the employment of the normalized current minimum distortion is questionable as it may give inaccurate estimate for blocks with complex textures.

One aim of a first aspect of the invention is to reduce the complexity of block motion estimation.

According to one aspect of the present invention there is provided a method of processing a stream of video data comprising a plurality of frames, each frame having a plurality of blocks, the method comprising: locating in a subsequent frame a block closest to a selected block in the current frame using a block distortion measure, wherein the block distortion measure is determined by: calculating a partial error result using a proportion of the pixels in the block; comparing the partial error result with a threshold value and rejecting blocks where the partial error result is greater than the threshold value; and calculating a full error result for non-rejected blocks only, wherein the threshold value is a function of the current minimum partial error result.

In the described embodiment, it is the current minimum partial error result multiplied by a threshold constant (λ).

Another aspect provides a system for processing a stream of video data comprising a plurality of frames, each frame having a plurality of blocks, for locating in a subsequent frame a block closest to a selected block in the current frame using a block distortion measure, the system comprising: means for calculating a partial error result using a proportion of the pixels in the block; means for comparing the partial error result with a threshold value and rejecting blocks where the partial error result is greater than the threshold value; and means for calculating the full error result for non-rejected blocks only, wherein the threshold value is a function of the current minimum partial error result.

The above aspects of the invention are preferably implemented in a source coding stage of a video transmission system, such as that illustrated in Figure 2A. Such a source coding stage can provide a channel coding stage which prepares video data for transmission.

For video streaming over packet erasure networks, an end-to-end video transmission system should adapt to time varying packet loss rate such that video quality can degrade gracefully in the presence of packet loss. Most of current video encoding schemes, such as MPEG-1/2/4, are based on hybrid coding of motion compensated prediction and transform coding of the residue prediction error. Different importance of video data mainly exists in two levels of such a hierarchical video coding structure, in terms of sensitivity to packet loss or error. Firstly, the different importance of video frames or video object planes (VOPs) exists at group of picture (GOP) level. Although motion compensated prediction gains high compression efficiency, the compressed bit-stream is fairly vulnerable to error propagation since the decoding of the current frame strongly depends on all of the preceding frames in the GOP. From such temporal dependency existing in GOP, as illustrated in Figure 3 we can see that the earlier an error occurs in GOP, the more frames will be corrupted. For example, an error in I-frame will corrupt all pictures in the GOP while an error in the last frame of the GOP does not corrupt any other pictures. Thus, the frames in a GOP have descending

importance from the earlier frames to the later frames. Secondly, with data partitioning in MPEG-4 error resilience tools such as described in Weiping Li, J.-R. Ohm, M. van der Schaar, Hong Jiang, Shipeng Li, "MPEG-4 Video Verification Model version 17.0," *ISO/IEC JTC1/SC29/WG11, N3515*, Beijing, July 2000, the different importance can also be found at resynchronization packet level when resynchronization packet is partitioned into motion part and texture part. Once the texture part is lost, data partitioning allows the use of motion part to conceal errors in an effective manner of motion compensation. However, the received texture part, which is usually encoded by transform coding of the residue prediction error, is usually discarded if corresponding motion part is lost. Thus, motion part is more important than texture part.

From view of different importance, the compressed bit-stream is inherently prioritized. Without any means of classifying data based on priorities, most current packet erasure networks are not straightforwardly suitable for transmitting such prioritized bit-stream.

It is an aim of another aspect of the invention to convert the prioritized bit-stream with different importance into non-prioritized packets.

The described embodiment of the present invention provides a video transport scheme which exploits the different importance of both GOP level and resynchronization level.

Although some existing ULP schemes for packet erasure networks have been proposed, these conventional schemes only focus on ULP for particular level, e.g. GOP level or resynchronization level, instead of fully exploiting the different

importance of both GOP level and resynchronization level from the context of video encoding hierarchy. Furthermore, the different sensitivity is not completely exploited even in its focused level for most of the conventional schemes. For example, GOP level ULP schemes are proposed in Albanese, J. Blomer, J. Edmonds, M. Ludy and M. Sudan, "Priority Encoding Transmission", *IEEE Trans. Information Theory*, vol. 42, Nov. 1996, pp.1737-1744, F. Hartanto and H.R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks", selected papers of *10th IEEE Workshop on Local and Metropolitan Area Networks*, 2001, pp. 126-132]. In, Albanese, J. Blomer, J. Edmonds, M. Ludy and M. Sudan, "Priority Encoding Transmission", *IEEE Trans. Information Theory*, vol. 42, Nov. 1996, pp.1737-1744 the Priority Encoding Transmission (PET) implementation for MPEG-1 allows user to set different priorities of error protection for different frames in a GOP. However, PET did not provide any explicit method to determine the optimal values of priority levels. In F. Hartanto and H.R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks", selected papers of *10th IEEE Workshop on Local and Metropolitan Area Networks*, 2001, pp. 126-132, a FEC assignment scheme is proposed, which empirically assigns FEC to I-frame and P-frame with fixed ratio by treating all P-frames equally, thus the temporal dependency relationship among P frames in GOP is not exploited.

Recently, several resynchronization level ULP schemes for wireless network have been proposed in S. Worrall, S. Fabri, A.H. Sadka, A.M. Kondo, "Prioritisation of Data Partitioned MPEG-4 Video over Mobile Networks", *IEEE Packet Video 2000*, Sardinia, Italy, May 1-2, 2000 and M. Budagavi, W. Heinzelman J. Webb and R. Talluri, "Wireless MPEG-4 Video Communication on DSP Chips", *IEEE Signal Processing Magazine*, Jan. 2000, pp. 36-53. These methods unequally protect the motion part and the texture part within a resynchronization packet and are proven effective in combating random noise-like errors in wireless channels. However, these methods are not applicable to the packet erasure networks. For video transmission over the packet erasure networks, one packet loss could

cause the loss of all the motion data, the texture data and their associated FEC data. Thus the ULP schemes for random noise-like errors may not be useful for the packet erasure networks. In T. James, Chung-How, and R. B. David, "Loss resilient H.263+ video over the Internet", *Signal Processing: Image Communication*, Vol.16, 2001, pp.891-908, James et al proposed a loss resilient scheme for transmitting H.263+ video over the Internet, which only assigns FEC to motion information but not to texture information. Obviously, the scheme in T. James, Chung-How, and R. B. David, "Loss resilient H.263+ video over the Internet", *Signal Processing: Image Communication*, Vol.16, 2001, pp.891-908 is not an optimal FEC assignment scheme, which will result in that motion information is unnecessarily tightly protected while texture information is without any protection. Furthermore, the scheme in T. James, Chung-How, and R. B. David, "Loss resilient H.263+ video over the Internet", *Signal Processing: Image Communication*, Vol.16, 2001, pp.891-908 did not provide any explicit method to determine how much FEC should assign to motion information.

It is therefore an aim of this aspect of the invention to robustly transport video streams, e.g. MPEG 4 over the packet erasure networks.

This is achieved in the preferred embodiment by using an unequal loss protection (ULP) philosophy, that is: partitioning the bit-stream data into several parts according to different importance, and then unequally protecting the parts from packet loss using forward error correction (FEC) codes.

In general terms, a further aspect of the invention provides a method of preparing a stream of video data for transmission, the method comprising: assigning a first number K of video packets to each of a group of packet blocks (BOP) to be transmitted in temporal sequence; generating for each packet block a second number of error correction packets, where the ratio of the first number of video packets to the total number N of packets in a block represents the code rate r and

wherein the second number of error correction packets is determined so that the code rate for a packet block earlier in the temporal sequence is no greater than a packet block later than it in the temporal sequence.

A still further aspect of the invention provides a system for preparing a stream of video data for a transmission, the system comprising: means for assigning a first number K of video packets to each of a group of packet blocks (BOP) to be transmitted in temporal sequence; and means for generating for each packet block a second number of error correction packets, where the ratio of the first number of video packets to the total number N of packets in a block represents the code rate r and wherein the second number of error correction packets is determined so that the code rate for a packet block earlier in the temporal sequence is no greater than a packet block later than it in the temporal sequence.

For a better understanding of the present invention and to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings.

Brief Description of the Drawings

Figure 1 illustrates different decimation patterns for a 16x16 pixel block;

Figure 2 is a flow chart of the BDM calculation in 2S-PDS algorithm;

Figure 2A is a schematic block diagram of a video transmission system;

Figure 2B is a schematic diagram of a system implementing the 2S-PDS algorithm;

Figure 2C is a schematic diagram of a system implementing HULP algorithm;

Figure 3 illustrates a temporal dependency of frames in GOP;

Figure 4 illustrates the ULP-GOP framework;

Figure 5 illustrates a number of BOPs illustrating the ULP-GPO optimization algorithm;

Figure 6 illustrates data partitioning;

Figures 7 and 7a illustrate the ULP-DP framework; and

Figure 8 illustrates a simplified implementation structure for the different importance partitioning algorithm (HULP).

Description of the Preferred Embodiment

Figure 2A is a schematic block diagram of a typical video transmission system comprising a source coding stage 2 and a channel coding stage 4. A stream of raw video data 6 is input to the source coding stage which comprises a block motion estimation block 8 and a coding block 10. Coded data is supplied from the source coding stage 2 to the channel coding stage 4 where it is processed in a manner making it suitable for a bus transmission over packet erasure networks as described in more detail in the following.

Reference will now be made to Figures 1 and 2 to explain a method for reducing the complexity of block motion estimation using a novel algorithm, referred to herein as a two stage partial distortion search (2S-PDS) algorithm. The technique for improving the block distortion measure is carried out in the block motion estimation block 8 using a suitably programmed computer or other means. In the 2S-PDS algorithm, as shown in the schematic block diagram of Figure 2B an early-rejection stage is introduced into the block distortion measure (BDM) calculation where the partial distortion is first calculated in block 14 from a portion of the candidate block CB and compared at comparator 15 with its current minimum as held in a minimum store 16 multiplied with a predefined threshold $\gamma > 1$. If this partial distortion is larger, this block is not likely to be a matching block and it is rejected without further calculating the full distortion. In general, the probability of false rejection is very small for large values of γ . However, larger γ will generally result in less candidate blocks being rejected in the early rejection stage and hence less complexity reduction. In a practical application, the value of γ can be adjusted to determine the trade-off between the performance and the complexity reduction.

The algorithm can be used together with different block matching algorithms such as FS, 3SS, DS and HS.

Figure 2 is a flow chart illustrating the steps of the block distortion measure calculation method. According to step 20, the partial sum of absolute errors ($SAE_{partial}$) is computed in an early-rejection stage. SAE is chosen as the matching criteria due to its low computational complexity. With reference to Figure 1, we refer to a block of $M \times N$ pixels as block (k, l) where the coordinate (k, l) is the upper left corner of the block and denote the intensity of the pixel with coordinates (i, j) in frame n by $F_n(i, j)$. In the early rejection stage the partial SAE is computed between the block (k, l) of present frame and the block $(k+x, l+y)$ of the previous frame with a pre-defined decimated pattern Φ :

$$SAE_{partial} = \sum_{(i,j) \in \Phi} |F_n(k+i, l+j) - F_{n-1}(k+x+i, l+y+j)| \quad (1)$$

Here the pattern Φ is a decimated version of a $M \times N$ block. Some possible decimation patterns Φ with different decimation factors are given in Figs.1a to 1c where the pixels marked with black are used in calculating SAE .

The full SAE is then calculated in block 17 in Figure 2B by using all the pixels in the block:

$$SAE_{full} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |F_n(k+i, l+j) - F_{n-1}(k+x+i, l+y+j)| \quad (2)$$

However, it is an important feature of this technique that the full SAE is not calculated in respect of all blocks. In step 20, for every matching block candidate (x, y) in a particular block matching algorithm, the value of the partial SAE $SAE_{partial}$ is computed initially. In step 22, $SAE_{partial}$ is compared with the current minimum value $SAE_{partial, min}$ times a pre-defined threshold λ . If $SAE_{partial} > \lambda \cdot SAE_{partial, min}$, the

block (k, l) is rejected without calculating the full SAE. In this way, substantial computational saving is achieved since most processing can be completed in this early-rejection stage. In the second stage, the value of the full SAE SAE_{full} is calculated and compared with the current minimum value $SAE_{full,min}$ at step 26. The coordinate of that current checking point (x, y) is labelled as the best matching block so far if $SAE_{full} < SAE_{full,min}$ and meanwhile the values of $SAE_{partial}$ and SAE_{full} are recorded as the current minimum values in the minimum store 16 at step 28. The search procedure proceeds according to the search algorithm used.

The value of λ plays a key role that determines the trade-off between the accuracy of the motion estimation and the computation complexity. For example, very large value of λ will result in few blocks being rejected in the early-rejection stage and hence the estimation results will resemble those obtained with full block SAE calculation. However the saving in computation complexity will be small. The value of λ can be selected empirically according to the requirements of a particular implementation. It is found that the using $\lambda = 1.1 \sim 1.3$ yields a satisfactory trade-off between the performance and complexity for most applications.

Thus there has been described above a novel 2S-PDS algorithm for block motion estimation where an early-rejection stage is introduced in the BDM calculation to reduce the computational complexity. The proposed 2S-PDS algorithm can be applied to different search strategies, such as the Full Search (FS), 3 Step Search (3SS), 4 Step Search (4SS), Diamond Search (DS), and the Hexagon Search (HS), to reduce the computation complexity for efficient implementation.

Reverting to Figure 2A, the coded video stream is supplied from the source coding stage 2 to the channel coding stage 4. In the channel coding stage, the data is processed in accordance with a hierarchical unequal loss protection (HULP) algorithm to provide data for a bus transmission of MPEG-4 video over packet erasure networks 12. The scheme is discussed in more detail in the

following. Figure 2C is a schematic block diagram showing the relevant components of the channel coding stage 4. In brief, a video packet assignor 40 assigns video packets of an incoming GOP to respective BOPs. Forward error correction (FEC) packets are generated for each BOP according to two different protocols, based on whether the video packet is in an I-frame or a P-frame. An I-packet FEC generator 42 generates FEC packets for I-packet video frames and a P-packet FEC generator 44 generates FEC packets for a p-frame packet.

FEC codes generated by the I-packet FEC generator each constitute a single code relating to a single video packet, and are denoted as FEC in Figure 2C. However, FEC codes for P-packet frames are generated separately for a motion part and a texture part of the packet and are labelled respectively FEC_{MP} and FEC_{TP} .

An FEC assignor 46 assigns the FEC codes to the BOPs according to assignment protocols ULP_GPO and ULP_DP discussed in more detail herein. In essence however these assignment protocols allow BOPs which are temporarily earlier to have more error code protection than later BOPs (ULP_GOP). Moreover, motion part FEC codes are assigned preferentially to texture part FEC codes (ULP_DP).

Implementations of the above-referenced concept are discussed in more detail in the following.

For simplicity of description, we concern the GOP structure as shown in Fig. 3, where one I-frame is followed by several P-frames. In spite of such GOP format without B-frame, the proposed algorithm in the following sections can be easily

extended to the other GOP structure with B-frames, since errors in B-frames do not affect other frames.

Protecting video packets against burst packet loss

Extensive studies have been presented to explore packet loss behaviour and loss patterns in packet erasure networks. All of the studies support two basic observations: 1) packet loss rate is time-varying, and 2) packet loss occurs in *bursts*, i.e., if one packet is lost it is very likely that consecutive packets will also be lost.

The robustness against packet losses, especially in cases where a feedback channel is not available, can be increased if the video packets are protected with an appropriate forward error correction (FEC) scheme, i.e., additional redundant packets are transmitted at sender-side and used to recover lost packets at the receiver-side. Since Reed-Solomon (RS) code is a class of FEC code with high performance against burst packet loss, RS code across packets is used to protect video packets in the scheme described herein. For (n, k) RS code across k video packets, the code rate r is k/n and the resulting n packets are called a block of packets (BOP). If each transmitted packet is marked with a sequence number, all the k video packets can be recovered from any subset consisting of at least k correctly received packets in this BOP by erasure decoding as described for example in M. Budagavi, W. Heinzelman J. Webb and R. Talluri, "Wireless MPEG-4 Video Communication on DSP Chips", *IEEE Signal Processing Magazine*, Jan. 2000, pp. 36-53 and B. Girod, K. Stuhlmuller, M. Link and U. Horn, "Packet Loss Resilient Internet Video Streaming", *SPIE Visual Communications and Image Processing 99*, January 1999, San Jose, CA.

Obviously, the more FEC packets, the more robust a video stream is. On the other hand, to compensate for the additional redundancy packets introduced by

FEC, the video bit rate must be decreased accordingly, so that the initial video quality at the sender-side is decreased. To determine the appropriate number of FEC packets in a BOP, we should calculate the target packet loss rate for (n, k) code after recovery at receiver-side, $PLR(k, n)$. Let $P(m, n)$ denote the probability density function (pdf) of m lost packets among n . Then $PLR(k, n)$ refers to the probability that more than $n - k$ packets lost, and it can be calculated as follows:

$$PLR(k, n) = \sum_{m=n-k+1}^n P(m, n) \quad (3)$$

$P(m, n)$ is determined by a channel estimator and it can be formulated as any distribution about packet loss rate, such as uniform, binomial and exponential distributions, and it can also be modeled as state based systems, e.g., the Internet can be modeled as a two-states Markov model as proposed in M. Budagavi, W. Heinzelman J. Webb and R. Talluri, "Wireless MPEG-4 Video Communication on DSP Chips", *IEEE Signal Processing Magazine*, Jan. 2000, pp. 36-53. The development of a suitable channel estimator depends on the particular application.

Assembling Video Packets into BOP

According to the above discussion, the protection capability of RS code depends on the BOP size and the code rate. If FEC is directly assigned to each frame for the applications of transmitting very low bit-rate video over burst packet loss channel, the performance of FEC may be very poor for those frames with very few packets. For example, if a compressed frame has 4 video packets and 4 FEC packets are assigned, then the code rate is $\frac{1}{2}$. When such BOP is transmitted over a burst loss channel with an average burst loss length of more than 4 packets, the produced 4 FEC packets are not enough to correct the burst packet loss normally. In order to avoid the situation where a BOP is composed of very few video packets, multiple compressed frames are assembled together to form a

relatively big BOP. For example, if we assemble 5 frames with the same number of video packets mentioned above together to form a BOP with 20 video packets, and then there are 20 FEC packets for the assembled BOP with the same code rate of $\frac{1}{2}$, which significantly improves robustness compared with transmitting each frame BOP individually. Thus, the bigger a BOP is, the more robust it is. However, the assembled BOP results in longer delay for erasure decoding. Therefore, the total number of video packets in a BOP directly relates to the performance of RS code in the burst channel. For the Internet application, the target number of video packets in BOP, κ , can be determined according to the end-to-end system delay constraint B. Girod, K. Stuhlmüller, M. Link and U. Horn, "Packet Loss Resilient Internet Video Streaming", *SPIE Visual Communications and Image Processing 99*, January 1999, San Jose, CA. The typical value of κ is 20-50 for video sequence with CIF format and compressed at bit-rate range of 200Kbps-1Mbps.

After determining the value of κ , we attempt to assemble video packets from a GOP into several BOPs in a sequential order from the packets in the earlier frames to that in the latter frames. With reference to Figure 4, after the video data part in one BOP is full (e.g. BOP1), the following video packets will be filled into the next BOP (BOP2) until it is full again. Such process is repeated until all video packets are filled. Finally, a number F_j of FEC packets are generated and appended in the rear part of each BOP as shown in Fig. 4, where

T the GOP length;

t the t -th frame in a GOP, $t = 1, 2, \dots, T$;

Z_t the number of video packets in frame t , which can be obtained in the encoding time;

z_t the z_t -th packet of the t -th frame, and $z_t = 1, 2, \dots, Z_t$;

J the total number of BOPs in a given GOP,

j the j -th BOP, $j = 1, 2, \dots, J$;

K_j the number of video packets in BOP j ;

N_j the total number of packets in BOP j ;

F_j the number of FEC packets in BOP j , and $F_j = N_j - K_j$.

Then, the 3-tuple (t, z_t, j) in Fig.4 means the z_t -th packet of the t -th frame in BOP j . For BOP j , K_j video packets are protected by F_j FEC packets.

To assemble the packets into BOPs, we try to make all BOPs have the same number of video packets except the last BOP. Then $K_1 = K_2 = \dots = K_{J-1} = \kappa$. To avoid too small a number of packets in the last BOP considering protection efficiency mentioned above, the number of BOPs is obtained by:

$$J = \text{round}((\sum_{t=1}^T Z_t) / \kappa) \quad (4)$$

And the number of video packets in the last BOP is given by:

$$K_J = \begin{cases} \kappa + (\sum_{t=1}^T Z_t) \bmod \kappa, & \text{if } (\sum_{t=1}^T Z_t) \bmod \kappa < \kappa / 2 \\ (\sum_{t=1}^T Z_t) \bmod \kappa, & \text{otherwise} \end{cases} \quad (5)$$

Thus we can see the number of video packets in the last BOP is at least $\kappa / 2 \leq K_J \leq 3\kappa / 2 - 1$.

Assume the bit budget R has been estimated for a GOP with the total number of encoded bits S known after encoding, the total amount of FEC F for the GOP can be determined by $F = R - S$. We now try to find an optimal error protection policy π corresponding to a defined performance criterion of quality degradation

D_j^π . Let the FEC assignment vector $\vec{F} \equiv [F_1, F_2, \dots, F_J]$, then the optimal FEC vector $\vec{F}^*$ can be obtained by minimizing the expected quality degradation as follows:

$$D^\pi = \sum_{j=1}^J d_j^\pi \cdot PLR(K_j, F_j + K_j) \quad (6)$$

subject to

$$\sum_{j=1}^J \lambda_j \cdot F_j = F \quad (7)$$

$$r_1 \leq r_2 \leq \Lambda \leq r_j \leq \Lambda \leq r_J \quad (8)$$

where λ_j is the length of FEC packet in BOP j and it is equal to the maximum packet length among the video packets in BOP j .

r_j is the code rate of (N_j, K_j) RS code and $r_j = K_j / N_j$. To generate FEC packets, all video packets are padded with stuffing bits in the rear parts to make them have length λ_j , and the stuffing bits are removed after generating FEC codes and are not transmitted.

The constraint (8) is referred to as “descending priority” constraint to apply more protection redundancy to earlier BOPs.

Peak Signal to Noise Ratio (PSNR) or Mean Square Error (MSE) is the most common performance criterion to measure image distortion. To compute the expected PSNR/MSE, it is therefore important to consider the effect of error propagation. If PSNR/MSE serves as the measurement of quality degradation due to error propagation, a theoretical model should be used to describe the

effect of packet loss on the decoded picture quality. The parameters of such model obviously vary with picture complexity of video sequence and codec complexity of video system. Furthermore, determining such parameters will involve decoding process with high computational complexity. Thus, PSNR/MSE is computationally prohibitive to measure quality degradation for most of practical video applications.

To make the measurement of the quality degradation easier, there is discussed herein a simple but practical performance criterion, namely *the expected length of error propagation* (ELEP). This criterion is motivated by the fact that the fewer frames in the GOP are corrupted, the better the reconstructed video quality would be. The length of error propagation d_j^{LEP} due to error propagation of packet loss in the j -th BOP can be given by:

$$d_j^{LEP} = \frac{1}{K_j} \sum_{t=1}^T \sum_{z_t=1}^{Z_t} (T - t + 1) \cdot \delta(t, z_t, j) \quad (9)$$

where $\delta(t, z_t, j)$ indicates whether or not a video packet (t, z_t) is in BOP j , which is equal to 1 if video packet (t, z_t) belongs to BOP j , otherwise it is 0.

Obviously, a trade-off exists between protection of earlier frames and that of later frames. If we assign more FEC to earlier frames, we have to assign less FEC to later frames. So, the length of error propagation due to the earlier frames is shortened while the probability of later frames being corrupted will increase. Based on such trade-off, the optimal FEC vector $\vec{F}^*$ can be found by minimizing the overall ELEP D^{LEP} given by:

$$D^\pi = D^{LEP} = \sum_{j=1}^J d_j^{LEP} \cdot PLR(K_j, F_j + K_j) \quad (10)$$

In the above ELEP computation, we assume that the superimposed error signals are uncorrelated temporally and spatially. The assumption of the temporal

uncorrelation is automatically met when individual error signals are temporally separated in the decoded video sequence. This is very common for the situation with low packet loss rates. The assumption of the spatial uncorrelation can be met when every video packet is equivalent to a resynchronization packet, such as MPEG-4 resynchronization packet and MPEG-1/2 slice.

Solving ULP-GOP optimization

Now, there is discussed an algorithm for solving the above optimization problem to find $\vec{F}^*$. Searching the globally optimal assignment seems computationally prohibitive for a practical system. Instead a local search hill-climbing algorithm is used to compute the sub-optimal FEC assignment as shown in Fig. 5. Initially, let $F_1 = F_2 = \dots = F_J = 0$. In each iteration, we examine a number of possible assignments equal to $2 \sum_{j=1}^J Q_j$, where Q_j is the search distance (maximum number of FEC packets that can be added or subtracted to a BOP in one iteration). Q_j is further determined by $Q_j = \max\{\lfloor \Delta \cdot K_j \rfloor, 1\}$ where Δ is a predefined constant used to control search distance. We compute D^{LEP} after adding or subtracting 1 to Q_j FEC packets to BOP j while satisfying the constraints (7) and (8). Accordingly, we choose the $\vec{F}$ corresponding to the lowest D^{LEP} , update the allocation of FEC packets to all affected BOPs, and repeat the search until none of the cases examined improves the D^{LEP} .

ULP-DP

Now, all available protection redundancy for a GOP has been assigned to BOPs in forms of FEC using the above ULP-GOP. To optimally protect motion parts and texture parts in those BOPs where video packets only come from P-frames, the FEC in such BOP is further unequally assigned by the ULP-DP algorithm

discussed herein. Those BOPs, which include any video packet from I-frame, are usually from the earlier frames in GOP, and normally protected tightly enough. In such BOP, some video packets may come from P-frame. Since they have been tightly protected by ULP-GOP, it is unnecessary to further protect the motion parts in the BOP with mixed video packets from I-frame and P-frame. In other word, only the BOP including pure P-frame video packets need to be further protected using ULP-DP.

ULP-DP Framework

To realize ULP-DP, we classify the data in each video packet into motion part (MP) and texture part (TP) as shown in Fig. 6. For BOP j , we generate $(N_j^{MP} - K_j)$ and $(N_j^{TP} - K_j)$ FEC packets to unequally protect the motion parts and texture parts respectively. Then, we can completely recover $(N_j^{MP} - K_j)$ loss packets for motion parts and $(N_j^{TP} - K_j)$ loss packets for texture parts. Fig. 7(a) illustrates an example with $K_j = 3$, $N_j^{MP} = 6$ and $N_j^{TP} = 4$. If the first to the third packets are lost, MP1, MP2 and MP3 can still be recovered from three received FEC packets while TP1, TP2 and TP3 cannot be recovered due to only one FEC packet received.

The FEC length for motion parts in BOP j , $L_j^{FEC_MP}$, is calculated by

$$L_j^{FEC_MP} = \max\{L_j^{MP}(1), L_j^{MP}(2), \dots, L_j^{MP}(i), \dots, L_j^{MP}(K_j)\} \quad (11)$$

where $L_j^{MP}(i)$ denotes the motion part length of i -th video packet in BOP j . Then, the FEC length for texture parts in BOP j , $L_j^{FEC_TP}$, is calculated by:

$$L_j^{FEC_TP} = \max\{L_j^{MP}(1) + L_j^{TP}(1), \dots, L_j^{MP}(K_j) + L_j^{TP}(K_j)\} - L_j^{FEC_MP} \quad (12)$$

where $L_j^{TP}(i)$ denotes the texture part length of i -th video packet in BOP j .

The motion part length L_j^{FEC-MP} is usually significantly shorter than its associated texture part length L_j^{FEC-TP} . A typical value of $L_j^{FEC-MP} / L_j^{FEC-TP}$ is 1/6. If we directly send the $(N_j^{MP} - N_j^{TP})$ short FEC packets associated with motion parts, e.g. FEC_MP2 and FEC_MP3 in Fig. 7(a), the packet utilization efficiency will be noticeably decreased. To improve the packet utilization efficiency, the $(N_j^{MP} - N_j^{TP})$ short FEC packets for motion parts are shifted to next BOP as shown in Fig. 7(b).

Redundancy-Distortion Optimization

Since we assign more FEC to motion parts, then we have to assign less FEC to texture parts. In other words, given the number of FEC, a tradeoff exists between the protections of motion parts and texture parts. Similarly, we can make the tradeoff by minimizing a given performance criterion for BOP. Thus we need to consider how to design the performance criterion of ULP-DP. Although the MSE is computationally prohibitive to measure the effect on error propagation in ULP-GOP step, it is reasonably practical to use MSE as the performance criterion of ULP-DP if we concern two reasons as follows. Firstly, the MSE increment due to motion part loss or texture part loss can be easily obtained as intermediate outputs in the motion estimation process. Secondly, error propagation within a BOP can be ignored, since a BOP is usually across very few frames and a minimum of error propagation across BOPs has been globally guaranteed by ULP-GOP in a GOP. Then, MSE can be used to locally measure the video quality of BOP. Thus, we can make the tradeoff between motion part protection and texture part protection by minimizing the expected MSE of BOP for given amount of redundancy, namely, redundancy-distortion optimization.

To implement the redundancy-distortion optimization in the j -th BOP, we define following notations for the i -th video packet in the j -th BOP:

$M_j(i)$ the number of macro-blocks in the packet;

$MSE_j^{OK}(i)$ the mean-square-error (MSE) value when both motion part and texture part are received;

$MSE_j^{MC}(i)$ the resulted MSE value when motion part is received but texture part is lost; in this case, motion compensation is used to reconstruct the current macro-blocks and the texture information is discarded;

$MSE_j^{BC}(i)$ the resulted MSE value when neither motion part nor texture part is received; in this case, the macro-blocks in the video packet are copied from the previous frame.

The values of the above notations are obtained in motion estimation in the encoding process. If all video packets in the j -th BOP are lost, it will result in a MSE value MSE_j^{BC} , which is approximated by:

$$MSE_j^{BC} = \frac{\sum_{i=1}^{K_j} M_j(i) \cdot MSE_j^{BC}(i)}{\sum_{i=1}^{K_j} M_j(i)} \quad (13)$$

Hence, the contribution of texture parts and motion parts to decrease MSE can be approximated by:

$$MSE_j^{TP} = \frac{\sum_{i=1}^{K_j} M_j(i) \cdot (MSE_j^{BC}(i) - MSE_j^{MC}(i))}{\sum_{i=1}^{K_j} M_j(i)} \quad (14)$$

$$MSE_j^{MP} = \frac{\sum_{i=1}^{K_j} M_j(i) \cdot (MSE_j^{MC}(i) - MSE_j^{OK}(i))}{\sum_{i=1}^{K_j} M_j(i)} \quad (15)$$

After doing ULP-GOP, we have gotten the total amount of FEC for BOP j by $F_j \times \lambda_j$. Then the optimal values of N_j^{MP*} and N_j^{TP*} can be obtained by minimizing the expected MSE after receiving the j -th BOP as follows:

$$E[MSE] = MSE_j^{BC} - MSE_j^{TP} \cdot P_j^{DEC_TP} - MSE_j^{MP} \cdot P_j^{DEC_MP} \quad (16)$$

subject to

$$(N_j^{MP} - K_j) \cdot L_j^{FEC_MP} + (N_j^{TP} - K_j) \cdot L_j^{FEC_TP} \leq F_j \times \lambda_j \quad (17)$$

$$N_j^{TP} \leq N_j^{MP} \quad (18)$$

where $P_j^{DEC_MP}$ and $P_j^{DEC_TP}$ are the decodable probability of motion parts and texture parts respectively. To estimate $P_j^{DEC_MP}$ and $P_j^{DEC_TP}$, we should determine their receivable probabilities $P_j^{REC_MP}$ and $P_j^{REC_TP}$ respectively. For motion part, its decodable probability $P_j^{DEC_MP}$ is equal to its receivable probability, hence

$$P_j^{DEC_MP} = P_j^{REC_MP} = (1 - PLR(K_j, N_j^{MP})) \quad (19)$$

For texture part, its decodable condition is that the texture part is received and its associated motion part is decodable. Thus, the decodable probability of texture part is given by:

$$P_j^{DEC_TP} = P_j^{REC_MP} \cdot P_j^{REC_TP} = (1 - PLR(K_j, N_j^{MP})) \cdot (1 - PLR(K_j, N_j^{TP})) \quad (20)$$

Since there is only about $\lfloor (F_j \cdot \lambda_j) / L_j^{FEC_MP} \rfloor \cdot \lfloor (F_j \cdot \lambda_j) / L_j^{FEC_TP} \rfloor$ feasible solutions in the above optimization, the global optimum can easily be found by exhaustive search with pruning.

A simplified structure for implementing HULP

The optimal protection policy can be achieved by the above mentioned ULP-GOP and ULP-DP, which need a channel estimator to determine the *pdf* of packet loss. If the channel estimator is not available for some applications, we can alternatively implement HULP in a simplified way as shown in Fig. 8. The simplified HULP scheme unequally protects the assembled BOPs in the order of importance from earlier frames to latter frames in the ULP-GOP point of view, as well as from motion parts to texture parts in the other view of the ULP-DP philosophy. The assignment of FEC includes six steps as in Fig. 8, where the FEC packets assigned in step i is labeled as " i ". If only FEC codes remain available, we try to assign $F(i,j)$ FEC packets to BOP j in the sequential order from earlier BOPs to latter BOPs in step $i=1,2,3,4,5$. (The dot lines in Fig.8 illustrate how FEC packets are assigned to six BOPs in Step 1, and it is similar for Step 2 to 5.) If FEC codes still remain available after the 5-th step, we assign the remnant FEC codes in Step 6, where FEC packets labeled as "6" are assigned to BOPs in a raster scan order as the dot lines of Fig.8. Once FEC codes are used up in a particular step, the assignment process stops immediately. In each step, the FEC codes may be assigned to motion parts and/or texture parts for a BOP as shown in Fig.8. The values of $F(i,j)$ for Step 1-5 are given as follows:

$$\begin{aligned}
 F(1, j) &= B; \\
 F(2, j) &= \begin{cases} \lfloor \alpha \cdot (J-1-j) \rfloor, & j \leq J-2 \\ 0, & j = J-1, J \end{cases} \\
 F(3, j) &= \begin{cases} 0, & j = 1 \\ F(1, j) + F(2, j), & j \neq 1 \end{cases} \\
 F(4, j) = F(5, j) &= \begin{cases} 0, & j = 1 \\ F(2, 1) - F(2, j), & j \neq 1 \end{cases}
 \end{aligned}$$

where B and α are two adjustable parameters. The recommended value of B is the estimated burst loss length. An empirical value of α is 1 for median packet loss rate (5-10%). In the example of Figure 8, $B=2$ and $\alpha=1$. However, their parameters may be adjusted to suit any particular application.

From step 1-3, we can see that the number of FEC packets for BOP is linearly decreased according to α in the order of descending sensitivity from the earlier BOPs to the latter BOPs, thus ULP-GOP is realized. In the context of step 1-5, we can see that the FEC codes for motion parts are assigned prior to texture parts, hence ULP-DP is realized.

There follows a more detailed description of the assignment process illustrated in Figure 8. In step 1, $F=2$. Therefore, two FEC packets are assigned to each of the J BOPs. These packets are labeled 1 in Figure 8. In BOP1, which is a packet from the i-frame, these FEC packets are full FEC codes. For the remaining BOPs, which are from the p-frame, the FEC packets are divided into FEC_MP and FEC_TPs. In step 1 therefore two FEC_MP packets are assigned to each of the BOPs in the p-frame.

In step 2, four full FEC packets are assigned to BOP1, three FEC_MP packets are assigned to BOP2, two FEC_MP packets are assigned to BOP3 and one FEC_MP packet to BOP4. The precise degradation of the assignment of packets in step 2 depends on α .

In step 3, no packets are assigned to BOP1. Five FEC_TP packets are assigned to BOP2 (being the number assigned in step 1, 2 and the number assigned in step 2, 3 taken together). These packets are FEC_TP packets, because with FEC_MP packets assigned to motion parts in step 1-2, the motion parts have been protected strongly enough in most cases. Further protection of motion parts is unnecessary in the theoretical point of view. In other word, the effective packet loss rate of motion parts has normally been low enough after protection of step 1-2. On the other hand, the texture parts are still without any protection after step 1-2. Thus, the protection alternatively emphasizes on texture parts from step 3, although some FECs are also assigned to motion parts in step 4 and step 6 in order to make full use of available FECs.

In step 4, no packets are assigned to BOP1, a single packet is assigned to BOP2, which is a FEC_MP packet, two FEC_MP packets are assigned to BOP3 and then an increasing number of FEC_MP packets, again according to α , to successive BOPs.

In summary, we only protect motion parts in step 1-2 so that motion parts can be protected strongly enough with higher priority. Texture parts have not any opportunity of protection until step 3. After step 1-2, motion parts have already been protected strongly enough, thus the protection alternatively emphasizes on texture parts from step 3. In the context of step 1-5, we can see that the FEC codes for motion parts are assigned prior to texture parts.

For example, in Fig.8, if all available FECs exhausts in BOP 2 of step 3, the texture parts in BOP 3-6 are still without any protection.

The above described HULP technique has the following attributes:

- 1) Video quality degrades gracefully when packet loss increases in the packet erasure networks. With HULP, the video information is gracefully dropped in the order of importance from the latter frames to the earlier frames at GOP level as well as from texture part to motion part at resynchronization packet level. Thus, the proposed HULP scheme is expected to adapt to time-varying packet loss rate in the packet erasure networks.
- 2) The ULP-GOP method optimally assigns FEC at GOP level based on the proposed performance criterion, the expected length of error propagation (ELEP). Since the ELEP is not relevant to picture complexity and codec complexity, it can be also used to not only MPEG-4, but also other motion compensated video encoding scheme, such as MPEG-1/2 and H.26X.
- 3) The ULP-DP method minimizes the overall distortion within BOP by optimally assigning the given amount of FEC to motion parts and texture parts

according to the channel status. A tactic packet merging method is used to eliminate the short FEC packets associated with motion parts in order to increase the packet utilization efficiency.

CLAIMS:

1. A method of processing a stream of video data comprising a plurality of frames, each frame having a plurality of blocks, the method comprising:
 - locating in a subsequent frame a block closest to a selected block in the current frame using a block distortion measure, wherein the block distortion measure is determined by:
 - calculating a partial error result using a proportion of the pixels in the block;
 - comparing the partial error result with a threshold value and rejecting blocks where the partial error result is greater than the threshold value; and
 - calculating a full error result for non-rejected blocks only,
 - wherein the threshold value is a function of the current minimum partial error result.
2. A method according to claim 1, wherein the selected block is used as a predictor for the located block in a coding step of the video data.
3. A method as claimed in claim 1 or 2, wherein the partial error result is calculated using the sum of absolute errors for the proportion of pixels.
4. A method according to any preceding claim, wherein the proportion of pixels are distributed uniformly over the block.
5. A method according to any preceding claim, wherein the partial error result which has just been calculated is compared to a prestored minimum and, if it is greater than the prestored minimum is held as the minimum partial error result used to determine the threshold value for the next iteration.
6. A method according to any preceding claim wherein the full error result is compared with a prestored minimum full error result and, if it is greater than the prestored minimum full error result, it is stored as the minimum full error result for

comparison with subsequent full error results for use in locating the closest block.

7. A method according to any preceding claim, wherein the function is the current minimum partial error result multiplied by a threshold constant (λ).

8. A system for processing a stream of video data comprising a plurality of frames, each frame having a plurality of blocks, for locating in a subsequent frame a block closest to a selected block in the current frame using a block distortion measure, the system comprising:

means for calculating a partial error result using a proportion of the pixels in the block;

means for comparing the partial error result with a threshold value and rejecting blocks where the partial error result is greater than the threshold value; and

means for calculating the full error result for non-rejected blocks only, wherein the threshold value is a function of the current minimum partial error result.

9. A system according to claim 8, wherein the function is the current minimum partial error result multiplied by a threshold constant (λ).

10. A system according to claim 8 or 9, which comprises a store for holding the minimum partial error result.

11. A method according to claim 8, 9 or 10, which comprises a store for holding the minimum full error results.

12. A method of preparing a stream of video data for transmission, the method comprising:

assigning a first number K of video packets to each of a group of packet blocks (BOP) to be transmitted in temporal sequence;

generating for each packet block a second number of error correction packets, where the ratio of the first number of video packets to the total number N of packets in a block represents the code rate r and wherein the second number of error correction packets is determined so that the code rate for a packet block earlier in the temporal sequence is no greater than a packet block later than it in the temporal sequence.

13. A method according to claim 12, wherein the group of packet blocks includes I-frame video data and P-frame video data, the I-frame video data being located in temporally earlier blocks.

14. A method according to claim 12 or 13, wherein the number of error correction codes in each packet block is determined by minimizing the expected length of error propagation for the group.

15. A method according to any of claims 12 to 14, wherein each video packet comprises a motion part and a texture part, the method comprising generating respective MP and TP error correction packets for each part.

16. A method according to claim 14, wherein respective MP and TP error correction packets are generated only from video packets from P-frames.

17. A method according to claim 15 or 16, comprising preferentially assigning error correction packets generated from motion parts over error correction packets generated from texture parts.

18. A method according to claim 15 or 16, wherein the assignment of MP error correction packets with respect to TP error correction packets is determined by minimizing the mean square error for the received block.

19. A method according to claim 15 or 16, when implemented as a series of assignment steps, wherein error correction packets generated from motion parts are preferentially assigned in earlier steps.

20. A system for preparing a stream of video data for a transmission, the system comprising:

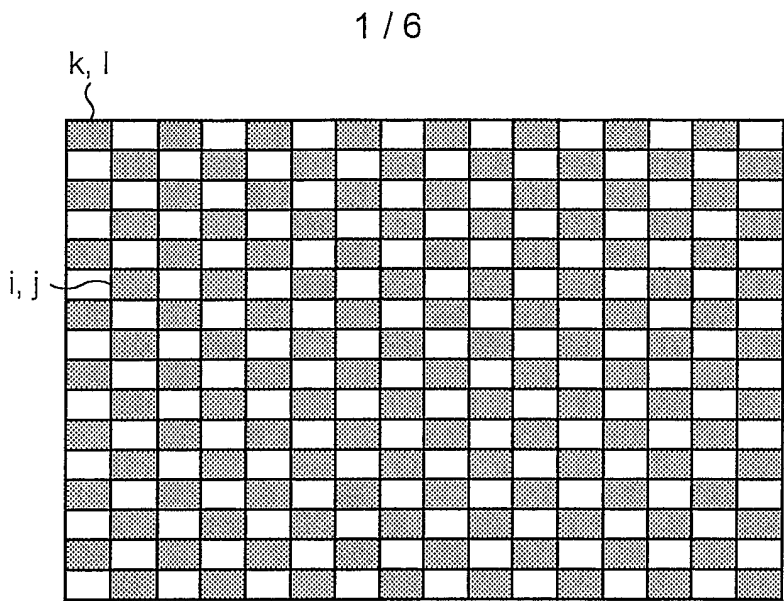
means for assigning a first number K of video packets to each of a group of packet blocks (BOP) to be transmitted in temporal sequence; and

means for generating for each packet block a second number of error correction packets, where the ratio of the first number of video packets to the total number N of packets in a block represents the code rate r and wherein the second number of error correction packets is determined so that the code rate for a packet block earlier in the temporal sequence is no greater than a packet block later than it in the temporal sequence.

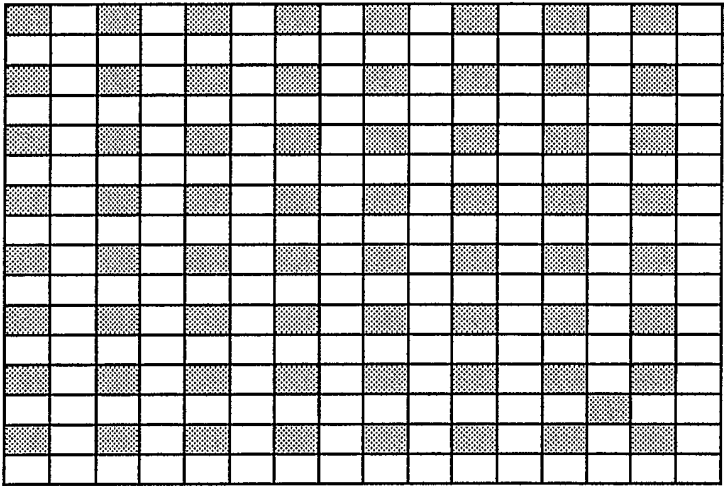
21. A system according to claim 20, wherein the means for generating the error correction packets comprises means for generating for I-frame video data full error correction codes.

22. A system according to claim 20 or 21, wherein the means for generating error correction packets comprises means for generating, for P-frame video data, error correction packets for motion parts and error correction packets for texture parts of said video data.

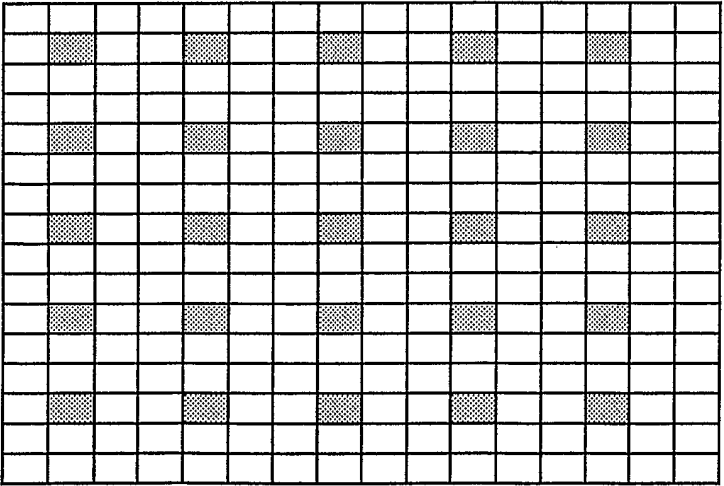
23. A video transmission system comprising a source coding stage including the system as claimed in claim 8 and a channel coding stage including the system as claimed in claim 20.



(a) 1:2 decimation



(b) 1:4 decimation



(c) 1:9 decimation

FIG. 1

2 / 6

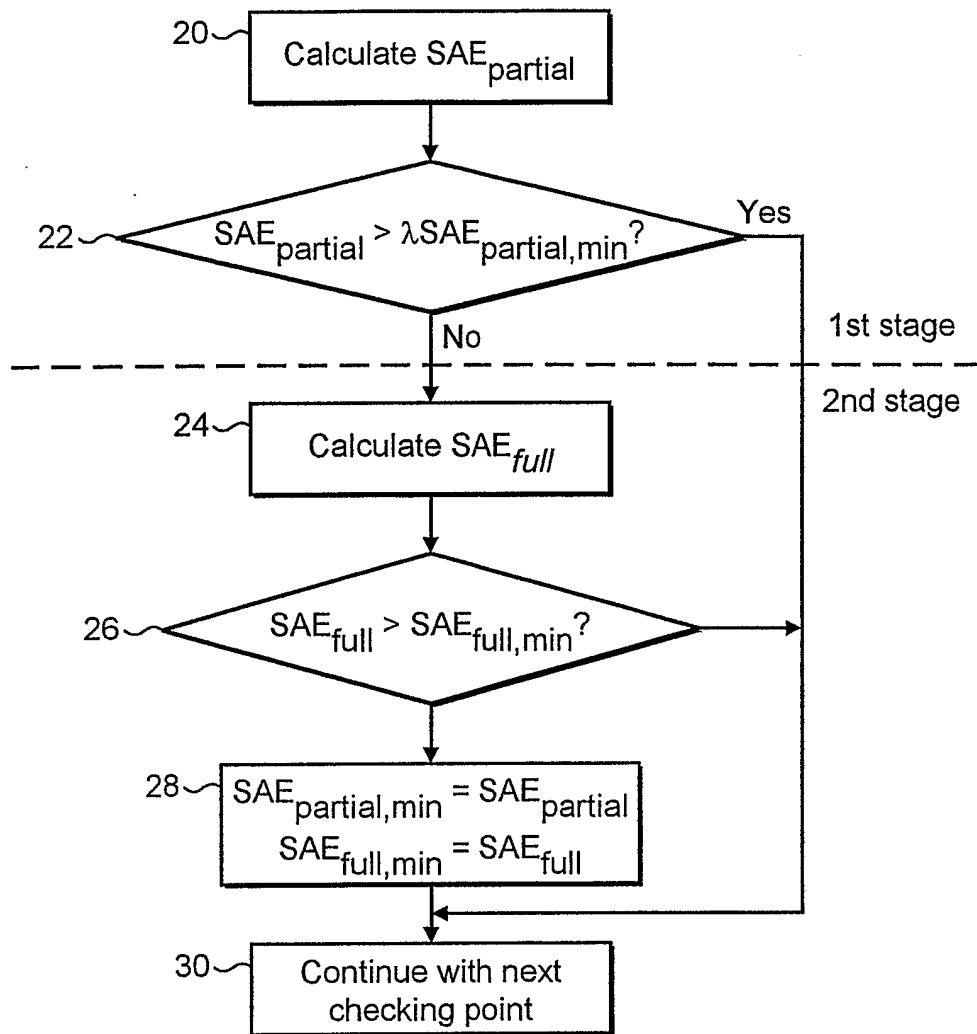


FIG. 2

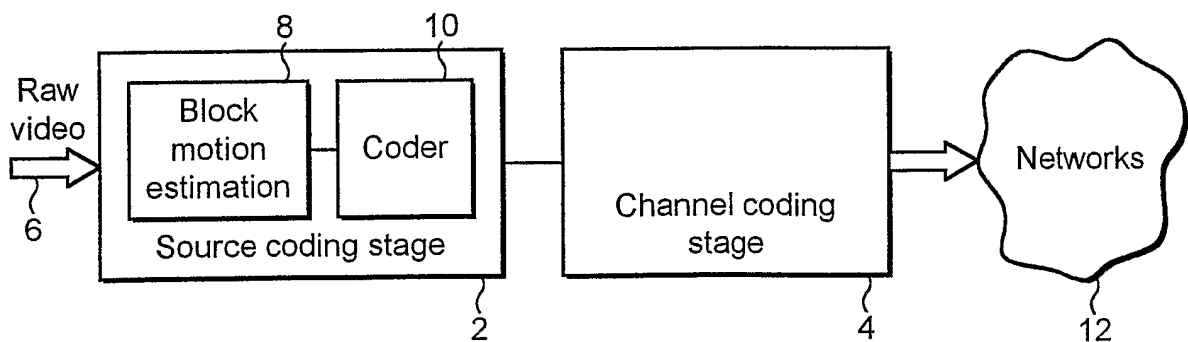


FIG. 2A

3 / 6

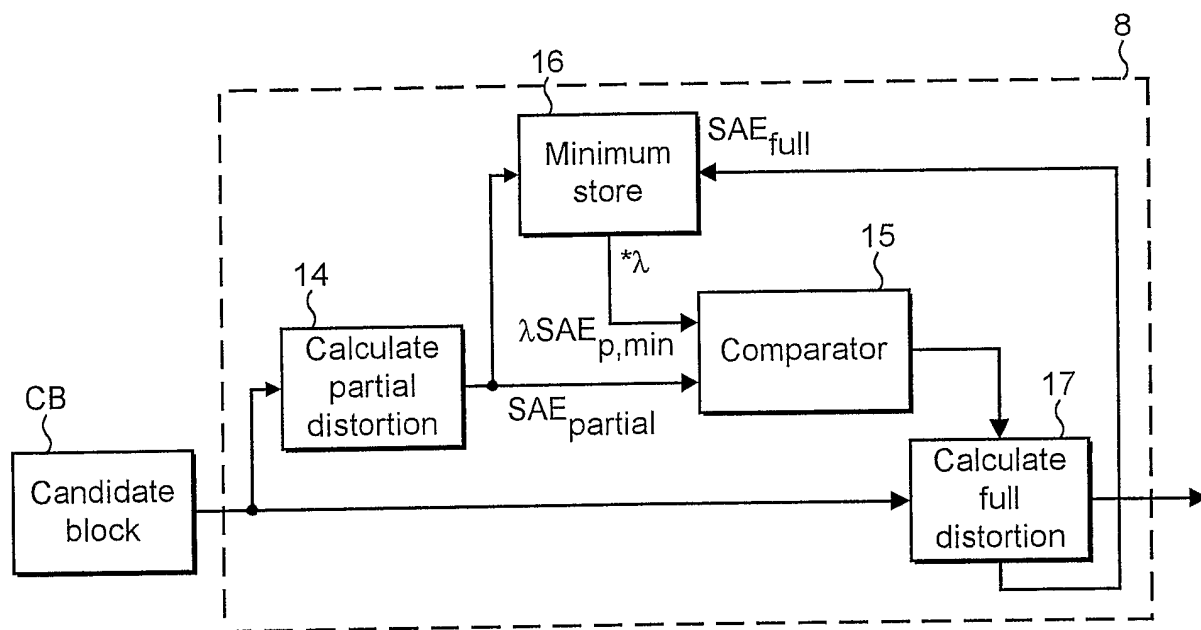


FIG. 2B

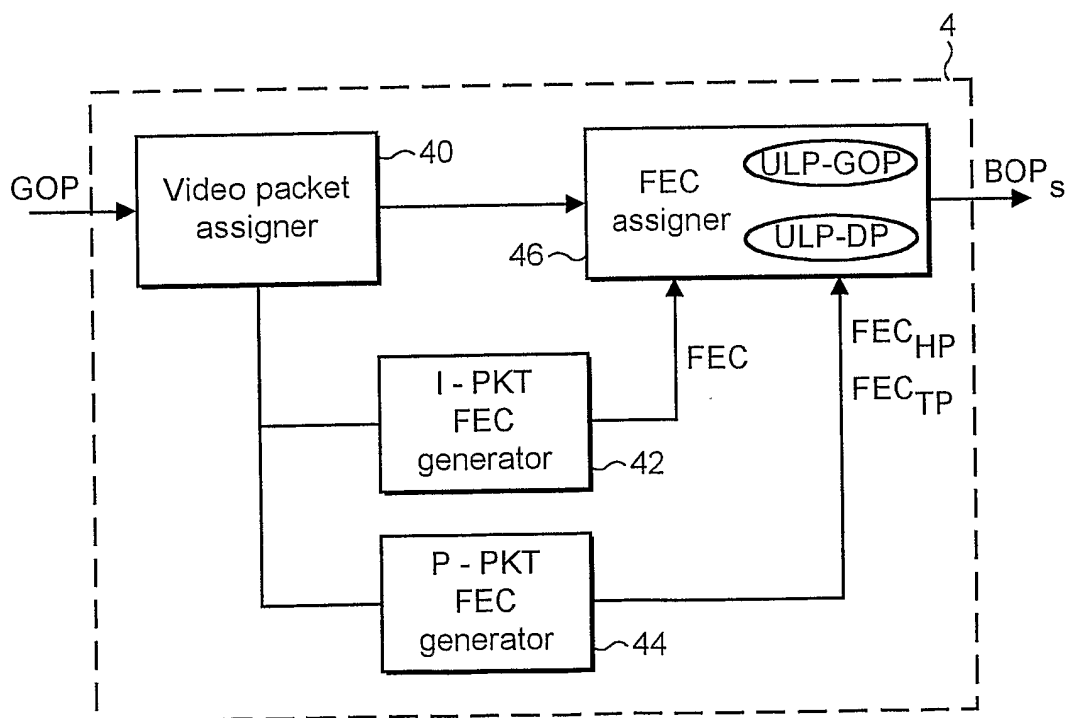


FIG. 2C

4 / 6

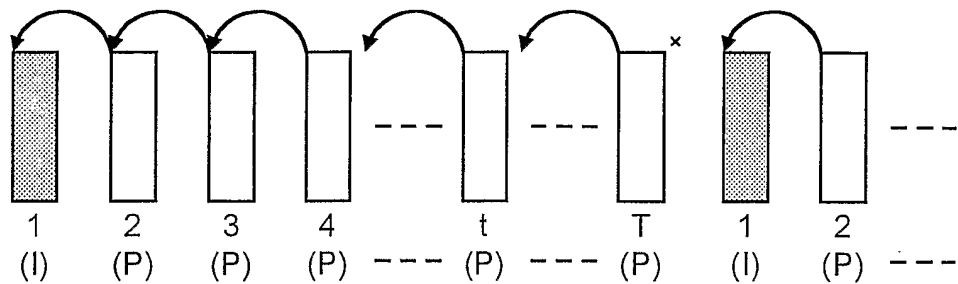


FIG. 3

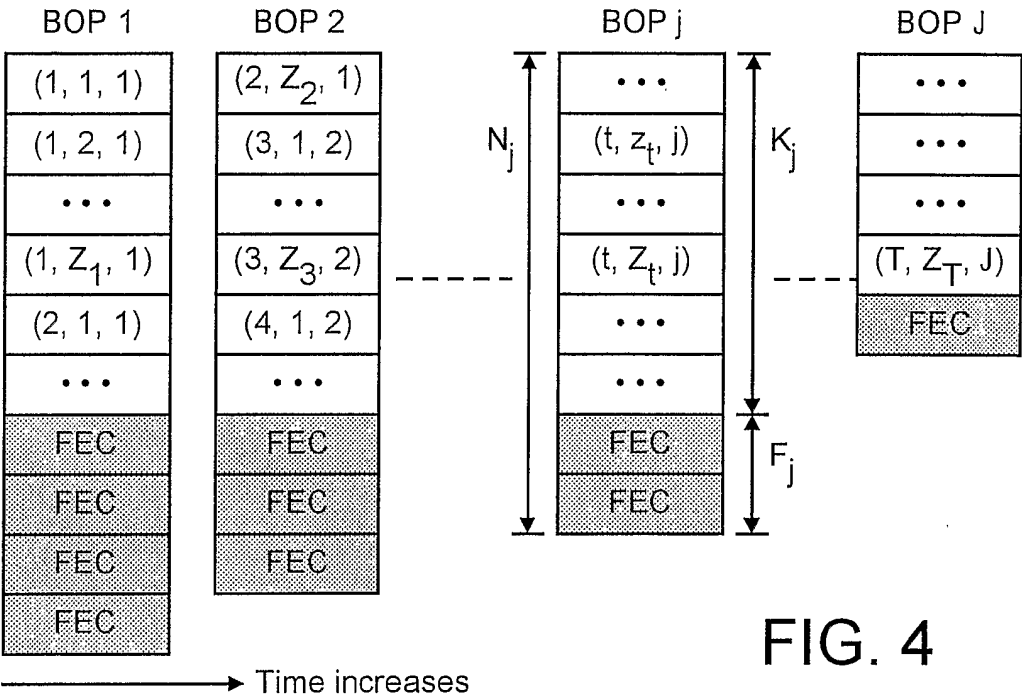


FIG. 4

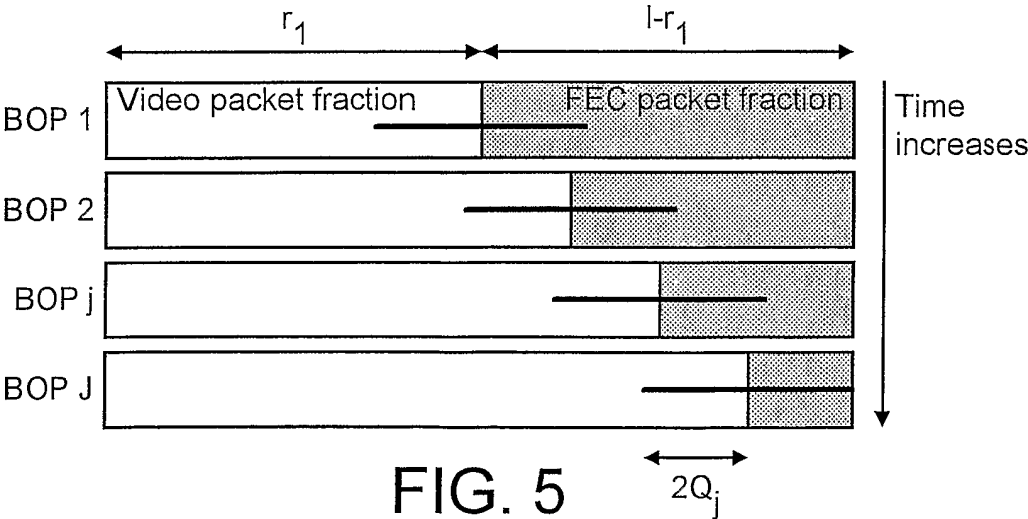


FIG. 5

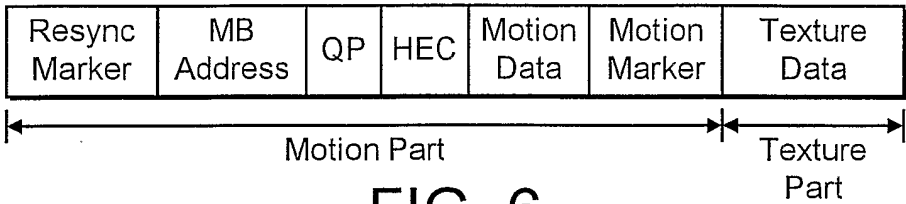


FIG. 6

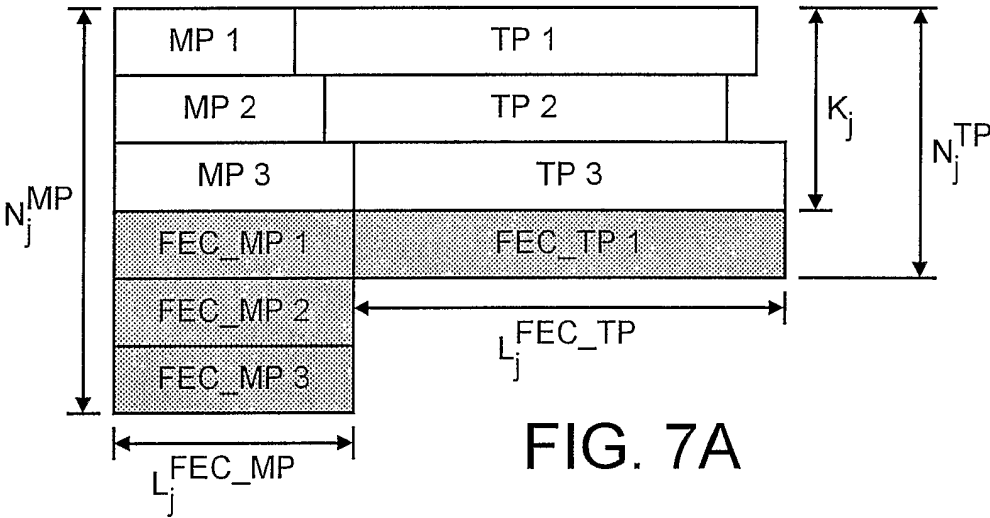


FIG. 7A

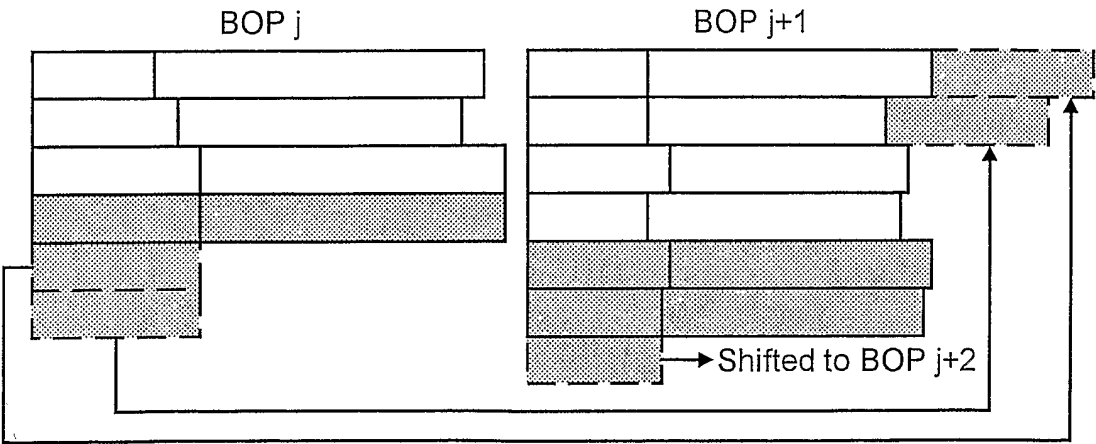


FIG. 7B

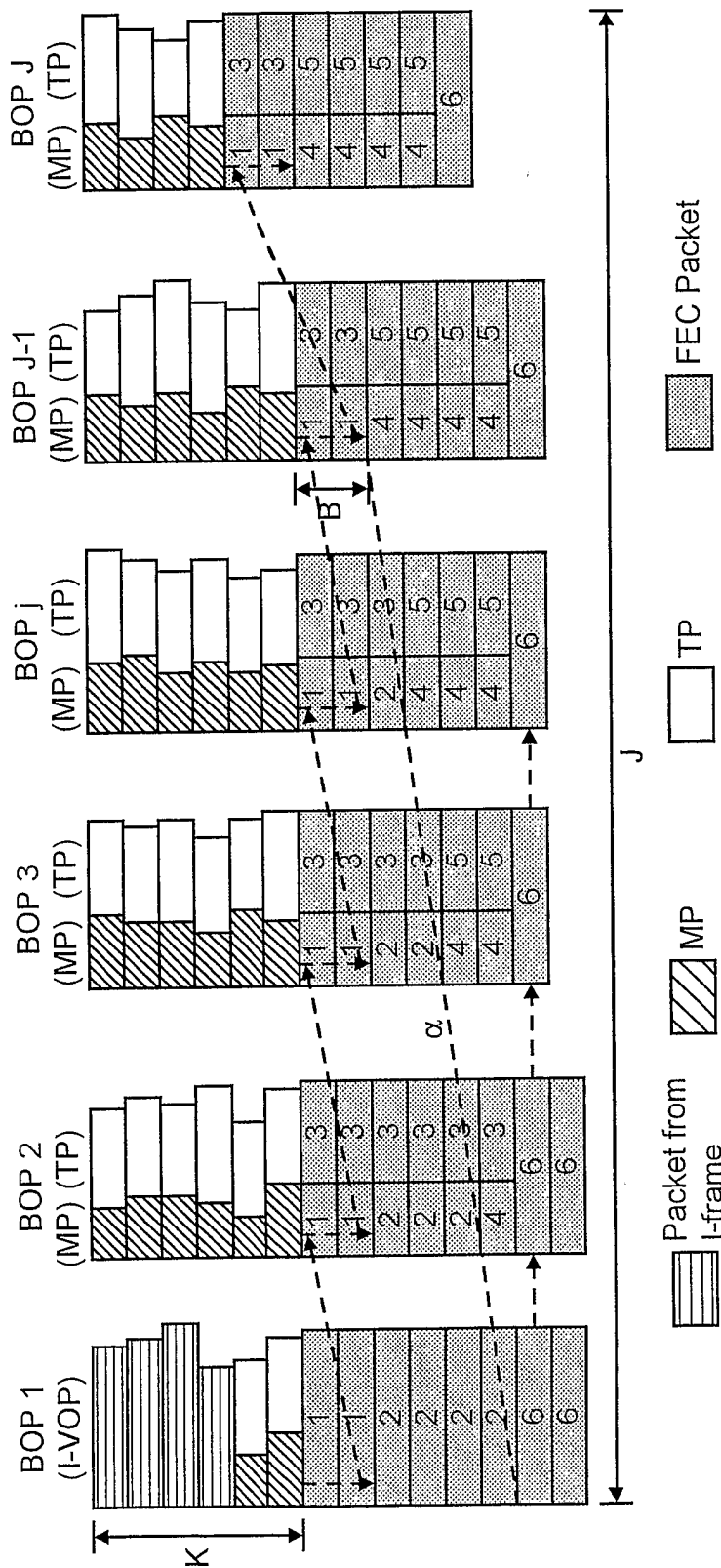


FIG. 8