Efficient transmission of video over highly error-prone channels

Mohammed Ghanbari FIEEE and Martin Fleury MIEEE

School of Computer Science and Electronic Engineering
University of Essex, Colchester, UK

Transmission of video over highly error-prone channels demands better protection of the compressed bitstream to ensure an acceptable quality of service. Many short distance error-prone channels use Automatic Repeat Request (ARQ), for retransmission of corrupted data [1]. Others suggest multiple description coding of video with spatio-frequency diversity transmission [2]. Some argue that layered coding with unequal error protection can guarantee a minimum picture quality [3]. However, in wireless channels with bursts of errors, in which those burst occur frequently due to slow and fast fading, the above techniques on their own may not be able to provide acceptable video quality. Of course, if the vital elements of layered encoded video can be efficiently protected, then such a technique can be fruitful.

One way to protect the vital video elements is by forward error correction such that they are almost guaranteed to be received safely. However, conventional fixed-rate control may not be suitable, as under varying channel conditions, the data may either be over- or under-protected. The former reduces the transmission efficiency and the latter may still not make the received data useable.

It appears that rateless or Fountain channel coding of video [4] is a simple and efficient method to protect the variable degree of video tolerance to errors that is required. In this coding method, a varying degree of redundancy is incrementally added to a group of symbols, to ensure that the symbols can be decoded under any adverse channel conditions. Thus, unlike fixed-rate Reed-Solomon (RS) codes, the coding rate is not fixed at the time of coding but can be dynamically varied. The degree of redundancy depends on channel error severity and, after a feedback request, symbols are gradually transmitted to the receiver, until the delay limit is exhausted. This not only makes transmission efficient, it also ensures the received data is decodable.

The class of Fountain codes [4] allows a continual stream of additional symbols to be generated in the event that the original symbols could not be decoded. It is the ability to easily generate new symbols that makes Fountain codes rateless. For a group of k symbols, decoding will succeed with small probability of failure if any of $k\ (1 + \varepsilon)$ symbols are successfully received, where ε is a small value, typically 0.05 for video applications. The probability of decoder failure is $\partial = 2^{-k\varepsilon}$, which for large k approaches the Shannon limit. Luby transform (LT) codes [5] reduce the complexity of decoding a simple Fountain code (which is of order k^3) by means of an iterative decoding procedure, provided that the column entries of the generator matrix are selected from a robust Soliton distribution. In the LT generator matrix case, the expected number of degree one combinations (no XORing of symbols) is $S = c \log_e(k/\partial)\sqrt{k}$, for small constant c. Setting $\varepsilon = 2 \log_e(S/\partial)\, S$ ensures that, by sending $k(1 + \varepsilon)$ symbols, these are decoded with probability $(1 - \partial)$ and with decoding complexity of order $k \log_e k$. Notice that the essential differences between Fountain erasure codes and RS erasure codes are that: Fountain codes in general (not Raptor codes [6]) are not systematic; and that even if there were no channel errors there is a small probability that the decoding will fail. In compensation, they are completely flexible; the Raptor variety has linear decode computational complexity; and generally their overhead is considerably reduced compared to fixed erasure codes. It should be noted that, compressed video unlike alpha-numeric data does not need to be perfectly decoded and such a small probability of failure is acceptable for most video applications. The symbol can be defined as a packet, block, byte or bit. By defining the symbol to be as small as feasible (a block of bits, byte, or bit) k is automatically made larger.

When one considers that the amount of redundancy added to the compressed video is proportional to the length of the compressed bit-stream, k, then, for efficient rateless coding only the important elements of video may need to be protected. For instance, through layering techniques, the most important parts of the compressed video bit-stream such as motion vectors (MVs), macroblock (MB) addresses, etc., can be protected such that the delivery of a reduced video quality stream at its lowest possible bit rate can be guaranteed. Scalable video coding is a good example, whereby, through a combination of quality, spatial and temporal scalability, a base-layer video at very low bit rates requires a minimal amount of redundancy through rateless channel coding to be effectively decoded. Another example is data-partitioned video, which, unlike scalable video coding, does not impose any overhead and is as efficient as can be. Data-partitioning (DP) is now an essential part of the Extended Profile of the H.264/Advanced Video Codec (AVC) [7]. In the following, we show how data-partitioned video can efficiently combine with rateless codes and how the same concept can be extended to scalable video.

H.264/AVC conceptually separates the Video Coding Layer (VCL) from the Network Abstraction Layer (NAL). The VCL specifies the core compression features, while the NAL facilitates the delivery of the H.264 VCL data to the underlying protocols such as RTP/UDP/IP, H.32X, or MPEG-2 transport stream. Each NAL unit could be considered as a packet that contains an integer number of bytes including a header and payload. The header specifies the NAL unit type and the payload contains the related data. The standard defines 31 different NAL unit types, when only NAL units 1 to 5 contain different VCL data [7] that will be of interest to this Letter.

The slices of an Instantaneous Decoder Refresh- (IDR-) or I-picture (i.e. a picture with all intra slices) are located in type 5 NAL units, while those belonging to a non-IDR or non-I-picture (P- or B-pictures) are placed in NAL units of type 1, and in types 2 to 4 when DP mode is active.

In type-1 and type-5 NALs, MB addresses, MVs and the transform coefficients of the blocks, are packed into the packet in the order that they are generated by the encoder. In type-5, all parts of the compressed bitstream are considered to be equally important, while in type-1, the MB addresses and MVs are much more important than the motion compensated DCT coefficients. In the event of errors in this type of packet, the fact that variable length encoded symbols appearing earlier in the bit-stream suffer less from the impact of errors than those which come later means that bringing the more important parts of the video data (such as header data and MVs) ahead of the less important data or separating the more important data altogether for better protection against errors can significantly reduce the impact of channel errors. It is worth noting that in the standard video codecs prior to H.264 [8], partitioning of the DCT coefficients within the *same* coding unit, which is a type of layered coding, is known as data partitioning. However, in more general terms, DP now refers to partitioning of the coded video stream into its degrees of importance.

However, in H.264, when DP is enabled, every slice is divided into three separate partitions and each partition is located in either a type-2, type-3 or type-4 NAL unit. A NAL unit of type-2, also known as partition A, comprises the most important information of the compressed video bit-stream for P- and B-pictures, including the MB addresses, MVs and essential headers. If any MBs in these pictures are intra-coded, their DCT coefficients are packed into a type-3 NAL unit, also known as partition B. Type-4 NAL, also known as partition C, carries the DCT coefficients of the motion-compensated inter-picture-coded MBs. As in I-slices all MBs are encoded then type-5 NAL units are very long. On the other hand, the A and B partitions of data-partitioned P- and B-slices are much smaller but their C-type partitions can be very long. In this Letter we propose an efficient method for rateless coding of A, B and C type NAL units to make video transmission more efficient.

For I-slices, a type-5 NAL can be Raptor-coded [6] with redundant information D, as shown in Fig. 1a. For data-partitioned P- and B-pictures, partitions A and B can be Raptor-coded with redundant D, as shown in Fig 1b. For partition C of P- and B-slices, a separate Raptor-code can be applied to the A, B and C partitions with redundant data in E, as shown in the Figure. Notice that this strategy is more efficient than were redundant E only to be generated for partition C. The reason is that, as the probability of decoder failure $\delta=2^{-k\varepsilon}$, including safely-detected A+B in C increases the length of data k, reducing the probability of failure. One way to ensure safe delivery of video is as follows:

For every k data symbols within the partition groupings (type-5, A+B, or A+B+C), a Raptor coder generates a rateless redundant data of r symbols. The compressed data can be partitioned into blocks of bits to form a symbol, as in theory r can be infinitely long to ensure all k data symbols can be safely decoded. For transmission purposes, each packet comprises K blocks of data, and the first Y blocks of their redundant data r are sent at the position of Y in Fig 1c. The packet also includes a Cyclic Redundancy Code (CRC) calculated from the K blocks or symbols. Recalculation of the CRC at the receiver and comparison with the sent CRC indicates whether the data decode was successful. In case of error, the transmitted data are stored and in the following packet additional redundant blocks of r, identified by X in Fig. 1c are sent. These new redundant blocks will help to decode the failed decoding and if the decoder still is not able to decode, more redundant blocks in the following packets will be sent. The process is continued, until the block is safely decoded.

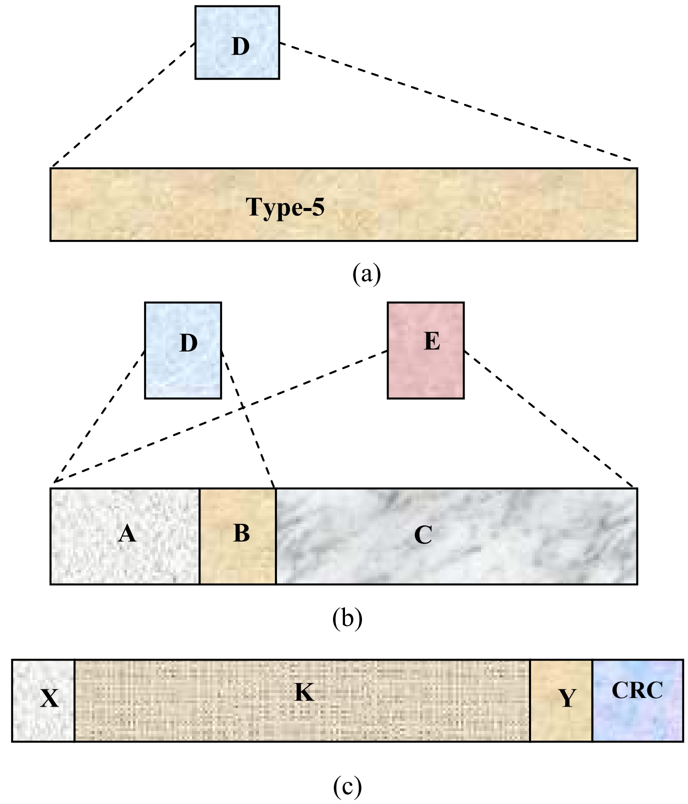


Figure 1. (a) I-slices and redundant data (b) Redundant codes for data-partitioned video (c) Packetized rateless coded data with CRC

Of course, for a delay-sensitive service such as video, transmission of additional redundant blocks cannot go on for ever, and there should be a limit. For instance, one could confine the decoding delay to be within a certain number of pictures (e.g. 15 pictures, equal to approximately half a second at 30 Hz.) To limit the number of transmissions of redundant blocks for previous data (X), the length of these blocks in the following packets can be gradually increased. For I-pictures/slices, when the length of type-5 NALs can be very long, the length of redundant code r is much longer than that of P-pictures. Fortunately, in the H.264 standard there are several B pictures/slices after each I-picture, and instead of transmission of B-pictures, one may just send the redundant D blocks of I-pictures. This is because, B-pictures can be easily discarded without significantly impairing video quality. For P- and B-pictures, since A+B is very small, their number of redundant blocks is also small, and can be easily decoded in a few following packets. For partition type C, though such partitions can be long, since their impact on picture quality is negligible and, as already mentioned, their decoder failure probability is small, they can be easily sacrificed in favor of sending the redundant blocks belonging to previous A+B blocks. Where no B-pictures are used, P-picture C type NALs can also be replaced by the redundant data for type-5 I-pictures. This procedure will significantly reduce the decoding delay under extremely adverse channel conditions.

Our simulations indicate that use of rateless coding on data-partitioned video can improve the video quality by 2-3 dB depending on video content. We believe, similar quality gains could be achieved if rateless coding were to be also applied to scalable coded video, as efficient transmission of scalable coded video greatly depends on the safe delivery of base layer pictures.

References

[1] M. Zorzi and R. R. Rao, "Error control and energy consumption in communications for nomadic computing," *IEEE Trans. on Computers*, vol. 46, no. 3, pp. 279–289, 1997.

[2] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. of the IEEE*, vol. 93, no. 1, pp. 59-70, 2005.

[3] M. M. Ghandi and M. Ghanbari, "Layered H.264 video transmission with hierarchical QAM," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp 451-466, April 2006.

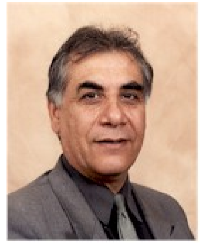
[4] D. J. C. MacKay, "Fountain codes," *IEE Proceedings: Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[5] M. Luby, "LT codes," *in Proc. of the 34rd Annual IEEE Symp. on Foundations of Computer Science*, pp. 271–280, Vancouver, Canada, Nov. 2002.

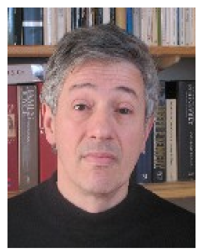
[6] A. Shokorallahi, "Raptor codes", *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2551-2567, 2006.

[7] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, , pp. 645-656, July 2003.

[8] M Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, IET Press, Stevenage, UK, 2003.



Mohammed Ghanbari is best known for his pioneering work on two-layer video coding for ATM networks (which earned him an IEEE Fellowship in 2001), now known as SNR scalability in the standard video codecs. He has served as an Associate Editorto IEEE Trans. on Multimedia (IEEE-T-MM from 1998-2004) He has registered for eleven international patents on various aspects of video networking and was the co-recipient of A.H. Reeves prize for the best paper published in the 1995 Proc. of IEE on the theme of digital coding. He is the co-author of "Principles of Performance Engineering", a book published by IET press in 1997, the author of "Video Coding: An Introduction to Standard Codecs", a book also published by IET press in 1999, which received the year 2000 best book award by the IEE, and the author of "Standard Codecs: Image Compression to Advanced Video Coding" also published by the IET press in 2003. Prof. Ghanbari has authored or co-authored about 450 journal and conference papers, many of which have had a fundamental influence in this field.



Martin Fleury (M'08) is a holds a degree in Modern History (Oxford University, UK) and a Maths/Physics based degree from the Open University, Milton Keynes, UK. He obtained an MSc in Astrophysics from QMW College, University of London, UK in 1990 and an MSc from the University of South-West England, Bristol in Parallel Computing Systems in 1991. He holds a PhD in Parallel Image Processing Systems from the University of Essex, Colchester, UK. He is currently employed as a Senor Lecturer at the University of Essex. Martin has authored over one hundred and forty articles and a book on high-performance computing for low-level image- and signal-processing algorithms (including document and image compression algorithms), performance prediction of parallel systems, software engineering, and vision systems. His current research interests are video communication over MANS, WLANs, PANs, BANs, MANETs, and VANETs.