VIDEO CODING WITH WAVELET-DOMAIN CONDITIONAL REPLENISHMENT AND UNEQUAL ERROR PROTECTION

James E. Fowler[†], Marco Tagliasacchi[‡], and Béatrice Pesquet-Popescu[*]

[†]*Dept. of Electrical & Computer Engineering*
GeoResources Institute
Mississippi State University, MS USA

[‡]*Dipartimento di Elettronica e Informazione*
Politecnico di Milano, Italy

[*]*ENST Paris*
Paris, France

ABSTRACT

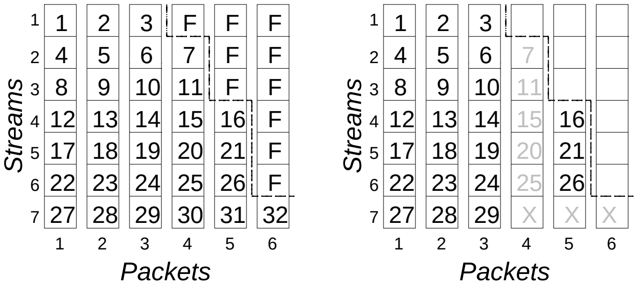
A simple and computationally lightweight video coder employing shape-adaptive, embedded intraframe coding and wavelet-domain conditional replenishment is proposed. Robustness to packet losses arises from packetization of the embedded bitstream with unequal error protection which is assigned to the packets with a fast, locally optimal procedure. Experimental results reveal that, when compared to H.264/AVC configured for low-complexity, error-resilient operation, not only does the proposed coder usually produce substantially superior rate-distortion performance as packet losses increase, it also achieves a significantly faster encoding speed.

Index Terms— Conditional replenishment, robust video coding, shape-adaptive wavelet transform

1. INTRODUCTION

In packet-based multimedia communications, unequal error protection (UEP) is a framework for forward error correction (FEC) that assigns error-protection codes to bitstreams such that the most important information receives the greatest protection in order to provide graceful degradation of reconstruction quality as packet losses increase. Bitstreams that result from embedded coders are particularly well-suited to UEP since an embedded bitstream is inherently organized in the order of decreasing importance, and there exist several algorithms (e.g., [1, 2]) for the design of UEP FECs that attempt to maximize the reconstruction quality of an embedded bitstream for a given probabilistic packet-loss model.

In this paper, we propose a simple application of UEP to video coding in order to combat packet losses. Additionally, we focus on simple implementation and lightweight computational complexity such as may be suited to portable wireless devices. To do so, we couple conditional replenishment (CR) with an embedded wavelet-based still-image coder. In CR, blocks that do not differ significantly from those of the previous frame (i.e., "skip" blocks) are simply replenished from the preceding frame at the decoder, while the other blocks are intraframe coded. In our technique, CR is deployed in the domain of a discrete wavelet transform (DWT) such that the skip blocks result in "holes" in the wavelet coefficients of the current frame. Consequently, we employ a shape-adaptive (SA) intraframe coder that effectively codes around the coefficients of the skipped blocks. For such SA intraframe coding, we use the BISK algorithm [3] which has been shown to consistently yield state-of-the-art SA coding; thus, the resulting video-coding system can be considered to be "wavelet-domain CR BISK" (WCR-BISK). Error robustness in the WCR-BISK system comes from packetizing the embedded SA-BISK bitstream with UEP FECs. Due to its lack of motion estimation and compensation (ME/MC), the WCR-BISK codec is rather lightweight computationally as well as fairly simple to implement.



Streams	1	2	3	4	5	6
1	1	2	3	F	F	F
2	4	5	6	7	F	F
3	8	9	10	11	F	F
4	12	13	14	15	16	F
5	17	18	19	20	21	F
6	22	23	24	25	26	F
7	27	28	29	30	31	32

Packets

(a)

Streams	1	2	3	4	5	6
1	1	2	3			
2	4	5	6	7		
3	8	9	10	11		
4	12	13	14	15	16	
5	17	18	19	20	21	
6	22	23	24	25	26	
7	27	28	29	X	X	X

Packets

(b)

Figure 1: Example packet arrangement for UEP of an embedded bitstream. (a) Transmitted packets; (b) data recovery after packet 4 is lost. Data bytes numbered according to their occurrence in the embedded bitstream, F = FEC byte (adapted from [1]).

We note that CR has been proposed in conjunction with DWTs in the past. For example, in [4], DWT-domain blocks are skipped via CR; however, unlike our DWT-domain CR approach, the skip decision is made in the spatial domain, and zerotrees are used for the coding of the blocks rather than SA intraframe coding. DWT-domain CR similar to our approach is used in [5]; however, vector quantization is used for the intraframe coding. Finally, SA intraframe coding is used to code non-skip blocks in [6]; however, traditional ME/MC is used rather than CR. None of these prior techniques include FECs for error resilience.

Below, we first overview the application of UEP to embedded bitstreams before describing our WCR-BISK system in detail. We then present experimental results that reveal superior rate-distortion performance under packet losses for WCR-BISK as compared to H.264/AVC, the state of the art in ME/MC-based video coding. Additionally, we find that the WCR-BISK encoder runs significantly faster despite a low-complexity H.264/AVC encoder configuration.

2. UEP FOR EMBEDDED BITSTREAMS

In the UEP of embedded bitstreams, FECs and data bytes form a "stream" with the number of streams equal to the number of bytes in each packet to be transmitted. An example of this arrangement is illustrated in Fig. 1(a). Since the bitstream is embedded, earlier parts of the bitstream are more important than latter parts; thus, a

This work was funded in part by the French Centre National de la Recherche Scientique and by the US National Science Foundation under Grant No. CCR-0310864.

in *Proceedings of the International Conference on Image Processing*, Atlanta, GA, October 2006, vol. 2, pp. 1869–1872.

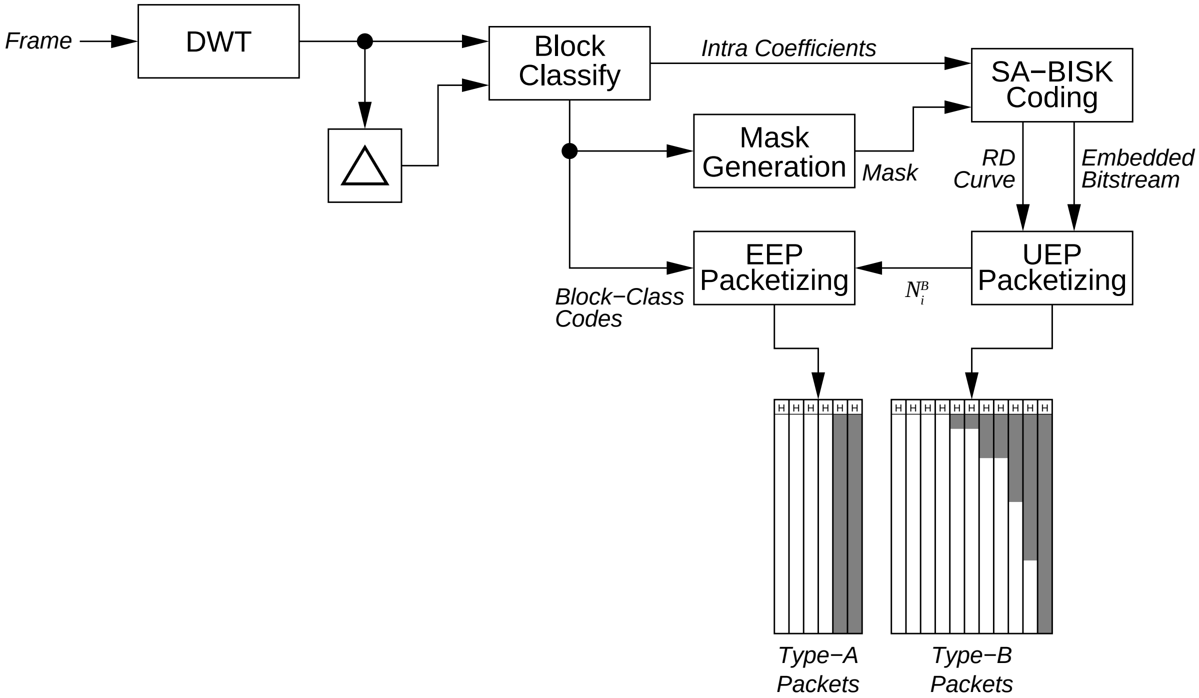


Figure 2: The WCR-BISK encoder. Δ indicates frame delay; gray shading indicates FECs; H indicates RTP packet headers; N_i^B indicates the number of UEP FECs in stream i of the type-B packets.

greater number of FECs are assigned to earlier streams than to latter streams. Algorithms such as [1, 2] attempt to arrange the FECs so as to maximize the expected PSNR of the reconstruction subject to the loss model. The tenet central to these UEP algorithms is that all the bytes of a stream can be recovered if the number of packets lost is less than or equal to the number of FEC bytes in that stream; such is the case when Reed-Solomon codes are applied to each stream to generate its FEC bytes. For example, if packet 4 in Fig. 1(a) is lost while the other packets are received, the initial 29 bytes of data can be recovered after inverting the FEC code, as illustrated in Fig. 1(b). (Even though bytes 31 and 32 are correctly received, they cannot be decoded without byte 30, which cannot be recovered.) In general, we assume that N_i FECs are assigned to stream i with $N_i \geq N_{i+1}$, and stream i will be recovered as long as N_i or fewer packets are lost. We note that $N_i = N_{i+1}, \forall i$, corresponds to the special case of equal error protection (EEP) in which all streams are protected with equal FEC strength.

Finally, we note that UEP algorithms for embedded bitstreams are generally driven by a rate-distortion curve in the form of the PSNR as a function of the received bytes of the embedded bitstream. Such a rate-distortion curve is easily generated simultaneously with the embedded bitstream during encoding.

3. WCR-BISK

The WCR-BISK encoder, depicted in Fig. 2, operates as follows. A 3-scale DWT is applied to the current frame, and the resulting DWT coefficients are partitioned into 8×8 cross-scale blocks as illustrated in Fig. 3. Each DWT-domain block is then classified into one of two block classes—`skip` blocks or `intra` blocks. This classification is based on the mean squared error (MSE) between the current DWT-domain block and the co-located block in the (uncoded) DWT of the previous frame. If the MSE is small, the block is classified as a `skip` block. Otherwise, it is classified as an `intra` block. A preset threshold determines this block classification, and the block class for each block is passed to the decoder as a single-bit code.

For `skip` blocks, none of the wavelet coefficients of the block are coded. On the other hand, for `intra` blocks, all coefficients in the block are coded following the embedded bitplane-coding methodology common to wavelet-based still-image coders. We employ the SA-BISK algorithm [3] as a SA embedded intraframe coder. In SA-BISK, only those coefficients within an arbitrarily shaped image "object" are coded while regions not belonging to the object are ignored. In the present video-coding context, the image "object" to be coded is the set of `intra` coefficients, while the `skip` coefficients constitute the non-object regions of the image. A mask constructed from the block classes indicates the object and non-object regions to the SA-BISK intraframe coder. We note that, in SA image coding, one employs a SA DWT [7] to transform the image object into the wavelet domain before applying the embedded SA coder; a SA DWT would be similarly employed for spatial-domain CR. In WCR-BISK, CR takes place in the DWT domain, such that the mask and object coefficients are constructed directly in the wavelet domain, and no SA DWT is employed.

After encoding of a frame, the resulting bitstream is packetized into 100-byte RTP packets, each consisting of a 12-byte RTP header [8] plus an 88-byte payload. The RTP packets are one of two types—type-A packets or type-B packets. The embedded bitstream resulting from the SA-BISK coder is placed in the type-B packets with UEP FEC protection. The SA-BISK coder generates a rate-distortion curve for the current frame simultaneously with its embedded bitstream; this rate-distortion curve is passed to the algorithm of [1] to generate a locally optimal UEP FEC arrangement for the embedded bitstream. This algorithm determines N_i^B, the number of FECs assigned to stream i of the B packets, for each i. The block-class codes, along with the N_i^B values, are placed in the A packets which are then protected with N^A packets of EEP FECs. Systematic Reed-Solomon codes are used for all FECs. The packets for the current frame are then transmitted in the order of type-A packets followed by type-B packets, and then the entire encoding and packetization processes are repeated for the next frame. The type of the packet—A or B—is specified using the payload-type (PT) field of the RTP header [8].

During decoding, the sequence number from the RTP header is used to determine which packets were lost in transmission. The current frame is reconstructed from the longest bitstream prefix that can be extracted from the received B packets after inverting the UEP FECs to recover as many of the missing B packets as possible. The block-class codes in the A packets are used to reconstruct the map of `skip`/`intra` blocks as needed for SA-BISK decoding. We note that, if greater than N^A type-A packets are lost, no frame can be reconstructed, since SA-BISK decoding, as well as correct interpretation of the B-packet UEP arrangement, depends on perfect recovery of the A-packet information in its entirety. In this case, the WCR-BISK decoder simply outputs the previously decoded frame in lieu of the current frame.

4. RESULTS

For WCR-BISK with CIF/SIF frames, there are 4 type-A packets of which $N^A = 1$ packet is EEP FEC. The algorithm of [1] assigns a locally optimal UEP FEC arrangement for the type-B packets. The UEP arrangement is optimized for an expected packet-loss rate of 10%, regardless of the actual packet-loss rate of the channel which we assume the encoder does not know. We have found that mismatches between the packet-loss rate for which the UEP arrangement is designed and the actual loss rate does not significantly affect the performance results.

The use of H.264/AVC over packet networks is discussed extensively in [9, 10]. We use JM 10.1 in extended profile and configure the encoder for low-complexity operation. Namely, we use a single reference frame, no B-slices, no RD-optimized mode decision, early skip-decision enabled, selective intra-mode decision enabled, and EZPS motion search. Furthermore, we have enabled several of the H.264/AVC error-resilience tools such as random intra-macroblock refresh and constrained intra prediction, although we do not use redundant slices, flexible macroblock ordering, or forced intra-GOB refresh. The output bitstream is constructed using a slice mode of 100 bytes per slice and then packetized into RTP packets. The H.264/AVC decoder uses motion-copy error concealment to compensate for lost packets.

We average PSNR over 100 trials and over all frames of the sequence at a given packet-loss rate, and the same packet-loss patterns are used for each coder. In Table 1, we compare the performance of our WCR-BISK scheme to H.264/AVC for a fixed packet-loss rate for bitstreams at three encoding bitrates (634 kbps, 1.27 Mbps, and 2.53 Mbps), while in Figs. 4–6, we plot average PSNR for a fixed encoding bitrate as the packet-loss rate varies.

In Table 1, we see that, for the highest-rate encoding, WCR-BISK outperforms H.264/AVC by a significant margin, up to nearly 8 dB for the "mother-daughter" sequence. WCR-BISK also yields average PSNR superior to that of H.264/AVC for all but one of the sequences for the 1.27-Mbps encoding. For the lowest-rate encoding, the results are more mixed, with WCR-BISK outperforming H.264/AVC for most of the sequences. In Figs. 4–6. we see that, although H.264/AVC yields substantially higher PSNR when the transmission channel is lossless, its performance tends to drop quickly as the packet-loss rate increases so that WCR-BISK maintains a significant advantage for all but the lowest packet-loss rates.

Finally, we note that, although we have configured H.264/AVC for low-complexity encoding for the results presented here, our WCR-BISK encoder implementation runs significantly faster due to its lack of ME/MC. Specifically, on a Pentium M 2-GHz machine with 1 GB of RAM, our WCR-BISK implementation encodes at 6.02 frames per second (fps) at CIF/SIF resolution, while the H.264/AVC encoder achieves only 2.15 fps, nearly three times slower. However, no efforts have been made to optimize the WCR-BISK coder for speed; several straightforward speedups (e.g., using fixed-point, rather than floating-point, arithmetic for wavelet transforms) should produce an even faster WCR-BISK coder.

5. CONCLUSIONS

In this paper, we have investigated a simple video coder that couples embedded SA intraframe coding with wavelet-domain CR. Experimental results demonstrate favorable performance as packet losses increase as compared to H.264/AVC in a low-complexity, error-resilient configuration. It is anticipated that, with its lightweight computational complexity and superior error resilience, the proposed WCR-BISK coder may be attractive for resource-constrained wireless devices communicating over error-prone links.

6. REFERENCES

[1] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Jour. Selected Areas in Comm.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[2] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proceedings of the* 33^{rd} *Asilomar Conf. Sig., Sys., and Comp.*, vol. 1, Pacific Grove, CA, Oct. 1999, pp. 342–346.

[3] J. E. Fowler, "Shape-adaptive coding using binary set splitting with k-d trees," in *Proc. ICIP*, vol. 2, Singapore, Oct. 2004, pp. 1301–1304.

[4] S. McCanne and M. Vetterli, "Joint source/channel coding for multicast packet video," in *Proc. ICASSP*, vol. 1, Washington, DC, Oct. 1995, pp. 25–28.

[5] W. Namgoong, N. Chaddha, and T. H. Y. Meng, "Low-power video encoder/decoder using wavelet/TSVQ with conditional replenishment," in *Proc. ICASSP*, vol. 6, Atlanta, GA, May 1996, pp. 3240–3243.

[6] K. Cinkler, "Very low bit-rate wavelet video coding," *IEEE Jour. Selected Areas in Comm.*, vol. 16, no. 1, pp. 4–11, Jan. 1998.

[7] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrary shaped visual object coding," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 10, no. 5, pp. 725–743, Aug. 2000.

[8] H. Schulzrinne, S. Casner, R. Federick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, 1996.

[9] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 13, no. 7, pp. 645–656, Jul. 2003.

[10] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 13, no. 7, pp. 657–673, Jul. 2003.

Table 1: Average PSNR (dB) for a packet-loss rate of 10%.

	634 kbps (0.25 bpp)			1.27 Mbps (0.5 bpp)			2.53 Mbps (1.0 bpp)		
	WCR-BISK		H.264/AVC	WCR-BISK		H.264/AVC	WCR-BISK		H.264/AVC
	skipped	PSNR	PSNR	skipped	PSNR	PSNR	skipped	PSNR	PSNR
Sequence	blocks	(dB)	(dB)	blocks	(dB)	(dB)	blocks	(dB)	(dB)
football†	42.2%	**25.0**	22.7	30.3%	**26.5**	23.3	19.3%	**28.1**	23.7
nyc†	31.3%	26.9	**27.4**	16.7%	**29.5**	28.0	3.2%	**32.1**	28.2
susie†	45.3%	**32.3**	32.2	39.1%	**34.9**	32.7	22.0%	**38.2**	32.9
foreman	38.9%	**30.2**	28.6	28.3%	**32.8**	28.9	15.3%	**35.1**	29.2
mother-daughter	78.3%	34.3	**35.1**	61.4%	**39.2**	35.3	49.1%	**43.1**	35.4
coastguard	35.0%	**26.3**	26.0	18.0%	**28.1**	28.0	3.5%	**30.4**	29.0
table-tennis	68.3%	27.0	**27.4**	58.8%	28.0	**28.3**	37.1%	**29.9**	29.0

Sequences are grayscale, 16 frames long, CIF (352×288) at 25 Hz except †SIF (352×240), 30 Hz.
Rates are encoding rates and do not include packetization overheads such as RTP headers.
Skipped blocks are expressed as the average percentage of blocks encoded in `skip` mode per frame.

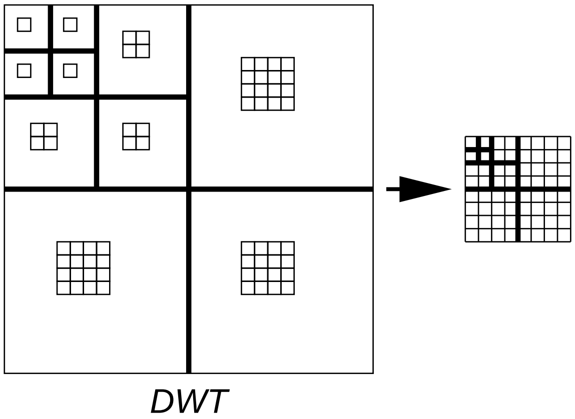


Figure 3: Extraction of blocks from the DWT.

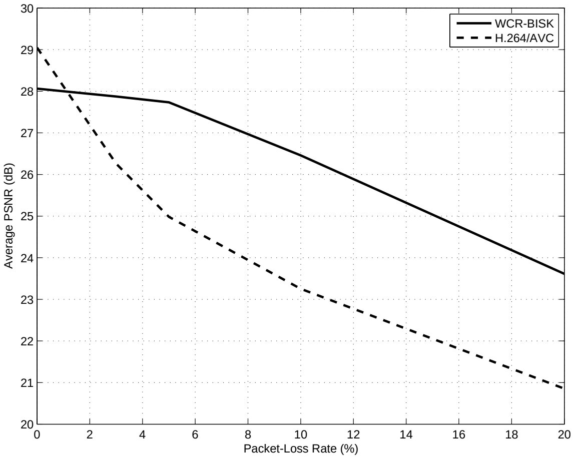


Figure 4: Average PSNR vs. packet-loss rate for “football” at 1.27 Mbps.

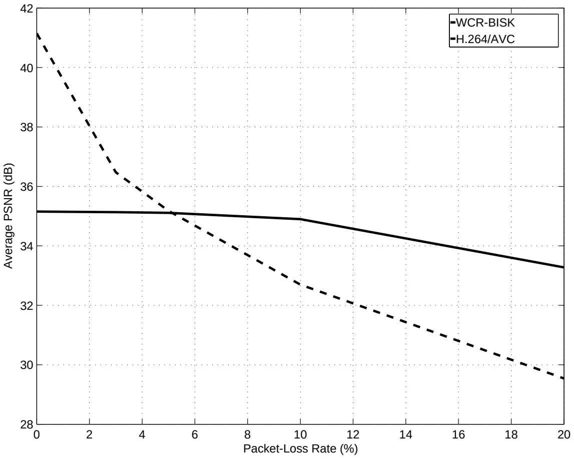


Figure 5: Average PSNR vs. packet-loss rate for “susie” at 1.27 Mbps.

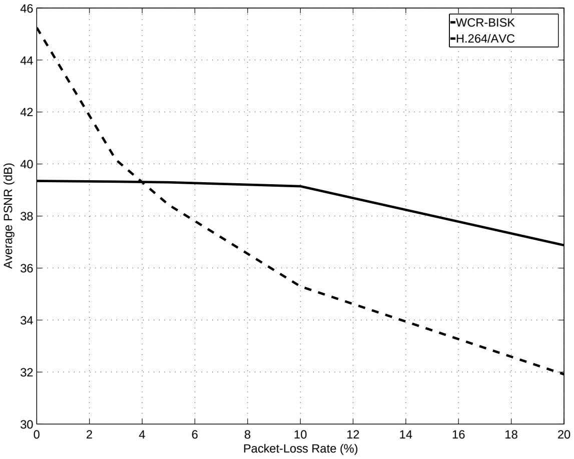


Figure 6: Average PSNR vs. packet-loss rate for “mother-daughter” at 1.27 Mbps.

in *Proceedings of the International Conference on Image Processing*, Atlanta, GA, October 2006, vol. 2, pp. 1869–1872.