1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

GOP-INTERLACED CODING SCHEME FOR ENHANCING COARSE-GRAIN PARALLELISM AND ERROR CONCEALMENT

Yen-Kuang Chen[‡], *Min-Tsong Chang*[+], *and S. Y. Kung*[*]

[‡]Intel Corp., Santa Clara, CA
[+]Fountain Technologies, Inc., Somerset, NJ
[*]Princeton University, Princeton, NJ

ABSTRACT

The main novelty of this work is redefining the structure of the group-of-picture (GOP). More specifically, in the current MPEG and H.263 video coding standards, the GOP is a group of *consecutive* frames and two GOPs have *no overlaps in time*. In our scheme, the GOP is a group of *non-consecutive* frames and the frames in two GOPs *interleave* with each other. There are many advantages of using the proposed encoding scheme, which interlaces the group-of-picture (GOP). First, the encoding scheme is easily paralleled in coarse-grain. So far, no state-of-the-art microprocessors can provide enough computational power for most real-time video compression standards. In order to bring the real-time video compression into microprocessor-based consumer products, the algorithms and the architecture need to be redesigned. The proposed GOP-interlaced coding scheme enables a linear-speedup of two in a dual-CPU system for real-time video compression. Second, it offers improved error-concealment capability. Because compressed image data comes with a problem of sensitivity to channel errors, error-concealment techniques are introduced to minimize the impact of these errors. The proposed GOP-interlaced coding scheme leads to a 5dB SNR improvement in our motion-interpolated error concealment over the original coding scheme.

Keywords: GOP-interlaced coding scheme, coarse-grain parallel processing, error concealment, video coding, video compression, real-time multimedia.

1. INTRODUCTION

A group of pictures (GOP) consists of a series of frames (I-frames, P-frames, and/or B-frames), as shown in Figure 1. In the coded bit-stream, the first coded frame in a GOP shall be I-frame. The I-frames can be encoded/decoded without referring to the other frames so as to assist random access into the video sequence. For example, applications that require fast-forward playback or fast reverse playback may use relatively short GOPs. On the other hand, P- or B-frames can only be encoded/decoded by referring to the previous decoded frames so as to maximize the compression ratio.

While an original GOP consists of an uninterrupted sequence of frames, we propose a new scheme in which the GOPs are interlaced. As shown in Figure 2, we let F_1, F_3, F_5, & F_7 (where F_1 means frame 1) to be GOP 1, and F_2, F_4, F_6, & F_8 to be GOP 2.

In this work, we propose a new encoding strategy that (1) enables short-delay coarse-grain parallel-processing the conventional video compression algorithms (e.g., MPEG, H.263), and (2) increases error concealment abilities in decoding process by redefining the structure of group-of-picture (GOP).

Real-time video encoding has various applications (e.g., video-phone, video-conference, and TV broadcasting) and requires a huge amount of computation. Although current microprocessors are faster than ever, no single CPU can provide enough computational power for most standard real-time video compression algorithms yet. In Section 2, we demonstrate that the proposed GOP-interlacing strategy offers the ease of coarse-grain parallelization of the conventional video compression algorithms (e.g., MPEG, H.263).

Compressed image data comes with a problem of sensitivity to channel errors. The phenomenon of channel errors is transformed into lost image blocks in the decoding process. The resulting image is flawed by the absence of square pixel regions that are noticeably perceived by human vision. As shown in Figure 3, the damage of blocks lost will be propagated until the end of the GOP. In order to minimize the impact of these errors, error resilience and error concealment techniques are introduced. In Section 3, we demonstrate that the GOP-interlacing strategy offers improved error concealment capability.

[‡]The work was conducted when Yen-Kuang Chen was with Princeton University. For further information, send E-mail to Yen-Kuang.Chen@intel.com.

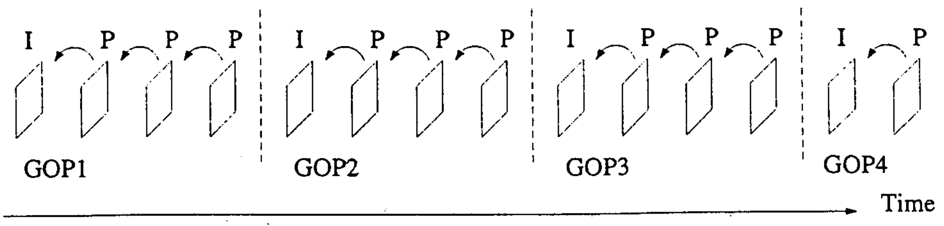

Figure 1: A group of pictures (GOP) consists of an uninterrupted series of frames. For example, frames 1–4 are in GOP 1, and frames 5–8 are in GOP 2. It starts with an I-frame, which can be encoded/decoded without referring to the other frames, so as to break the dependency in a sequence into some small groups. After that, it follows with some motion-compensated frames (e.g., P-frames, B-frames) so as to maximize the compression ratio. The letter I indicates I-frames. The letter P indicates P-frames. The arrow indicates the dependency between frames.

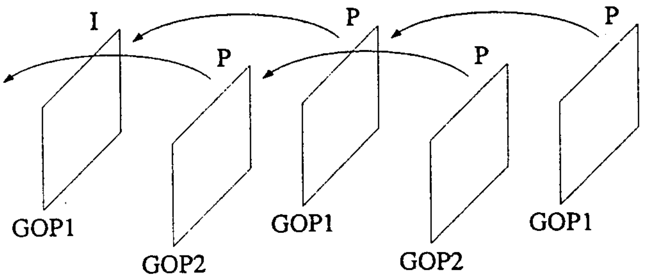

Figure 2: Instead of assigning frames 1–4 to GOP 1 and frames 5–8 to GOP 2, The GOP-interlaced scheme assigns frames 1, 3, 5, & 7 to GOP 1 and frames 2, 4, 6, & 8 to GOP 2. Consequently, the encoding of frame 4 no longer depends on frame 3, but depends on frame 2.

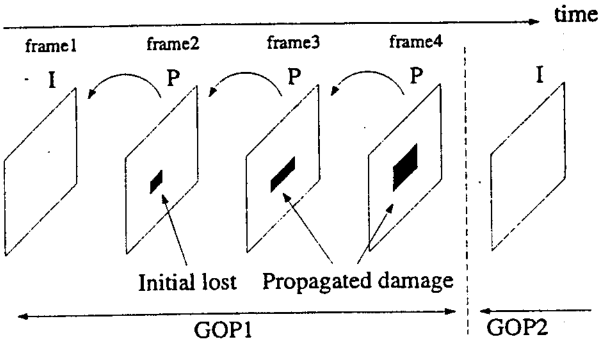

Figure 3: In most video compression standards, temporary redundancies are removed by the motion prediction from previous frame. Without error concealment, a damage in a block could be damages in the consequent prediction frames (P-frame) until the end of the group of pictures (GOP). However, the damage cannot cause any damages in other GOPs.

2. SHORTER-DELAY IN COARSE-GRAIN PARALLEL PROCESSING

In this section, we show an easy two-time speedup of the video encoding processing after we interlace the GOPs. We do not decrease the computational complexity of the encoding process nor exploit the fine-grain parallelism. We focus solely on microprocessor-level parallelism, and hence the algorithm is transformed due to the architecture constraint.

For successful systems, the inherent interaction of various design parameters comprising hardware and software issues must be taken into account. Algorithms should be designed to facilitate dedicated processing modules; architectures should be tailored to achieve higher efficiency for the special application domain [3]. Two types of approaches have been tried to bring the real-time video compression into consumer products, e.g., home PC. First, smarter algorithms are used to reduce computation power requirement, for example, Intel's Proshare [6]. Second, parallel and pipeline processing techniques (especially, fine-grain parallelism) are used to provide more computation power, for example, MAX-2 for HP's PA-RISC [7], VIS for SUN's UltraSparc [8], MMX for Intel's x86 [5]. However, limited approaches have been proposed for coarse-grain parallel architectures where there are two or more CPUs working together with expensive intercommunications.

Parallel processors, either customer designed (application-specific) or commercially available (general-purpose), are widely used for computationally intensive applications. There are two basic approaches for parallel processing. One is fine-grain parallelism; the other one is coarse-grain parallelism. Fine-grain parallelism is characterized by each of tasks containing a small amount of computation. Coarse-grain parallelism, on the other hand, is characterized by the fact that there is a huge amount computation in each task.

There are numerous methods that fine-grain parallel-compress the incoming video. Particularly, they parallel the operations within a GOP, e.g., SIMD operations for motion estimation and DCT. The latency of the tasks is short, and therefore the response time delay is short. However, since

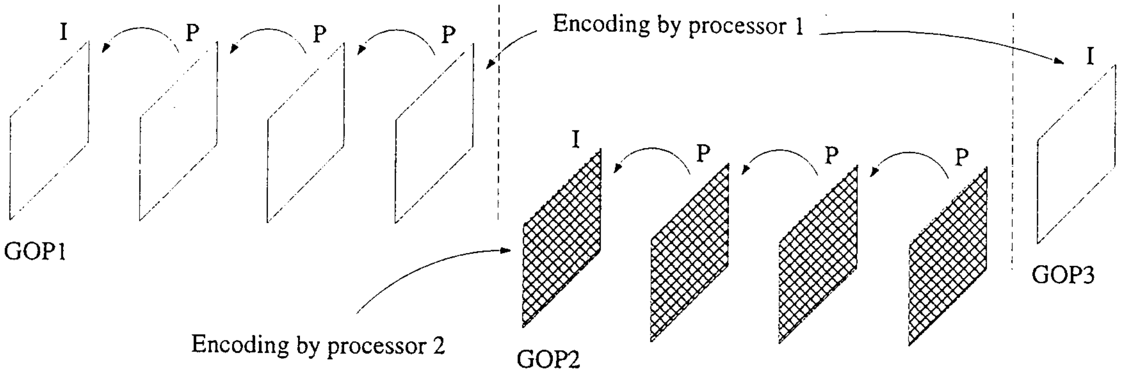

Figure 4: An approach toward parallel processing current video coding standards in coarse-grain. The task is divided by the unit of GOPs. The coding of a GOP is assigned to a processor. Since there is almost no dependency between GOPs, there is almost no communication necessary between processors. It is very easy to be implemented in a system with multiple general-purpose processors.

fine-grain parallel programs have high communication-to-computation-ratios, low communication cost is required in the systems. Therefore, fine-grain parallelism is often used in customer design processors where processing elements are closely tied together, e.g., systolic array.

Because of the high communication cost in current multi-microprocessor systems, coarse-grain parallelism must be exploited. There is also a method that can coarse-grain parallel-compresses the incoming video. It parallels the operations between GOPs. For example, as shown in Figure 4, the tasks are divided by the unit of GOP and a GOP is assigned to a processor. F_1,F_2,F_3,F_4 are processed by processor 1, and F_5,F_6,F_7,F_8 are handled by processor 2. There is almost no communication required between processors because there is almost no dependency between GOPs. In this case, the tasks are more easily implemented in general-purpose architectures. However, since the latency of the tasks is long, the delay is long. For example, we cannot start the parallel processing until we receive the starting frame of the GOP 2. While the long delay is probably allowed for off-line applications, it is one big drawback for the real-time applications (e.g., video-conferencing, video-phone).

Instead of such a long-delay coarse-grain parallel implementation, the proposed new scheme in which the GOPs are interlaced leads to a short-delay coarse-grain parallel implementation. As shown in Figure 5, we let F_1, F_3, F_5, F_7 (where F_1 means frame 1) to be GOP 1, and F_2, F_4, F_6, F_8 to be GOP 2. Encoding of GOP 1 will be assigned to processor 1 while encoding of GOP 2 will be assigned to processor 2. The delay of this parallelism is small because we can start the parallel encoding processing as soon as we receive F_2.

This coarse-grain parallel implementation has almost no communication required between processors, and so it is easily implemented in general-purpose microprocessor architectures. Additionally, this proposed GOP-interlaced coding scheme enables a short delay in the coarse-grain parallelism. In short, this scheme offers the ease of a linear-speedup of two in a dual-CPU system for real-time video compression (e.g., MPEG, H.263).

3. HIGHER QUALITY IN ERROR CONCEALMENT

Although this new coding scheme *does not* comply with current video coding standards, it would provide a better error-concealment scheme.

Aimed at masking the effect of missing blocks, error concealment plays a critical role in recovering the viewing quality of impaired video. As shown in Figure 6, error-resilient coding and error concealment techniques are introduced to reduce/recover the lost blocks in the damaged image frame using the available (received) video data without the need for retransmission of the missing data [2]. There are two basic steps in the receiver. (1) The concealment process must be supported by an appropriate transport format, which helps to identify the image pixel regions that correspond to lost video data. (2) Once the image regions (i.e., macroblocks, slices, etc.) to be concealed are identified, several methods have been proposed to conceal the lost blocks in the damaged image frame using the available (received) video data.

The specific details of the concealment procedure will depend upon the compression algorithm being used, and on the level of algorithmic complexity permissible within the decoder. A number of error concealment approaches have been developed to recover the damaged regions by adaptive interpolation in the spatial, temporal, and frequency domains [13]. For example, spatial error concealment algorithms first extract local geometric information (e.g., edges) from a neighborhood of surrounding undamaged pixels, and then reconstruct each lost pixel by spa-

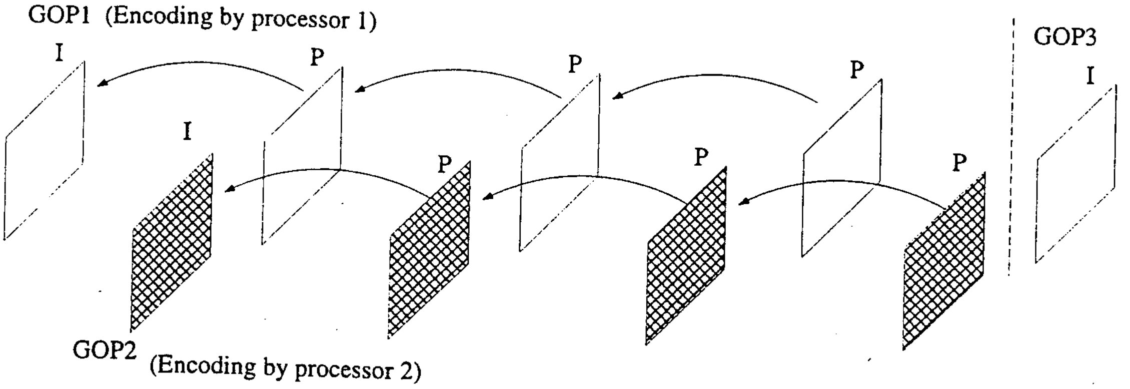
GOP1 (Encoding by processor 1)
I
P
P
P
GOP3
I
I
P
P
P
GOP2 (Encoding by processor 2)

Figure 5: The approach toward parallel processing the GOP-interlaced video compression scheme in coarse-grain. The task is divided by the unit of GOPs. The coding of a GOP is assigned to a processor. Since there is almost no dependency between GOPs, there is almost no communication necessary between processors. It is very easy to be implemented in a system with multiple general-purpose processors. The delay of this parallelism is small because we can start the parallel encoding processing as soon as we receive F_2.

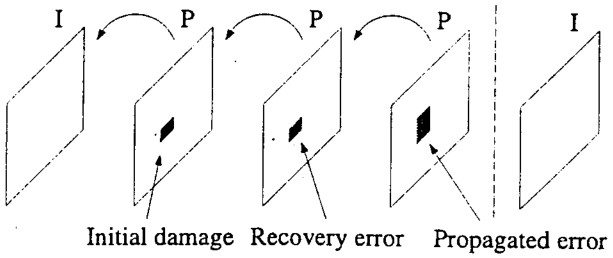

Figure 6: The goal of error concealment is to recover/reduce the lost using the available information. Although some damages are remained after error concealment, the overall damages are less than those damages without error concealment (cf. Figure 3).

tial interpolation [9, 10]. There are also some temporal error-concealment techniques. One is to simply replace the lost region from the corresponding regions of the previous frame. Another is to replace the lost region from the affine transformation of the previous frame [11]. Still another is to have temporal error concealment with motion compensation [1].

We focus primarily on improving the temporal error concealment with motion concealment. Our method does not identify the image regions that correspond to lost video data. We would like to assume the following:

1. *Physical motion field is piecewise continuous on temporary domain.*
   Therefore, we can approximate a true motion vector by either interpolation or extrapolation of the motion vectors from different frames.
2. *Physical motion vectors are used in the encoding process.*
   Many motion estimation algorithms are based on the search of minimal residue. However, it was shown that such motion estimation algorithms often miss the physical motion due to noise. Here, we assume that a true motion estimation algorithm is used in the encoding processing [4, 12].

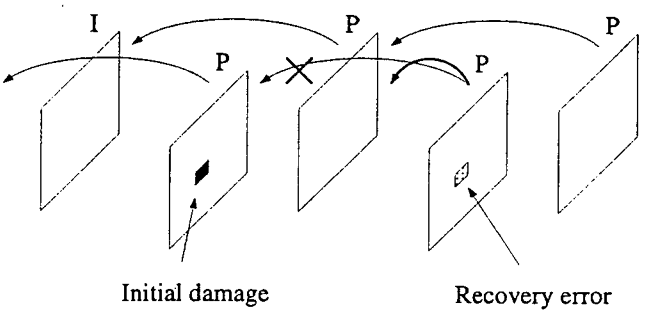

Figure 7: Error concealment in interlaced GOP. Original frame 4 should be reconstructed from motion compensation of frame 2. If there are some errors in receiving frame 2, then frame 4 will be reconstructed from frame 3 using half motion.

In original coding standards, a GOP contains F_i, F_{i+1}, $\cdots$, F_{i+n}. In an error-free condition, F_{i+j+1} depends on F_{i+j}. If there are some errors in receiving some blocks in F_{i+j}, then we will replenish the blocks from the corresponding position of F_{i+j-1}. Because some region in F_{i+j} is damaged, in order to reconstruct F_{i+j+1}, we make motion compensation from frame F_{i+j-1}. That is, if the original motion vector is $\vec{v}$ then the new motion vector should be $2\vec{v}$.

Figure 7 shows the new error-concealment scheme. In our error-concealment scheme, a GOP contains F_i, F_{i+2}, $\cdots$, F_{i+2n}. In an error-free condition, F_{i+2j+2} should refer to F_{i+2j}. If some image blocks in F_{i+2j} are lost, in order to reconstruct F_{i+2j+2}, we motion compensate from F_{i+2j+1} which belongs to another GOP and is not affected by error in F_{i+2j}. That is, the blocks (in F_{i+2j+2}) that refer to those damaged blocks (in F_{i+2j}) are now referring to F_{i+2j+1}. Assuming the original motion vector from F_{i+2j} to F_{i+2j+2} is $\vec{u}$, then the new motion vector from F_{i+2j+1} to F_{i+2j+2} should be $\vec{u}/2$.

Because interpolation is usually more accurate than extrapolation, we can see that our method could provide better motion compensation, and, therefore, better error-concealment result.

4. SIMULATION RESULTS

We first perform our simulations on the low bit-rate conditions suggested in MPEG-4 VM (Table 1). Because of frame skipping, the motion fields are usually large. The interpolation of motion vectors is more accurate than the extrapolation as shown in Table 2. Here, we isolate the error-concealment distortion from the compression distortion. Therefore, the SNR is compared with the ideal decoder reconstructed frames (not with the original uncompressed frames). We also perform our simulations on high bit-rate conditions, where no frame is skipped. As shown in Table 3, our method is 5dB SNR better on the average.

In summary, our method not only increases the short-delay coarse-grain parallelism but also increases the accuracy in error concealment.

5. REFERENCES

[1] S. Aign and K. Fazel, "Temporal & Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec," in *Proc. of IEEE Int'l Conf. on Communications*, vol. 3, pp. 1778–1783, 1995.

[2] M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error Concealment of Lost Motion Vectors with Overlapped Motion Compensation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 560–563, June 1997.

[3] Y.-K. Chen and S. Y. Kung, "Multimedia Signal Processors: An Architectural Platform with Algorithmic Compilation," *Journal of VLSI Signal Processing Systems*, vol. 20, no. 1/2, pp. 183–206, 1998.

[4] Y.-K. Chen and S. Y. Kung, "Rate Optimization by True Motion Estimation," in *Proc. of IEEE Workshop on Multimedia Signal Processing*, pp. 187–194, June 1997.

[5] Intel, "Intel MMX Technology–Developer's Guide." `http://developer.intel.com/drg/mmx/manuals/index.htm`, 1998.

[6] Intel, "Intel Video Phone." `http://www.intel.com/product/videophone`, 1998.

[7] R. B. Lee, "Subword Parallelism with MAX-2," *IEEE Micro*, vol. 16, no. 4, pp. 51–59, Aug. 1996.

[8] M. O'Connor, "Extending Instructions for Multimedia," *Electronic Engineering Times*, no. 874, p. 82, Nov. 1995.

[9] P. Salama, N. B. Shroff, E. J. Coyle, and E. J. Delp, "Error Concealment Techniques for Encoded Video Streams," in *IEEE Int'l Conf. on Image Processing*, vol. 1, pp. 9–12, 1995.

[10] H. Sun and W. Kwok, "Concealment of Damaged Block Transform Coded Images Using Projections onto Convex Sets," *IEEE Trans. on Image Processing*, vol. 4, no. 4, pp. 470–477, Apr. 1995.

[11] A. Tsai and J. Wilder, "MPEG Video Error Concealment for ATM Networks," in *Proceedings of the IEEE Int'l Conf. on Acoustics, Speech & Signal Processing*, vol. II, (Atlanta, GA), pp. 2002–2005, May 1996.

[12] A. Vetro, H. Sun, Y.-K. Chen, and S. Y. Kung, "True Motion Vectors for Robust Video Transmission," to appear in *Proc. of SPIE Visual Communications and Image Processing*, Jan. 1999.

[13] Y. Wang and Q.-F. Zhu, "Error Control and Concealment for Video Communication: A Review," *Proc. of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

ID	Sequences	Bit Rate (kbps)	Frame Rate (Hz)	Format
1	Container, Hall monitor, Mother & daughter	10	7.5	QCIF
2	Container, Hall monitor, Mother & daughter	24	10	QCIF
3	News	48	7.5	CIF
4	Coastguard, Foreman	48	10	QCIF
5	Coastguard, Foreman, News	112	15	CIF

Table 1: Our test conditions for low bit-rate conditions.

		PSNR (dB)		
Sequence	ID	Extrapolation	Interpo. (Ours)	Improvement
Container	1	35.77	46.34	10.57
Hall monitor	1	34.52	40.08	5.56
Mother&daughter	1	36.27	42.73	6.46
Container	2	35.97	46.94	10.97
Hall monitor	2	35.03	41.32	6.29
Mother&daughter	2	36.73	43.93	7.25
News	3	32.41	37.69	5.28
Coastguard	4	30.30	36.03	5.73
Foreman	4	26.81	33.20	6.39
Coastguard	5	29.82	36.31	6.49
Foreman	5	28.05	34.56	6.51
News	5	33.89	40.16	6.27

Table 2: Side-by-side comparison of the error concealment of the original GOP arrangement and the proposed interlaced GOP method. The block lost rate is 1%. There are 15 frames (1 I-frame and 14 P-frames) in a GOP. Note that the SNR is compared with the ideal decoder reconstructed frames (not with the original uncompressed frames).

	PSNR (dB)		
Sequence (30Hz CIF)	Extrapolation	Interpo. (Ours)	Improvement
Akiyo	41.05	53.09	12.04
Coastguard	31.49	40.16	9.67
Container	37.50	48.05	10.55
Foreman	30.32	39.03	8.71
Hall monitor	36.91	44.68	7.77
Mother & daughter	39.59	49.76	10.17
News	35.54	42.90	7.36
Stefan	26.10	30.94	4.84

Table 3: Side-by-side comparison of the error concealment of the original GOP arrangement and the proposed interlaced GOP method under high bit-rate conditions (30Hz CIF). The block lost rate is 1%. There are 15 frames (1 I-frame and 14 P-frames) in a GOP.