Journal of Information Hiding and Multimedia Signal Processing
Ubiquitous International

Volume **1**, Number **2**, April **2010**

Trick-Play Optimization for H.264 Video Decoding

Jian Lou
Motorola Inc.
6450 Sequence Drive, CA 92121, USA
lou@motorola.com

Shan Liu
StreamOcean Inc.
2882 Sand Hill Road Suite 150, Menlo Park, CA 94025, USA
shanliu@streamocean.com

Anthony Vetro
Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, MA 02139, USA
avetro@merl.com

Ming-Ting Sun
Department of Electrical Engineering
University of Washington
185 Stevens Way, Seattle, WA 98195, USA
sun@ee.washington.edu

Received June 2009; revised April 2010

Abstract. *Supporting digital Video Cassette Recording (VCR) trick-play functionalities (e.g. random access, fast-forward play, fast-reverse play) is desirable for compressed video streams. However, due to strong Inter-frame dependencies introduced by motion compensated prediction (MCP), the computational complexity and memory requirement is drastically increased. In this paper, we address the problem of digital VCR trick-play functionalities by investigating different prediction schemes. Specifically, we propose prediction schemes named G-Group and Binary Reference GOP Structure (BRGS), for the H.264/AVC video coding standard to achieve the trick-play functionalities while keeping low decoder complexity and memory requirement. We show that different prediction schemes represent different tradeoffs between the prediction distance (which affects the video quality) and the decoder complexity. We also address the problem of how to compare different prediction schemes by formulating it into an unconstrained optimization problem so that the minimal costs of different schemes can be compared. By comparing the solutions offered by G-Group and BRGS to the convex-hull of the operation points, we show that G-Group and BRGS are close to the global optimal solutions. The same approach can be used to evaluate other prediction schemes. Our proposed approaches are flexible and general, and can be easily adapted to evaluate and to achieve a good trade-off between compression performance and complexity saving for the trick play modes.*
Keywords: Video compression, H.264, Optimization, VCR, Trick-play

---

1. **Introduction.** Recently, most of the video contents for consumer applications are encoded using various video coding standards, since compressed digital videos are more preferable than the traditional video cassette in terms of storage and transmission efficiency. Due to the need for quick and user-friendly browsing of video content, it is

highly desirable for multimedia systems to support Video Cassette Recording (VCR) functionalities, such as random access, fast-forward play, and fast-reverse play. While it is straightforward to implement the trick-play modes for the video cassettes, it is not a trivial task for compressed video streams. Most of the video coding standards are based on the hybrid motion compensated prediction (MCP) framework, where the correlation between successive video frames is utilized to achieve higher compression ratios. However, MCP introduces Inter-frame dependencies and makes the video data in each frame not self contained, which is unfriendly for VCR functionalities. That is, in order to display one video frame in a trick-play mode, not only does the target frame need to be decoded, but also any reference frames that the target frame is predicted from need to be decoded as well. Therefore, the computational and memory complexity is dramatically increased compared to the normal decoding process.

Figure 1 presents the traditional "IBBP" GOP prediction structure, where all the P-frames reference the nearest previous I/P-frames and all the B-frames reference the nearest two I/P-frames. An example can be used to show how MCP affects the decoder complexity. In the figure, if the 8^{th} frame which is a B-frame is to be displayed during the process of trick-play, not only the 6^{th} and 9^{th} frames need to be decoded, but also the 0^{th} and 3^{rd} frames. Therefore, the decoder has to be much faster than the normal decoder in order to achieve the required trick-play speed. Moreover, if the frames with larger indices are to be accessed, the decoder complexity will be drastically increased.

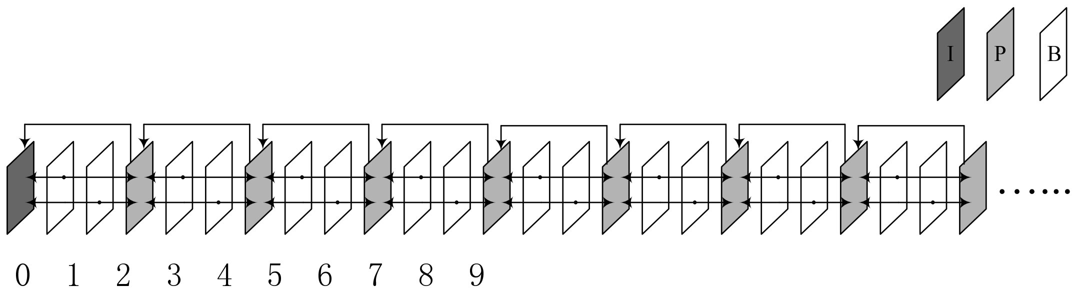


Figure 1. Traditional IBBP GOP structure

Several previous schemes on MPEG-2 or MPEG-4 have been proposed addressing the implementation of video trick-play modes. Several approaches utilizing transcoding between different frame types have been proposed [1]–[3]. However, extra complexity and higher storage cost are required to perform the transcoding. The approach proposed in [3] also causes drift due to the motion vector approximation. In [4], a scheme that stores both the forward-encoded and backward-encoded bitstreams in the server is proposed to reduce the reverse-play complexity while maintaining a low bandwidth. However, this doubles the storage requirement of the server and cannot be used for applications that require real-time or low-delay encoding. Center-biased motion vector distribution characteristics of video sequences are utilized in [5] for MPEG-2 video reverse-play. However, the bandwidth savings are highly dependent on the statistics of the coded sequence; thus, it is not always efficient for high motion sequences whose motion vectors are not center-biased. Furthermore, the scheme cannot be applied in the context of H.264/AVC due to the different semantics of the standards. As in MPEG-2, the non-coded MBs have zero value motion vectors, but in H.264 the skipped MBs indicate the MBs have zero value motion vector differences after prediction.

H.264/AVC permits the use of more pictures than just the previously decoded one for motion-compensation prediction and the multiple reference pictures could be organized in List 0 and List 1 used for prediction of P/B-frames. Another important feature of H.264/AVC is the decoupling of referencing order from display order. That is by using the explicit command indicated by the bits coded at the slice header, previously decoded pictures stored in the Decoded Picture Buffer (DPB) could be marked as a short term or long term reference picture, used as the references in different orders other than the default sequential one and could be removed from the DPB. The techniques are called reference picture list reordering commands and Memory Management Control Operations (MMCO), which allow the encoder to choose the ordering of pictures for referencing and display purposes with a high degree of flexibility [6]. With these techniques, encoders can flexibly choose any short-term and long-term references pictures as the reference to be used for prediction.

With the great flexibility of reference picture selection provided by H.264, different GOP prediction structures can be used for optimizing the implementation of trick-play modes. This does not change the standard compliant decoders and they could decode the bitstreams encoded with optimized GOP structures with much lower complexity when trick-play modes are enabled. Two frame-sized GOP structure is utilized in [7] to achieve complexity reduction, but it is only optimal for speedup factor of 2 and it also deteriorates the compression efficiency. Therefore, more generalized schemes and analysis should be provided. More importantly, the criteria to select the best GOP prediction structure and the associated parameters needs to be considered.

In terms of applicability, it should be noted that decoders can be designed to perform trick-play operations without knowing the specific prediction structure in advance. In this general case, the performance may be optimized for some typical prediction structures and could be expected to vary with alternative prediction structures. However, prediction structures that are specifically designed to optimize distortion and complexity of trick-play operation would have a clear benefit for those applications in which the encoder and decoder are part of the same platform and the encoding configuration is known to the decoder. Digital video cameras and digital video recorders are two important applications and we encounter such devices in a wide array of applications, most notably surveillance and consumer video storage, both of which have trick-play requirements. Another very useful application is medical video recording and playback in which smooth frame-level playback is critical.

In this paper, we first analyze the impact of VCR functionalities on H.264/AVC decoder complexity and memory requirement in Section 2. Then, we address the problem of VCR functionalities by proposing two efficient GOP prediction structures in Section 3. In Section 4, we show that different prediction schemes represent different tradeoffs between the prediction distance and the decoder complexity. In Section 5, we address the problem of how to compare different prediction schemes by formulating it into a constrained optimization problem, and converted it into an unconstrained optimization problem, so that the minimal costs of different schemes can be compared. In Section 6, by comparing the solutions offered by G-Group and BRGS to the convex-hull of the operation points, we show that G-Group and BRGS are close to the global optimal solution. The same approach can be used to evaluate other prediction schemes. In Section 7, we conclude this paper.

2. **Complexity and Memory Impact on Decoders of VCR Functionalities.** In this section, we provide an analysis of the computational complexity for H.264/AVC decoders when applying VCR functionalities. Here, we denote the index of the first I-frame as 0, and define N as the number of frames in one GOP and M as the Inter frame distance between every two successive I/P-frames, which we assume fixed throughout the

whole GOP. Also, we will use R_P, R_{BL0}, and R_{BL1} as the number of reference frames used for P-frames, B-frame List 0, and B-frame List 1, respectively. The bitstreams are encoded with a conventional prediction structure where every P-frame is predicted from the nearest forward I/P-frame and every B-frame uses the nearest I/P-frame as the references. Forward reference frames are in List 0 and backward reference frames are in List 1.

2.1. **Random access.** When random access to a specific frame is required, the frame to be displayed and any frames that it depends on need to be decoded. The complexity to access the j-th frame (if j is larger than N, we can use mod(j, N) instead) in one GOP is denoted as $C_{RA}(j)$ which indicates the number of frames to be decoded.

$$C_{RA}(j) = \begin{cases} 1 & \text{I-frame} \\ j/M + 1 & \text{P-frame} \\ \min((\lceil j/M \rceil + R_{BL1}), N/M + 1) & \text{B-frame} \end{cases} \tag{1}$$

Within one GOP, the decoding complexity for a P-frame is determined by its index j, since larger j means more previously encoded P-frames need to be decoded due to MCP reference dependencies.

2.2. **Fast-forward play.** In fast-forward play, we can jump to the next I-frame as the starting point. We denote the speedup factor as s. After s/g GOPs, the frame to be displayed will again be an I-frame, where $g = \gcd(s, N)$ stands for the greatest common divisor of s and N. Therefore, our analysis is based on $N \times s/g$ frames starting from an I-frame. Moreover, we assume N is larger than s, which is usually the case in practice.

In general, fast-forward play can be regarded as accessing the frames with indices of 0, s, $2s$, $(N/g\text{-}1) \times s$. In these frames, only the first frame is an I-frame and there are $N \times s/(g \times h)$-1 P-frames and the remaining are B-frames, where $h = \mathrm{lcm}(s, M)$ stands for the least common multiple of s and M. If all the decoded frames for displaying the current frame are discarded although some of the decoded frames could be reused for decoding future frames to be displayed, the average number of frames that need to be decoded for fast-forward play is

$$\overline{C_{FF}} = \frac{1 + \sum_i (i/M + 1) + \sum_j \min((\lceil j/M \rceil + R_{BL1}), N/M + 1)}{N/g} \tag{2}$$

where i and j stand for P-frame and B-frame indices respectively. The three parts in the numerator indicate the total decoding complexity for I-frame, P-frames, and B-frames, respectively.

If we store those decoded frames which could be used for future display, the decoding complexity and bandwidth requirement will be reduced, but the buffer memory will be increased. The decoding complexity defined in (2) will be reduced by the number of necessary frames buffered in the memory, and the memory size will be increased by the maximum number of frames buffered for future use in addition to R_{BL0}+R_{BL1} in the unit of frame size.

2.3. **Fast-reverse play.** In the fast-reverse play mode, the problem is similar to that for fast-forward play assuming the ending point is an I-frame. If the decoder discards the decoded frames, the average complexity for fast-reverse play is the same as that for fast-forward. The reason is that both fast-forward and fast-reverse play modes follow the same periodicity which indicates that all the frames to be displayed are the same for the two modes. We could choose to buffer those decoded frames needed for future reverse display use. Those frames will not only serve as the reference frames for future frames, but also need to be displayed. Therefore, buffering these frames will save the computational complexity. However, the maximum decoding speed and maximum buffer size will be greater for fast-reverse play compared to fast-forward play due to the forward encoding nature in video compression.

3. **Complexity and Memory Efficient GOP Structures.** A straightforward scheme to reduce the computational complexity and memory requirement is to assign the I-frame as the only reference for all the P-frames in the same GOP and every B-frame uses the nearest 2 I/P-frames as the references. We refer to this scheme as "All P Ref I". The dependencies between P-frames have been eliminated in this scheme, so the complexity for accessing one P or B-frame is minimal. The obvious drawback of this method is that the performance of MCP will be greatly degraded when the distance between I/P-frames becomes too long. There are many possible prediction schemes which offer better tradeoffs between the prediction distance and the decoder complexity. In the following, we propose two example schemes.

3.1. **G-Group GOP structure.** In order to reduce the Inter-frame dependencies without sacrificing too much coding efficiency, we introduce a scheme called G-Group GOP Structure where several successive P-frames are grouped together which only use the last P-frame in the previous group or the I-frame as the reference. Here, G stands for the number of P-frames in one group. The index of the reference frame, RI_P, for the current P-frame with index j is

$$RI_P(j) = \lfloor (j/M - 1)/G \rfloor MG \tag{3}$$

In addition, every B-frame still uses the nearest 2 I/P-frames as the references in order to keep the low Inter-frame dependencies. Figure 2 provides an example with G=4.

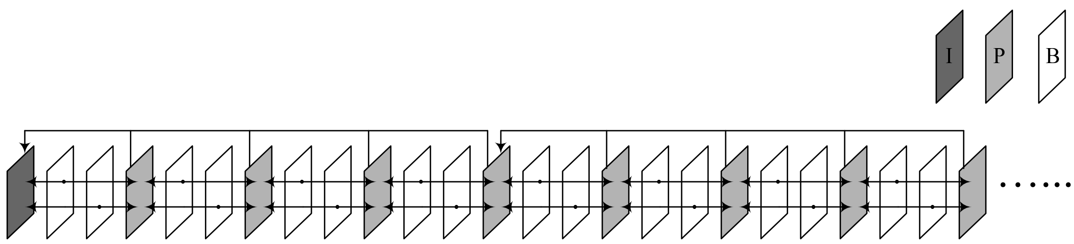


FIGURE 2. G-Group GOP structure (M=3, G=4)

The Longest Forward Prediction Distance ($LFPD$), the Average Forward Prediction Distance ($AFPD$), the Random Access Worst Complexity ($RAWC$), and the Random Access Average Complexity ($RAAC$) for G-Group are

$$LFPD_{G-Group} = MG \tag{4}$$

$$AFPD_{G-Group} = \frac{M+G}{2} \tag{5}$$

$$RAWC_{G-Group} = \lceil N/G \rceil + 1 \tag{6}$$

$$RAAC_{G-Group} = \frac{1 + G \sum_{i=1}^{\frac{N-1}{MG}} (i+1) + (M-1) \sum_{j=1}^{\frac{N-1}{MG}} ((j+3)G - 1)}{N} \tag{7}$$

LFPD and *AFPD* are indicators of coding efficiency, while *RAWC* and *RAAC* are indicators of decoder complexity. According to (4)-(7), when N and M are fixed, G can serve as the parameter to control the coding efficiency and complexity. Generally, with larger G, the average and worst prediction distance will be larger which leads to worse coding efficiency, but lower the trick-play complexity.

3.2. **Binary reference GOP structure.** Note that the G-Group scheme linearly increases the prediction distance and reduces the complexity. It is possible to use a logarithmic scheme which we refer to as Binary Reference GOP Structure (BRGS). The scheme uses an index derived from the binary code of the number, j/M, where j is the frame index of a P-frame. Let L represent the number of bits for the binary code. The index of the reference frame for the P-frame with index j is

$$RI_p(j) = \text{mod}(j/M, 2^L) - 2^k + \lfloor (j/M - 1)/2^L \rfloor 2^L \tag{8}$$

where k stands for the position of last 1 in the binary code of mod(j/M, 2^L). Figure 3 provides an example of BRGS with L=3.

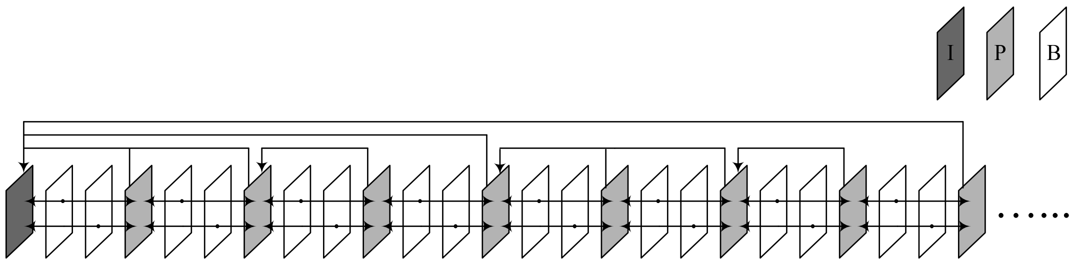


FIGURE 3. Binary reference GOP structure (M=3, L=3)

It can be shown that:

$$LFPD_{BRGS} = M2^L \tag{9}$$

$$AFPD_{BRGS} = \frac{M + L + 1}{2} \tag{10}$$

$$RAWC_{BRGS} = \frac{(N-1)2^L}{M} + L \tag{11}$$

$$RAAC_{BRGS} = \frac{1 + \sum_{i=0}^{\frac{N-1}{M2^L}-1} (2^L + L2^{L-1} + 1 + (M-1)(3 \cdot 2^L + L2^{L-1}) + iM2^L)}{N} \tag{12}$$

The parameter L can be utilized to control the tradeoff between the prediction distance (and thus coding efficiency) and the decoder complexity.

The average number of frames that need to be decoded for fast-forward/-reverse play can be derived by replacing the three components in (2) with the counterparts in the two schemes, i.e., with (6) and (11). The maximum buffer size required for fast-reverse play will be reduced, due to the fewer frames buffered to be used for future display.

4. **Coding Efficiency and Decoder Complexity of G-Group and BRGS.** We conduct experiments to evaluate the performance and gain insights of the G-Group and BRGS prediction schemes. In our experiments, we set N to 30 frames and M to 3. The number of reference frames in every list is set to 1. The necessary frames are stored in the memory for future display. We choose two G-Group schemes with G=2 and G=4, and BRGS with L=3 as the examples which could be applied in practical applications with the typical GOP setting (N=30 and M=3).

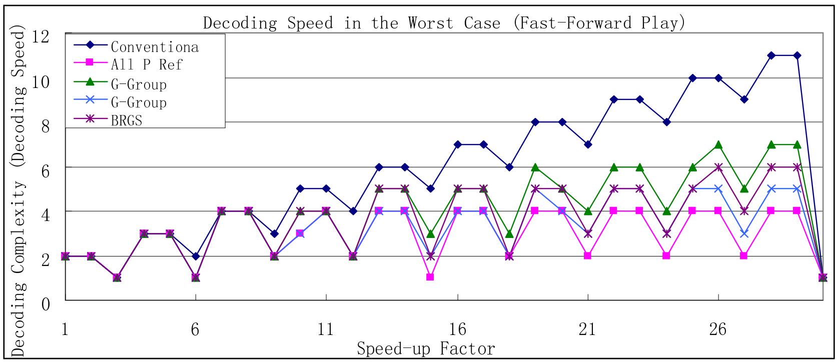
Table 1. Average number of frames to be decoded for random access

	Conventional	All P Ref I	G-Group(G=2)	G-Group(G=4)	BRGS(L=3)
LFPD	3	27	6	12	24
AFPD	1.97	5.69	2.38	3.21	3.21
RAWC	12	4	8	6	6
RAAC	6.83	3.23	4.83	3.83	3.83

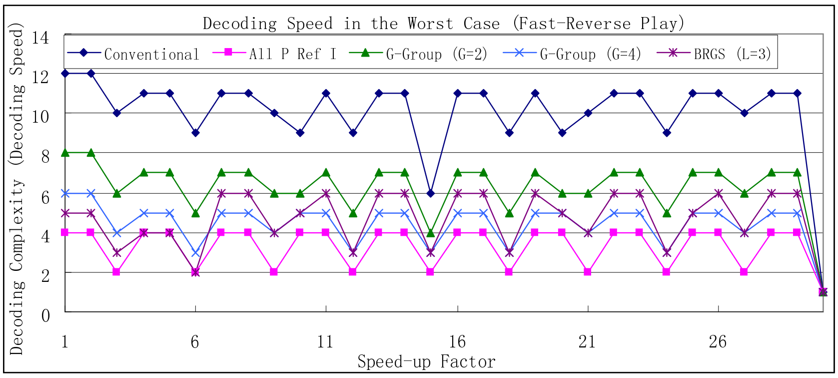
Table 1 compares the average number of frames to be decoded for random access which indicates that both G-Group and BRGS can achieve similar complexity as the "All P Ref I" scheme but with shorter prediction distances. Figure 4 shows the decoding speed needed for the fast-forward and fast-reverse play modes in the worst case. Here, "decoding speed" is defined as the number of frames to be decoded in the interval between displaying 2 succeeded frames. Figure 5 shows the decoding speed on average. These figures indicate that, in the worst case, the "Conventional" structure requires the decoder to be 12 times faster than the normal decoding speed without trick mode while the "All P Ref I" structure only needs to be 4 times faster. The decoding speeds needed for the G-Group and BRGS are between those of the two reference structures. The complexity saving for G-Group with G=2, G-Group with G=4, and BRGS with L=3 compared to the conventional prediction scheme for the worst case and the average case can be up to 50% and 44.5%, 66.7% and 66.7%, and 66.7% and 77.8%, respectively. It should also be noted that certain applications may target only a limited subset of speed-up factors. These plots also show that certain subsets of speed-up factors, e.g., multiples of three (3x, 6x, 9x, etc.), can be supported with lower decoding complexity.

The maximum buffer sizes needed for the different schemes are presented in Figure 6. For fast-forward play, there is no difference between different schemes, however the memory buffer for the fast-reverse play mode which is of more importance [1]-[5] can be reduced to 55.6%, 33.3%, and 33.3% for G-Group with G=2, G-Group with G=4, and BRGS with L=3, respectively.

The coding efficiency of different schemes is verified using the H.264/AVC reference software, JM12.3 [8]. We choose four 1280×720 format sequences, "City", "Cyclists", "Horses", and "Night". High complexity R-D optimization is enabled. Table 2 lists the PSNR loss or equivalent bitrate increase [9] compared with the "Conventional" approach when applying different GOP structures. Even though "All P ref I" structure gives the best trick-play complexity, it introduces 17.27% bitrate increase. The average bitrate increases from G-Group and BRGS schemes are between 3.97% and 7.6%. As can be seen from the sample R-D curves for "Cyclists" in Figure 7, the G-Group and BRGS are much



(a) Fast-forward play



(b) Fast-reverse play

FIGURE 4. Decoding speed in the worst case

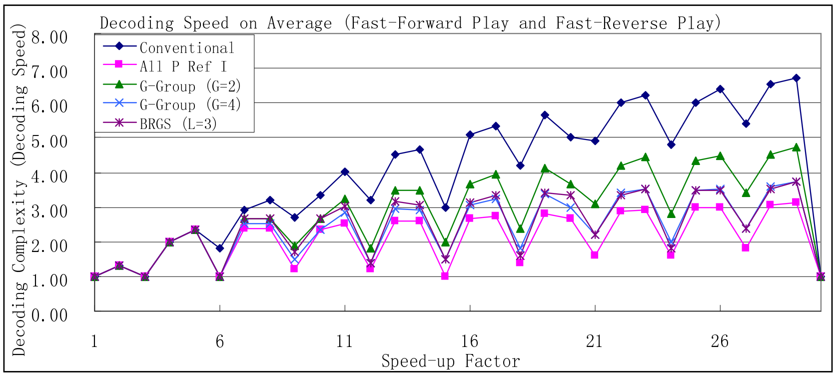
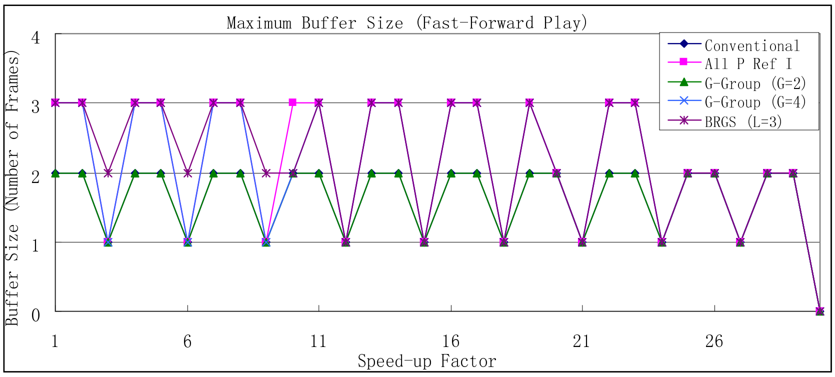


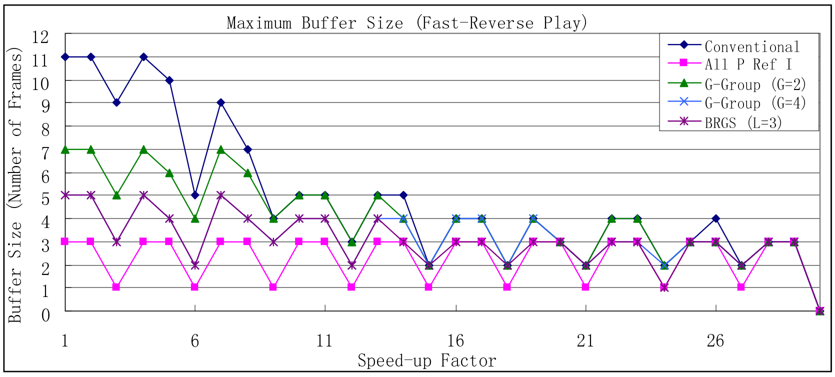
FIGURE 5. Decoding speed on average

better than the "All P ref I" structure. Also, when videos are encoded at higher quality, e.g., 38 dB, the performance loss from the proposed schemes becomes negligible.

These experimental results indicate that good tradeoffs between coding efficiency, complexity, and memory saving for the trick-play modes can be made with different GOP structures. With appropriate parameters, G-Group and BRGS can greatly reduce the computational complexity and memory requirement for VCR functionalities in H.264/AVC without much loss of the coding efficiency.



(a) Fast-forward play



(b) Fast-reverse play

FIGURE 6. Max buffer size

TABLE 2. Compression performance for different GOP structures compared with the "Conventional" scheme

Sequence		All P Ref I	G-Group(G=2)	G-Group(G=4)	BRGS(L=3)
City	ΔPSNR	-0.68dB	-0.15dB	-0.27dB	-0.31dB
	ΔBitrate	23.21%	5.09%	9.21%	10.43%
Cyclists	ΔPSNR	-0.46dB	-0.09dB	-0.19dB	-0.19dB
	ΔBitrate	17.43%	3.44%	7.22%	7.37%
Horses	ΔPSNR	-0.41dB	-0.10dB	-0.20dB	-0.20dB
	ΔBitrate	16.24%	3.87%	8.01%	8.07%
Night	ΔPSNR	-0.39dB	-0.11dB	-0.14dB	-0.15dB
	ΔBitrate	12.18%	3.48%	4.17%	4.55%
Average	ΔPSNR	-0.49dB	-0.11dB	-0.19dB	-0.21dB
	ΔBitrate	17.27%	3.97%	7.15%	7.6%

5. **Evaluation of Prediction Schemes.** As shown in Section 4, different prediction schemes represent different tradeoffs between the prediction distance and decoder complexity. The prediction distance is related to the coding efficiency and thus the quality of the compressed video. In general, the longer the prediction distance, the worse the coding efficiency. Furthermore, in different schemes there is a different parameter (e.g., L in BRGS and G in G-Group) that controls the tradeoff between the prediction distance and the decoder complexity. Different prediction schemes with different values of the parameters will give different coding efficiency and decoder complexity. A natural question

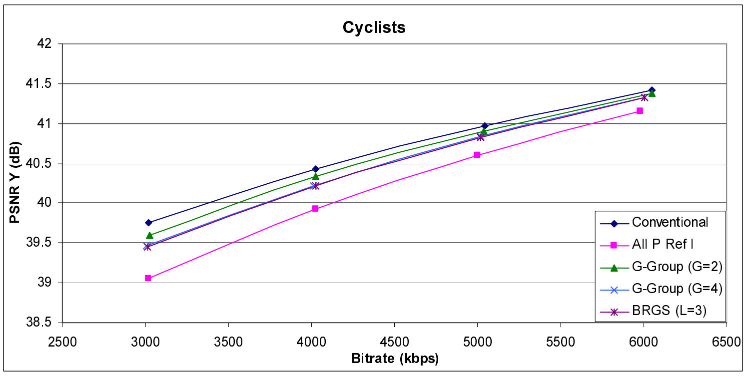


FIGURE 7. R-D performance for different GOP structures

is how to evaluate these different schemes, i.e., given an application, how do we choose the prediction scheme and the value of the parameter to give the best results. This is not a trivial question, since the tradeoff behaviors are different for different prediction schemes as shown above.

We formulate this problem into a constrained optimization problem. Different prediction schemes basically try to achieve the optimal prediction distance (which affects the video quality) under a complexity (frames to be decoded, decoder buffer size required, etc.) constraint. We aim to solve for the optimal prediction structure with respect to coding efficiency (rate and distortion) as well as complexity (decoding speed and memory) incurred by the trick-play operations. Specific trick-play operations include random access, and fast-forward/-reverse play within a specified range of speedup factors. In the following, we will use random access as an example for discussions.

Let $\theta \in \Theta$ denote a particular combination of coding parameters. The constrained optimization problem can be stated as: given a coding rate, maximize the video quality under a decoder complexity constraint. It can be formally stated as follows:

$$D^* = \min_{\theta \in \Theta} \{D(\theta)\} \text{ subject to } C(\theta) < C_{max} \tag{13}$$

where C denotes the complexity (could be the average complexity or the worst-case complexity). A more general formulation of the problem would explicitly impose a rate constraint in (13) and consider tradeoffs in the space of distortion, complexity and rate. Our current work assumes that the rate is fixed and focuses on tradeoffs between distortion and complexity.

The constrained in (13) can be converted to an unconstrained optimization problem by using a Lagrange multiplier which can be specified by a user depending on the requirements of a specific application to reflect the relative importance of the prediction distance and the complexity. The cost function J is defined as:

$$J(\theta) = C(\theta) + \lambda \times D(\theta) \tag{14}$$

The parameter λ is chosen to reflect the relative importance ratio between distortion and complexity. In the classic rate-distortion optimization framework for video coding, a reasonable value of λ was determined to be proportional to the square of the quantization parameter [10]. Similarly, we might expect a reasonable value of λ for the constrained optimization problem in (14) to be dependent on the GOP parameters. This requires further study since there are many dependencies to consider, so the selection of λ including its dependence on the GOP parameters and chosen functions for complexity and distortion is considered outside the scope of the current work. In practice, λ could be iteratively adjusted if more emphasis on complexity or distortion is desired.

Different distortion models related to the prediction schemes could be used. If we use the prediction distance (could be the average prediction distance or the worst-case prediction distance), PD, as a distortion model, Equation (14) becomes:

$$J(\theta) = C(\theta) + \lambda \times PD(\theta) \tag{15}$$

Given the complexity and the PD as functions of the parameters of different prediction schemes (i.e., L in BRGS and G in G-Group) as in Equation (4)-(7), We can derive close form solutions for the optimal parameter and the optimal cost of a prediction scheme as a function of λ, GOP size (N), Inter frame distance between every two successive I/P-frames (M), and VCR trick mode parameters (e.g., speedup factor in the fast-forward/-reverse play). The minimal costs of different schemes can then be compared to determine which is a better scheme for a certain application.

Use the G-Group prediction scheme as an example, if we choose to use $RAAC$ and $AFPD$ to represent the complexity and the PD in Equation (15), then (15) can be written as:

$$J = \frac{M+G}{2} + \lambda \times \frac{7MNG + N^2 - 2MN - 4NG - 5MG + 2M + 4G - 1}{2MNG} \tag{16}$$

In order to find out the optimal G which gives the minimum cost, we calculate the partial derivative of (16) with respect to G, and set it to 0 to obtain:

$$G^* = \sqrt{\frac{\lambda \times (N^2 - 2MN + 2M - 1)}{MN}} \tag{17}$$

where G^* is the optimal parameter for the G-Group under a given λ, N, and M. With G^*, the optimal cost can be easily calculated for comparisons with those from other prediction schemes. Following a similar process, we could derive the optimal G and J if other forms for complexity and prediction distance are used. Furthermore, the above process can be applied on other regular prediction structures with appropriate control parameters, e.g., BRGS with L.

6. **Comparisons to the Global Optimal Solutions.** Although the G-Group and BRGS prediction schemes provide good solutions, it is not clear how do they compare to the global optimal solution. Since there could be many possible prediction schemes, in general, it is desirable to have an approach which we can use to evaluate a particular prediction scheme to see how close it is to the global optimal solution.

To address this problem, we plot the operation points by varying the prediction structures. An example prediction structure is shown in Figure 8. Given a prediction structure, and a complexity and distortion measure, we can calculate the corresponding complexity and the distortion as an operation point. We plot all the operation points corresponding to all the possible prediction structures. Each regular prediction scheme with a specific parameter value will also be represented as an operation point in the plot. From the relative positions of the operation points to the convex-hull of the operation points, we can know how close the particular scheme is to the optimal solution (solid line in Figure 9). As an example, using $RAAC$ as the complexity measure and $AFPD$ as the distortion measure, we plot the operation points of all possible predictions structures in Figure 9.

In the figure, we choose the GOP size to be 10 and there is no B-frame. We also connect the operation points of G-Group and BRGS with different parameters. By comparing the operation points of G-Group and BRGS to the convex-hull, we show that G-Group and BRGS are close to the global optimal solutions. It can be noted from Figure 9 that G-Group is more preferable than BRGS in terms of granularity for the number of choices on the coding efficiency and decoder complexity tradeoff, as the number of G-Group structures can be the number of pictures in one GOP minus 1, but the number of BRGS structures is only Log_2(number of pictures in one GOP). The same approach can be used to evaluate other prediction schemes.

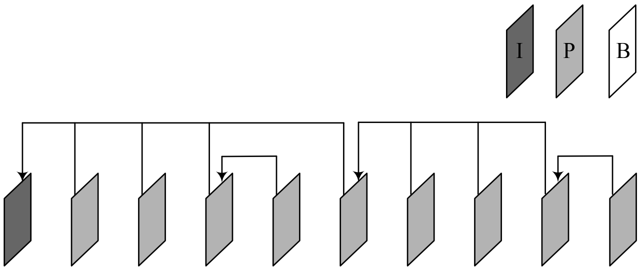


Figure 8. GOP structure with $RAAC$=2.5 and $AFPD$=2.11

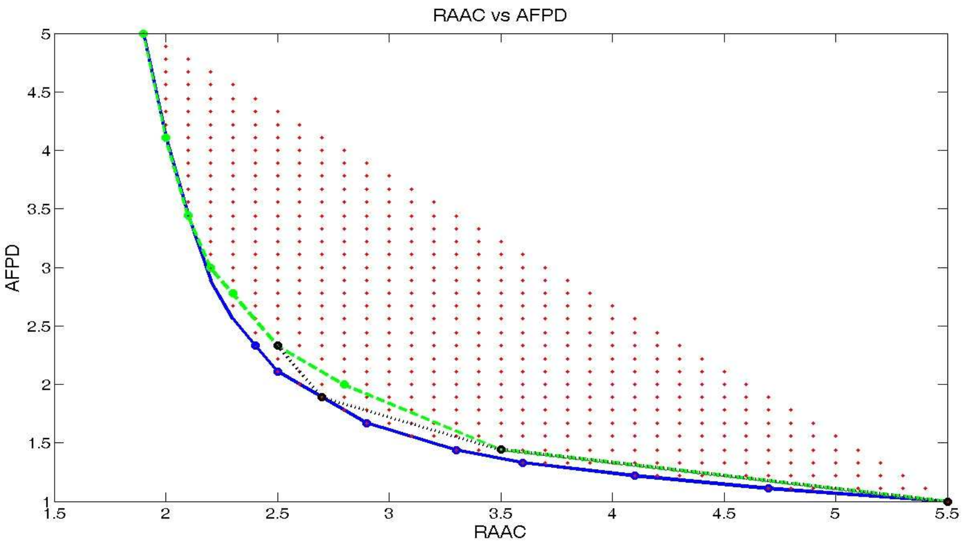


Figure 9. Operation points for different GOP prediction structures (dashed line: G-Group; dotted line: BRGS; solid line: convex hull)

7. **Conclusions.** Due to the Inter-frame dependencies introduced by modern video coding standards, the computational complexity and memory requirement for decoders can be drastically increased when VCR functionalities need to be supported. In this paper, we first analyze the impact of trick-play modes on the decoder complexity and buffer size. Then, we propose two drift-free schemes called G-Group GOP Structure and Binary Reference GOP Structure for H.264/AVC which can reduce the computational complexity and buffer size when applying VCR functionalities relative to the conventional GOP structures. These schemes do not change the standard compliant decoders and result in only 4.0%–7.6% bitrate increase while requiring much less complexity and smaller buffer size for trick-play modes. Users can choose appropriate parameters to control the GOP

structure to satisfy their constraints on coding efficiency, decoder complexity and memory requirement. We also propose an approach to choose the optimal parameters for different prediction schemes by jointly optimizing the coding efficiency and the decoder complexity under the trick-plays. Different prediction schemes can be compared with the optimal cost and be compared to the global optimal solution.

REFERENCES

[1] M. S. Chen and D. D. Kandlur, Downloading and stream conversion: supporting interactive playout of videos in a client station, *Proc. of IEEE Int'l Conf. on Multimedia Computing and Systems*, pp. 73–80, May 1995.

[2] S. J. Wee, Reversing motion vector fields, *Proc. of IEEE Int'l Conf. on Image Processing*, vol. 3, pp. 209–212, Oct. 1998.

[3] S. J. Wee and B. Vasudev, Compressed-domain reverse play of MPEG video streams, *Proc. of SPIE Conf. on Multimedia Systems and Applications*, pp. 237–248, Nov. 1998.

[4] C. W. Lin, J. Zhou, J. Youn, and M. T. Sun, MPEG video streaming with VCR functionality, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 415–425, Mar. 2001.

[5] C. H. Fu, Y. L. Chan, T. P. Ip, and W. C. Siu, New architecture for MPEG video streaming system with backward playback support, *IEEE Trans. Image Processing*, vol. 16, no. 8, pp. 2169–2183, Sept. 2007.

[6] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, Overview of the H.264/AVC video coding standard, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[7] J. H. Jeong, Y. J. Lee, H. Y. Kim, and M. J. Kim, Trick play method for HD H.264 set-top box, *IEEE Trans. Consumer Electronics*, vol. 54, no. 2, pp. 817–822, May 2008.

[8] Available at: http://iphome.hhi.de/suehring/tml/download/old_jm/jm12.3.zip

[9] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, *VCEG document, VCEG-M33*, Mar. 2001.

[10] G. J. Sullivan and T Wiegand, Rate-distortion optimization for video compression, *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.