Enhanced Mode-Adaptive Fine Granularity Scalability

Wen-Hsiao Peng,[1] Yen-Kuang Chen

Intel Corporation, Santa Clara, California 95052

ABSTRACT: In this article, we present a scalable video compression algorithm to deliver higher compression efficiency with limited drifting error. MPEG-4 Fine Granularity Scalability (FGS) compresses the video into a base layer and an enhancement layer. Currently, because the enhancement layer is predicted from the poor-quality base layer, the compression efficiency is low. To improve the compression efficiency, we construct enhancement-layer predictors from (1) macroblocks of current reconstructed base-layer frame, (2) macroblocks of previously reconstructed enhancement-layer frame, and (3) the average of previous two. On the other hand, the unpredictable receiving manner of enhancement layer could cause predictor mismatch error. The predictor mismatch error further results in drifting error. To minimize the drifting error, we create an adaptive mode-selection algorithm, in the encoder, which first smartly estimates possible drifting error of the decoder side and then uses the best macroblock modes wisely. In this article, we show that predictors constructed jointly from the base-layer frame and the enhancement-layer frame can reduce the drifting error. And, predictors constructed from the base-layer frame can stop the drifting error. As compared to other advance FGS schemes, our algorithm shows 0.3–0.5 dB PSNR improvement with a less complex structure. Although compared to MPEG-4 FGS, more than 1–1.5 dB quality improvement can be gained. © 2004 Wiley Periodicals, Inc. Int J Imaging Syst Technol, 13, 308–321, 2003; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ima.10090

Key words: fine granularity scalability; layered video coding; streaming video

I. INTRODUCTION

In MPEG-4 streaming video profile (N3315, 2000), a fine-granularity-scalability (FGS) codec is defined. Scalable video compression codec offers appropriate visual quality with truncated bit-stream. Such an embedded property helps us in channel bandwidth adaptation, graceful degradation from packet error, the system level compression efficiency, and so on (Wu et al., 2001a). To offer scalability at very fine granularity, MPEG-4 FGS utilizes layered and bit-plane coding. Specifically, MPEG-4 FGS compresses the video into a base layer and an enhancement layer (N3315, 2000; Li, 1998, 2001). The base layer offers a minimum guaranteed visual quality. Then, the enhancement layer refines the quality over that offered by the base layer.

While offering good scalability at fine granularity, the compression efficiency of MPEG-4 FGS is often much lower than that of a nonscalable codec. Currently, in MPEG-4 FGS, the enhancement layer is predicted from the base layer. In most applications, the base layer is encoded at very low bit rate (Wu et al., 2001a) and the reconstructed base layer is often with poor visual quality. Because the poor-quality predictor cannot remove the redundancy effectively, the coding efficiency is inferior.

Using the enhancement layer for better prediction can improve the coding efficiency (He et al., 2002; Huang et al., 2002; van der Schaar et al., 2001; Peng et al., 2001; Rose et al., 1998, 2001; Wu et al., 2001a, 2001b). Wu et al. (2001a, 2001b) construct better macroblock predictor from a previous enhancement-layer frame. In addition to a previous enhancement-layer frame, Huang et al. (2002) further exploit the previous base-layer frame while producing a frame-based predictor. In our previous work (Peng et al., 2001), we offer three macroblock prediction modes: (1) Type B: from the current reconstructed base-layer frame, (2) Type E: from the previously reconstructed enhancement-layer frame, and (3) Type BE: from the average of previous two. Although differing in constructing enhancement-layer predictor, all the advance FGS schemes try to find a better predictor by enhancement-layer frame to improve coding efficiency.

Although the coding efficiency can be improved by using enhancement-layer frame, drifting error could occur at low bit rate. This is because the enhancement layer is not guaranteed being received in an expected manner. The predictor mismatch between the decoder and the encoder would produce the drifting error. Wu et al. (2001a, 2001b) stop the drifting error by enabling a predictor that artificially creates mismatch error during encoding. The predictor is enabled by a mode decision mechanism (Wu et al., 2001c). In Huang's work (2002), they apply a predictive leak factor between 0 and 1 to decay the drifting error. Their method is to multiply the previous enhancement-layer frame with a fractional factor, alpha. In this article, we adaptively use Type B and Type BE predictors to offer two schemes, reset and fading mechanism, to stop/reduce the drifting error. During our predictor selection, we estimate the possible drifting error by introducing a dummy reference frame in the encoder.

While preserving the scalability of MPEG-4 FGS, our goal in this article is to offer better coding efficiency at all bit rates. Figure 1 characterizes our goal in terms of rate-distortion and compares our goal with different approaches. Besides a theoretical framework of improving the enhancement-layer prediction and reducing the enhancement-layer drifting error, our main contributories in this work include the following:

Correspondence to: Wen-Hsiao Peng; e-mail: pawn@royals.ee.nctu.edu.tw

[1] Wen-Hsiao Peng is currently a Ph.D. candidate at Institute of Electronics Engineering, Nation Chiao Tung University, Hsin-Chu, Taiwan. The majority of the work is completed when he was with Microprocessor Research Labs, Intel Corporation

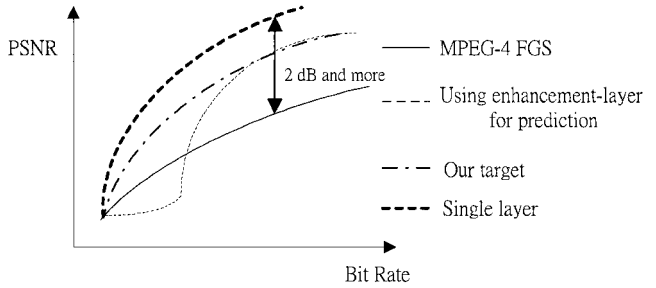

Figure 1. Rate-distortion performance of MPEG-4 FGS, single layer codec, codec using enhancement-layer for prediction and our target.

1. We adaptively construct macroblock predictors from previous enhancement-layer frame, current base-layer frame, and the combination of both. We offer two new Type E and Type BE macroblock predictors in additional to the original Type B macroblock predictor.
2. Although two of the three modes can improve the coding efficiency, two of the three modes can reduce the drifting error. In particular, we adaptively use Type E and Type BE predictors to increase the predictor efficiency. And, we adaptively, at the macroblock level, enable the reset mechanism with the Type B predictor and the fading mechanism with the Type BE predictor.
3. We create a dummy reference frame in the encoder to "accurately" model the drifting error for each macroblock, and thus, our best predictor selection is according to the "improvement gain" and the "drifting loss."

Our proposed algorithm has the following unique features:

- Different from Huang et al. (2002), our Type BE predictor combines the previous enhancement-layer frame with the more correlated current base-layer frame, instead of the less correlated previous based-layer frame.
- As compared to Wu et al. (2001a, 2001b, 2001c), our reset mechanism is simply from Type B predictor; we do not need to artificially create the mismatch error.
- Our predictor selection algorithm simultaneously considers the performance at high bit rate and low bit rate, whereas Wu et al. (2001c) only consider the low bit-rate performance.
- Our fading mechanism only needs a simple fractional factor 0.5; we do not need a complicated fractional factor as in Huang et al. (2002).
- The decoder for our proposed scheme is less complex than others.

We call our proposed enhanced mode-adaptive fine-granularity-scalability codec, EMFGS.

Experiment results show that our approach can efficiently reduce the drifting error while still maintaining more than 1.5 dB PSNR gain at high bit rate over the current MPEG-4 FGS. Further, as compared to other advance FGS algorithms (Huang et al., 2002; Wu et al., 2001c), we can reach similar or better performance by 0.3–0.5 dB in most cases.

The rest of this article is organized as follows. Section II formulates the problem. Section III describes our proposed EMFGS scheme, including the prediction modes and the mode selection algorithm. Section IV analyzes the prediction mode distributions in different conditions. Section V depicts our encoder and decoder structure. Section VI further compares our approach with other advanced FGS schemes. Section VII demonstrates the rate-distortion performance of our proposed codec. Finally, Section VIII summarizes our work.

II. PROBLEM STATEMENT

The problem that we would like to solve in this article is how to construct a better enhancement-layer predictor that improves coding efficiency while minimizing the degradation from drifting error.

When our EMFGS, MPEG-4 FGS, and other advanced FGS schemes compress the video into a base-layer and an enhancement-layer, there are three assumptions:

1. Base layer is guaranteed to be received without error.
2. Base layer is of low bit rate and low quality. Thus, the enhancement-layer prediction residue is large.
3. Enhancement layer is not received in an expected manner by encoder/server. Thus, there could be predictor mismatch if we use enhancement layer for enhancement-layer prediction.

Based on these assumptions, this section describes (a) our formulation of minimizing the enhancement-layer prediction residue, (b) the problem when the decoder receives less enhancement layer than expected, (c) our formulation of the enhancement-layer predictor mismatch errors, and (d) our target of constraining the errors. Table I lists our symbol definition throughout this article.

A. Improved Enhancement-Layer Predictor to Minimize the Enhancement-Layer Prediction Residue.

Although current MPEG-4 FGS predicts the enhancement-layer from the base-layer, we can exploit the available reconstructed frames at time t for better enhancement-layer predictor. Currently, the enhancement-layer predictor $P_E(t)$ is like the following function:

$$P_E(t) = f(I_B(t)). \tag{1}$$

Table I. Symbol definitions.

Symbols	Meaning
$I_o(t)$	The original source frame
$\tilde{I}_o(t)$	The reconstructed frame at decoder (with error)
$I_B(t)$	The reconstructed frame of base layer
$I_E(t)$	The reconstructed frame of enhancement layer at encoder
$\tilde{I}_E(t)$	The reconstructed frame of enhancement layer at decoder (with error)
$P_E(t)$	The enhancement-layer predictor
$\tilde{P}_E(t)$	The reconstructed predictor of enhancement layer at decoder (with error)
$\varepsilon(t)$	Enhancement layer
$\hat{\varepsilon}(t)$	The enhancement layer used for prediction at encoder
$\tilde{\varepsilon}(t)$	The received enhancement layer at decoder (with error)
$d(t)$	Distortion/transmission error of enhancement layer
$m.c._t\langle x\rangle$	Motion compensation operation at time instance t
$IQ \cdot Q\langle x\rangle$	Quantization and inverse quantization operation
$Trun.\langle x\rangle = \hat{x}$	Truncation operation
$\|x - y\|$	Norm between x and y. It is defined as sum of absolute difference in this article.

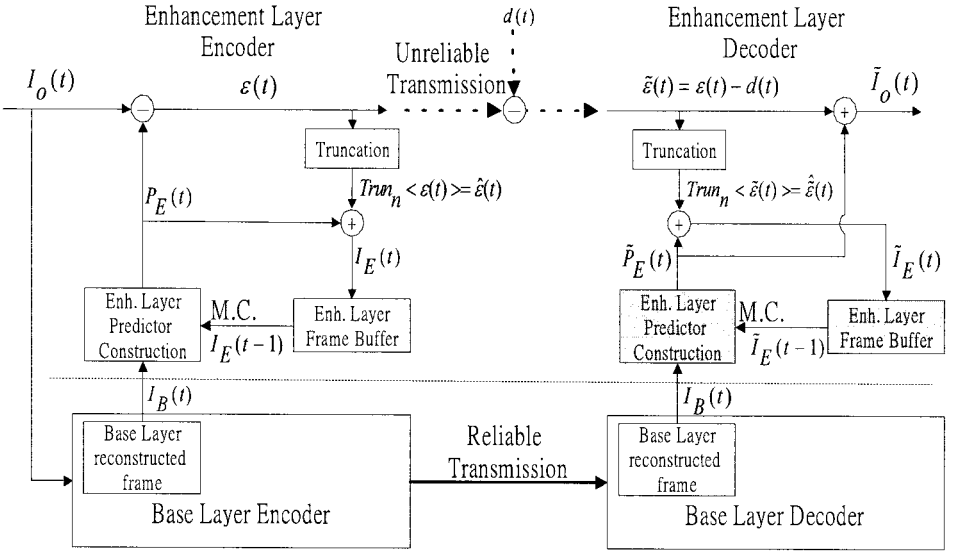

Figure 2. The proposed end-to-end streaming model.

To construct a better enhancement-layer predictor $P_E(t)$, we can optimally exploit all the available reconstructed frames at time t, as illustrated below:

$$P_E(t) = f(I_B(t), I_B(t-1), \ldots, I_E(t-1), I_E(t-2), \cdots). \quad (2)$$

Because Eq. (2) offers more selection of constructing predictors than Eq. (1) does, it is easier to minimize the enhancement layer prediction residue as

$$\min\|I_O(t) - P_E(t)\|. \quad (3)$$

Because the residue is smaller, the reconstructed enhancement-layer frame $I_E(t)$ will have better quality.

While the optimal predictor requires multiple frame buffers and motion compensation loops, our predictor is restricted to be constructed from the current base-layer frame and from the previous enhancement-layer frame, for lower complexity; that is,

$$P_E(t) = f(I_B(t), I_E(t-1)). \quad (4)$$

In this case, we only need two frame buffers and motion prediction loops.*

B. Predictor Mismatch Error between the Encoder and the Decoder. Although we can construct a better enhancement-layer predictor from the reconstructed enhancement-layer frames, using the reconstructed enhancement-layer frames as predictors could create a mismatch problem. This is because the decoder may not receive the enhancement layer in the expected manner.

When the decoder receives less enhancement layer, a distorted enhancement-layer predictor at decoder is reconstructed. Because the enhancement-layer predictor is constructed from the reconstructed base-layer frame as well as the reconstructed enhancement-layer frame, the enhancement-layer predictor becomes $\tilde{P}_E(t)$ instead of $P_E(t)$ at decoder side as shown below:

$$\tilde{P}_E(t) = f(I_B(t), \tilde{I}_E(t-1)). \quad (5)$$

The difference between $P_E(t)$ in Eq. (4) and $\tilde{P}_E(t)$ in Eq. (5) is the predictor mismatch between the encoder and the decoder.

The predictor mismatch will create errors in the decoded pictures. This is because the decoder output picture equals to the summation of the predictor and the residue, as the following:

$$\tilde{I}_O(t) = \varepsilon(t) + \tilde{P}_E(t). \quad (6)$$

In this case, even if we received the correct residue $\varepsilon(t)$ at time t, we cannot reconstruct prefect pictures without errors:

$$Error = I_O(t) - \tilde{I}_O(t) = \varepsilon(t) + P_E(t) - (\varepsilon(t) + \tilde{P}_E(t)) = P_E(t) - \tilde{P}_E(t). \quad (7)$$

C. Drifting Error and Accumulation Error. We can further illustrate the mismatch problem in more details, using a simple end-to-end transmission model shown in Figure 2. At the encoder, the enhancement-layer residue is

$$\varepsilon(t) \equiv I_o(t) - P_E(t) \quad \forall\, t \geq 0, \quad (8)$$

and its reconstructed frame for the future predictor construction is

$$I_E(t) \equiv Trun_n\langle\varepsilon(t)\rangle + P_E(t) = \hat{\varepsilon}(t) + P_E(t) \quad \forall\, t \geq 0. \quad (9)$$

Through an erasure transmission channel, the received enhancement layer at the decoder is modeled as the subtraction of an error term $d(t)$ from the original enhancement-layer as the following:

$$\tilde{\varepsilon}(t) \equiv \varepsilon(t) - d(t). \quad (10)$$

* For simplicities of the presentation, we will use Eq. (4) instead of Eq. (2) for the rest of the theoretic framework. Readers can easily replace Eq. (4) with Eq. (2) for more detailed theoretic derivation.

Therefore, the reconstructed enhancement-layer frame at the decoder for future predictor construction is

$$\tilde{I}_E(t) \equiv Trun_n\langle\tilde{\varepsilon}(t)\rangle + \tilde{P}_E(t) = \hat{\tilde{\varepsilon}}(t) + \tilde{P}_E(t) \quad \forall\, t \geq 0. \quad (11)$$

To illustrate the worse case of mismatch effect, we define the enhancement-layer predictor as the previously reconstructed enhancement-layer as below:

$$P_E(t) = f(I_B(t), I_E(t-1)) \equiv m.c._t(I_E(t-1)). \quad (12)$$

Accordingly, the equivalent predictor at the decoder can be written as the following:

$$\tilde{P}_E(t) = m.c._t(\tilde{I}_E(t-1)). \quad (13)$$

To represent the equivalent predictor at the decoder as function of received enhancement-layer, we substitute Eq. (11) into Eq. (13). After the recursive substitution, we have the following expression:

$$\begin{aligned}\tilde{P}_E(t) &= m.c._t(\tilde{I}_E(t-1)) = m.c._t(\hat{\tilde{\varepsilon}}(t-1) + \tilde{P}_E(t-1)) \\ &= m.c._t(\hat{\tilde{\varepsilon}}(t-1) + m.c._{t-1}(\hat{\tilde{\varepsilon}}(t-2) + m.c._{t-2}(\hat{\tilde{\varepsilon}}(t-3)\cdots \\ &\qquad + m.c._1((\hat{\tilde{\varepsilon}}(0) + \tilde{P}_E(0)))))). \quad (14)\end{aligned}$$

By further substituting Eq. (10) into Eq. (14), we group all the transmission errors together as the following:

$$\begin{aligned}\tilde{P}_E(t) &= m.c._t(\hat{\varepsilon}(t-1) + m.c._{t-1}(\hat{\varepsilon}(t-2) + m.c._{t-2}(\hat{\varepsilon}(t-3)\cdots \\ &\quad + m.c._1(\hat{\varepsilon}(0) + \tilde{P}_E(0))))) - m.c._t(\hat{d}(t-1) + m.c._{t-1}(\hat{d}(t-2) \\ &\qquad + m.c._{t-2}(\hat{d}(t-3)\cdots + m.c._1(\hat{d}(0))))) = P_E(t) \\ &\qquad - MismatchError, \quad (15)\end{aligned}$$

where $\tilde{P}_E(0) = P_E(0) = I_B(t)$ because the enhancement-layer predictor is from base layer at the first frame. The first term in Eq. (15) is the enhancement-layer predictor $P_E(t)$ at the encoder, and the grouped error terms become the equivalent predictor mismatch error, as the following:

$$MismatchError = P_E(t) - \tilde{P}_E(t) = \sum_{i=0}^{t-1} \hat{d}(i), \quad (16)$$

where we save the expressions of the motion compensations for notation simplicity.

The early transmission error creates two kinds of errors: (a) drifting error: single transmission error of enhancement layer at time j drifts to the predictors after time j, i.e., $\{P_E(t) : t > j\}$; and (b) accumulation error: the equivalent predictor mismatch error at time j is the accumulation of transmission errors before time j, i.e., $\sum_{i=0}^{j-1} \hat{d}(i)$. It is a consequence of drifting error and temporal prediction.

D. Formulation of Constraining Predictor Mismatch. Although a better predictor can bring coding gain at high bit rate, it could introduce drifting error at low bit rate. Hence, our goal is to find the $f(\,)$ in Eq. (4) that minimizes the enhancement-layer prediction residue, as described in Eq. (3),

$$\min\|I_o(t) - P_E(t)\|$$

and constraints the predictor mismatch described by the following:

$$\|P_E(t) - \tilde{P}_E(t)\| \leq Threshold. \quad (17)$$

Furthermore, in our practice, instead of a constant threshold to constrain drifting error as Eq. (17), we use a dynamic threshold, which is proportionally determined by the coding gain as the following:

$$\min(\lambda * \|I_o(t) - P_E(t)\| + \|P_E(t) - \tilde{P}_E(t)\|). \quad (18)$$

This new criterion chooses predictors that bring maximum coding gain while considering under drifting error. The threshold to constrain mismatch error becomes dynamic because it is proportionally determined by the coding gain. And, the factor λ is used as a trade-off factor for performance at high bit rate or at low bit rate.[†]

Table II. Proposed macroblock adaptive predictors.

Modes	Mathematical Representation
Type B	$P_E(t) \equiv I_B(t)$
Type E	$P_E(t) \equiv m.c._t\langle I_E(t-1)\rangle$
Type BE	$P_E(t) \equiv 0.5 * I_B(t) - 0.5 * m.c._t\langle I_E(t-1)\rangle$

III. PROPOSED SCHEME: MACROBLOCK ADAPTIVE PREDICTOR

While the previous section formulates the problem, this section describes our proposed scheme, including our new enhancement-layer predictor modes for better coding efficiency, our mechanisms to reduce drifting error, and our adaptive mode-selection scheme.

A. Enhancement-layer Predictor Modes for Better Coding Efficiency. To increase the coding efficiency, we need to minimize the prediction residue, as Eq. (3). Our method is to offer a set of better predictors through the available enhancement-layer frames.

We create three macroblock modes for enhancement-layer predictor. In addition to the predictor of MPEG-4 FGS, we have two additional predictors that utilize the previous enhancement-layer frame and the current base-layer frame. Their corresponding mathematical formulations are listed in Table II, and we describe the functionality of each mode as following:

- Type B: The predictor is from the current reconstructed base-layer frame. This is the same as current MPEG-4 FGS.
- Type E: The predictor from previously reconstructed enhancement-layer frame.
- Type BE: The predictor produced by the average of the previously reconstructed enhancement-layer frame and the current reconstructed base-layer frame.

We adaptively use these three predictors to minimize the prediction residue. For example, because the base-layer pictures are compressed at lower quality, the motion-compensated enhancement-layer frames often offer better quality, and thus Type E can be used to increase the picture quality of the predictor. On the other hand,

[†] In our implementation, λ equals to 1.5.

Table III. Generalized Type BE predictor.

Modes	Mathematical Representation[a]
Generalized Type BE	$P_E(t) \equiv (1 - \alpha)*I_B(t) + \alpha*m.c._t\langle I_E(t-1)\rangle$

[a] α is a real number between 0 and 1.

Type B is useful for video regions that motion estimation cannot efficiently reduce inter-frame correlation, e.g., fast-motion region, occlusion region, etc. Additionally, Type BE mode can improve coding efficiency by taking the best of Type B and Type E.

B. Drifting Error Reduction. To constraint the predictor mismatch as Eq. (17), we construct two mechanisms, fading and reset, from our Type BE and Type B predictors.

1. Fading Mechanism. Our fading mechanism is to have mismatch error propagates to later frames in a decaying manner. Mismatch errors far away from the current time instance will fade away, and therefore contribute only ignorable error. The fading mechanism comes form our Type BE predictor by only referring partial enhancement-layer for prediction.

Table III shows the generalized form of our Type BE. Through the generalized definition, Type E, Type BE, and Type B in Table II are simply cases with α being 1, 0.5, and 0. When we replace Type E predictor in Eq. (13) with generalized Type BE predictor, we get the predictor mismatch of generalized Type BE as the following:

$$MismatchError = P_E(t) - \tilde{P}_E(t) = \sum_{i=0}^{t-1} \alpha^{t-i}\hat{d}(i) \qquad (19)$$

by applying the same derivation in Section II.

When we set α between 0 and 1, the fading mechanism is enabled. Mismatch error at time i is therefore faded by a factor of α^{t-i}. Figure 3 shows the predictor mismatch error $\sum_{i=0}^{t-1} \alpha^{t-i}\hat{d}(i)$ versus time, where $\hat{d}(t)$ is assumed as a constant **C**. Figure 3 demonstrates that our fading mechanism can uniformly distribute the mismatch error and significantly reduce the amount of the accumulation error. Also, smaller α has stronger robustness to drifting error. This is because we use less enhancement-layer for prediction. In an extreme case, the enhancement-layer predictor of MPEG-4 FGS is with $\alpha = 0$. On the other hand, such predictor helps less on coding gain improvement.

In our scheme, we use $\alpha = 0.5$ for our Type BE to enable the fading mechanism.

2. Reset Mechanism. Our reset mechanism is to stop mismatch error propagates to a later frame. When we take generalized Type BE predictor with $\alpha = 0$, Eq. (19) equals to 0 as well. That is, mismatch errors from any previous time instance will disappear, and therefore there will be no drifting error. The generalized Type BE predictor with $\alpha = 0$ is our Type B predictor in Table II. By only referring base-layer for prediction, our reset mechanism breaks the inter-dependency among enhancement-layer frames.

C. Macroblock Predictor Selection. In order to enable drifting reduction mechanisms while maximizing the coding gain, we introduce a mode selection algorithm, which adaptively switches among our three predictors for each macroblock.

As stated in Eq. (3) and Eq. (17), our goal is not only to maximize the coding efficiency but also confine the drifting error. Therefore, while choosing prediction mode, we have to simultaneously consider the coding gain and the potential drifting loss for a predictor. To do so, we first assign each prediction mode with two factors, estimated improvement gain and estimated drifting loss.

1. Improvement Gain. In our approach, the improvement gain at high bit rate is approximated by the peak-signal-to-noise-ratio (PSNR) improvement of chosen predictor over Type B. We formulize the approximation as

$$PSNR_{gain} = 10*(\log(MSE_{TypeB}) - \log(MSE_{TypeX})) \approx 20*(\log(MSAD_{TypeB}) - \log(MSAD_{TypeX})), \qquad (20)$$

where MSE_{TypeX} and $MSAD_{TypeX}$ are mean square error (MSE) and average of sum of absolute difference (MSAD) between the chosen predictor Type X (X = B, BE, or E) and the source macroblock. To have less complexity, we use square of MSAD to approximate MSE.

2. Drifting Loss. To estimate the drifting loss at the decoder, we simulate the drifting behavior by exploiting a dummy predictor and dummy reference frame in the enhancement-layer prediction loop of encoder. In Table IV, we list the enhancement-layer predictor $P_E(t)$ and the reference frame $I_E(t)$ as well as their dummy terms. The enhancement-layer predictor $P_E(t)$ uses a higher number, e.g., 3, of bit-planes for the reference frame reconstruction. On the other hand, the dummy predictor $P_{E\_dummy}(t)$ takes a lower number, e.g., 1, of bit-planes to construct dummy reference frame $I_{E\_dummy}(t)$. In this case, we can create the mismatch on reference frames and simulate the drifting error at the decoder.

With the dummy predictor, the estimated drifting loss for a prediction mode is the following:

$$PSNR_{loss} = 10*(\log(MSE_{TypeX}) - \log(MSE_{TypeX\_dummy})) \approx 20*(\log(MSAD_{TypeX}) - \log(MSAD_{TypeX\_dummy})). \qquad (21)$$

To bring coding gain with drifting error consideration, our mode selection algorithm should take the prediction mode that can satisfy the following:

$$\max(\lambda*\textit{Estimated improvement gain} + \textit{Estimated drifting loss}). \qquad (22)$$

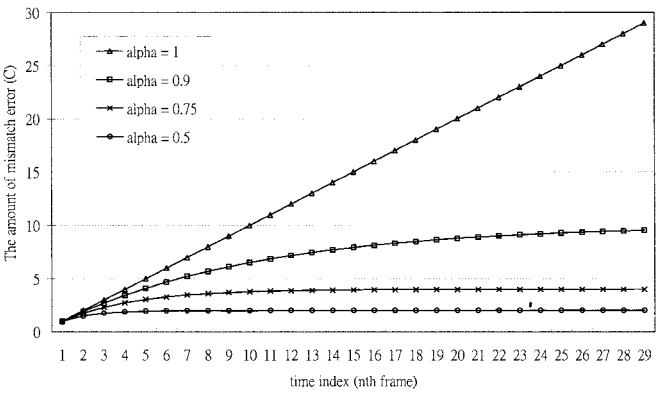

Figure 3. The mismatch error of enhancement-layer predictor versus time. Truncated mismatch error occurring at each frame is assumed as a constant **C**. The first frame is coded by intra-frame and the rest are by inter-frame.

Table IV. Dummy predictor and dummy reference frame for drifting loss estimation.

Improved Enh. layer predictor	$P_E(t) \equiv \alpha * m.c._t\langle I_E(t-1)\rangle + (1-\alpha) * I_B(t)$
Dummy Enh. layer predictor	$P_{E\_dummy}(t) \equiv \alpha * m.c._t\langle I_{E\_dummy}(t-1)\rangle + (1-\alpha) * I_B(t)$
Improved Enh. layer reference frame	$I_E(t) = P_E(t) + Trun._{n=H}(I_o(t) - P_E(t))$
Dummy Enh. layer reference frame	$I_{E\_dummy}(t) = P_{E\_dummy}(t) + Trun._{n=L}(I_o(t) - P_E(t))$

α is 0 for Type B, 0.5 for Type BE, and 1 for Type E. In our practice, $H = 3$ and $L = 1$.

IV. PREDICTION MODES ANALYSIS

In this section, we analyze the characteristic of the proposed predictors and their probability distributions for better coding efficiency at different motion characteristics, different base-layer qualities, and different enhancement-layer bit rates. Without considering the drifting error, we use the minimum SAD [as shown in Eq. (20)] as the criterion during profiling. The analysis is to understand which mode shows significant coding gain, how unequal the probability distribution is, and how many bit-planes should we use in the enhancement-layer for prediction.

A. Prediction Mode versus Motion Characteristic. Because the enhancement layer has better picture quality than the base layer, Type E and Type BE normally results in better coding gain than Type B (the only one in MPEG-4 FGS). In typical video sequences, more than 80% of the predictor modes are Type E or Type BE.

Type E is the dominating prediction mode (when we use three enhancement-layer bit-planes for prediction) among the three proposed prediction modes. This is because Type E predictor normally shows the most significant coding gain. However, the performance of Type E depends heavily on motion characteristic of the input sequence. Because Type E is constructed via the motion compensation, the efficiency of motion estimation affects the performance of Type E. Figures 4 and 5 show that Type E makes up 90% or more in slow-motion sequences, whereas the percentage drops to 50% in fast-motion sequences. Not only in fast-motion sequences, the percentage of Type E also drops when the frame rate decreases, as shown in Figure 6.

When Type E cannot efficiently predict the current block in low-frame-rates or fast-motion sequences, Type BE replaces Type E to improve coding efficiency improvement. The percentage of Type BE is normally higher than that of Type B (~20% vs ~10%), as shown in Figures 4 and 5. When the frame rate decreases, the percentage of Type BE increases. Thus, Type BE is competitive to Type E in low-frame-rate and fast-motion sequences.

Because the total percentage of Type E and Type BE is more than 80% over a wide range of sequences with different motion characteristics, we can expect that our proposed scheme will provide significant coding gain over MPEG-4 FGS.

B. Prediction Mode versus Base-layer Quality. In addition to the efficiency of motion estimation, the quality of the base layer also affects the prediction mode distributions. Particularly, when the base layer is not coarsely quantized, the quality of the base layer is higher. Figures 7 and 8 show the variations of the prediction mode distributions with different enhancement- and base-layer qualities. Comparing part (a) of Figures 7 and 8 with part (c) of Figures 7 and 8, or comparing part (b) of Figures 7 and 8 with part (d) of Figures 7 and 8 shows that Type B itself is a good predictor with finely quantized base layer. Moreover, because no motion compensation is needed in Type B predictor, Type B has no defect from nonideal motion compensation.

C. Prediction Mode versus Enhancement-layer Quality. Besides the motion characteristics and the base-layer quality, the number of bit-planes used for prediction also changes the prediction mode distribution. Figures 7 and 8 show the experiments where the enhancement-layer quality is proportional to the number of enhancement-layer bit-planes used for prediction, whereas the base-layer quality is controlled by the quantization parameter.

With less enhancement-layer bit-planes for prediction, the percentage of Type E over that of Type B is less. This is because the quality improvement of Type E predictor becomes less significant. Figures 7 and 8 even show that Type B has a comparable or even higher percentage than Type E when only one bit-plane is used for prediction. The phenomenon becomes even more obvious when the base layer is finely quantized as shown in parts (c) and (d) of Figures 7 and 8.

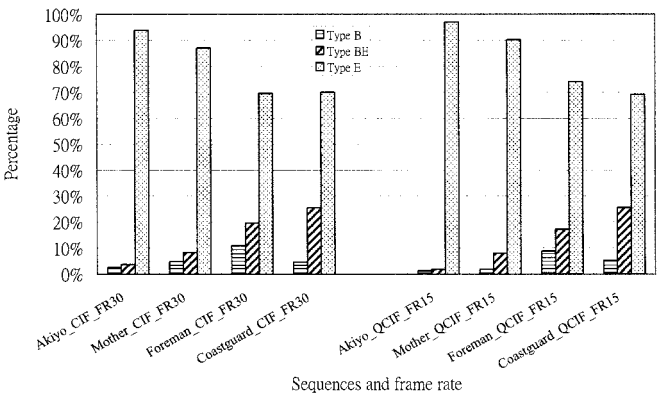

Figure 4. Prediction mode distributions with sequences of CIF format at 30 frames/s and QCIF format at 15 frames/s.

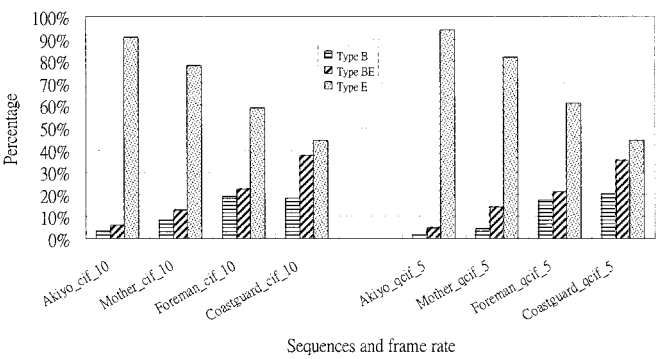

Figure 5. Prediction mode distributions with sequences of CIF format at 10 frames/s and QCIF format at 5 frames/s.

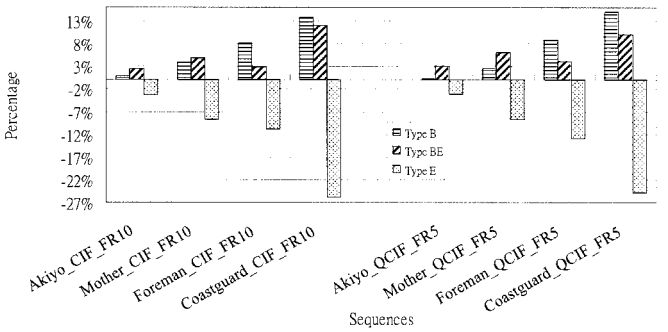

Figure 6. Variation of prediction mode distribution when lower frame rate. The frame rate is changed from 30 frames/s to 10 frames/s for CIF sequences and from 15 frames/s to 5 frames/s for QCIF sequence.

Table V summarizes the best proposed predictor in different scenarios. Our results are consistent with those in Rose et al. (2001).

D. Number of Enhancement-layer Bit-plane Used for Prediction. From the experiments in Figures 7 and 8, Type E and Type BE predictors are more dominant if more than two enhancement-layer bit-planes are used for prediction. The distributions become fairly stable if three or more bit-planes are used for prediction. Because Type E and Type BE are the predictors that provide significant coding gain, we believe that we should use at least three bit-planes in our scheme.

On the other hand, the amount of enhancement layer used for prediction determines the maximum mismatch error. Following our derivation in Section II, the maximum mismatch error happens when decoder does not receive any enhancement-layer, i.e., $\tilde{\varepsilon}(t) = 0$. In this case, we learn that $d(t) = \varepsilon(t)$ from Eq. (10). Thus, Eq. (16) can be expressed as the following:

$$\|P_E(t) - \tilde{P}_E(t)\| \leq \sum_{i=0}^{t-1} \|\hat{\varepsilon}(i)\|, \tag{23}$$

where the right-hand side of the inequality is the amount of the enhancement layer used for prediction. Apparently, using less enhancement layer for prediction can have less predictor mismatch. In an extreme case, the predictor mismatch error in Eq. (23) is zero when we do not use any enhancement layer for prediction. However, this brings less coding gain.

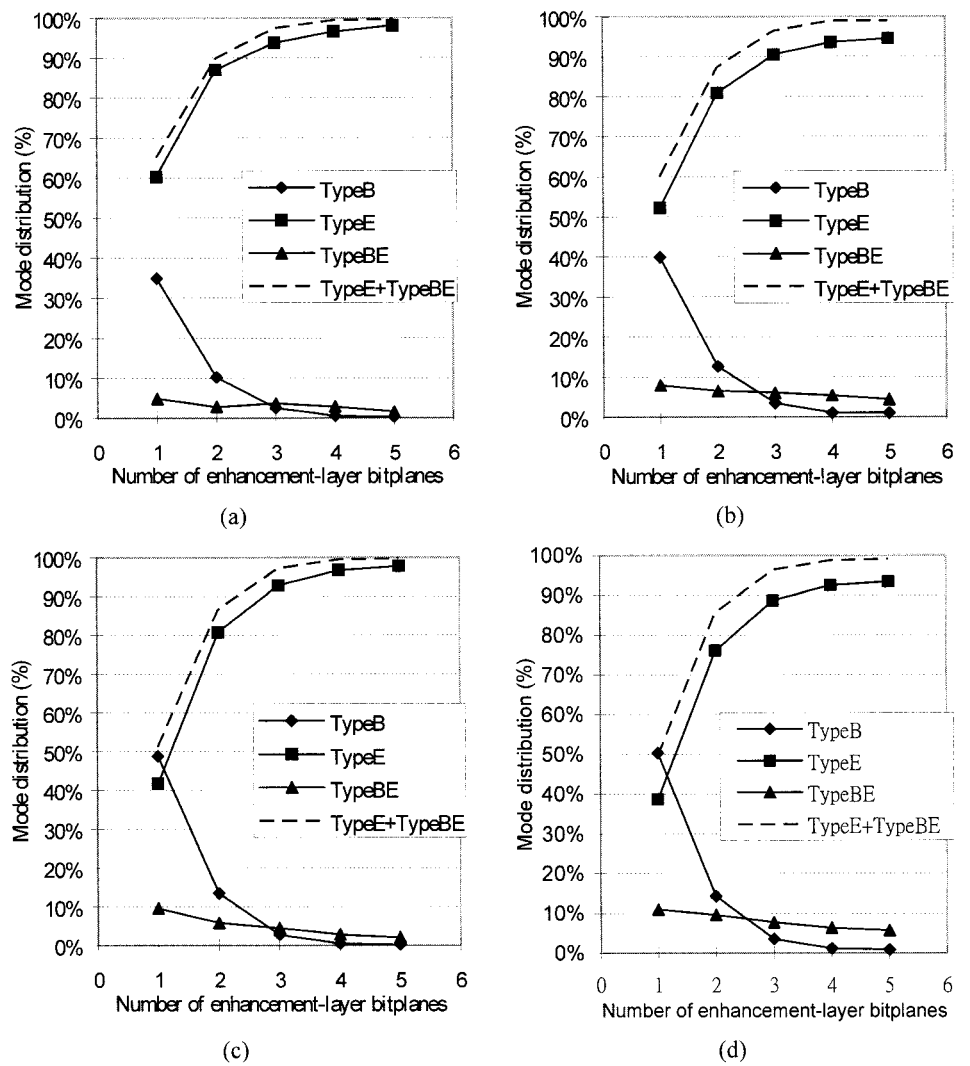

Figure 7. Prediction mode distribution of Akiyo versus frame rate, quantization parameter of base-layer and the number of enhancement-layer bitplanes for prediction. (a) Base-layer Qp = 31 and frame rate = 30 frames/s. (b) Base-layer Qp = 31 and frame rate = 10 frames/s. (c) Base-layer Qp = 15 and frame rate = 30 frames/s. (d) Base-layer Qp = 15 and frame rate = 10 frames/s.

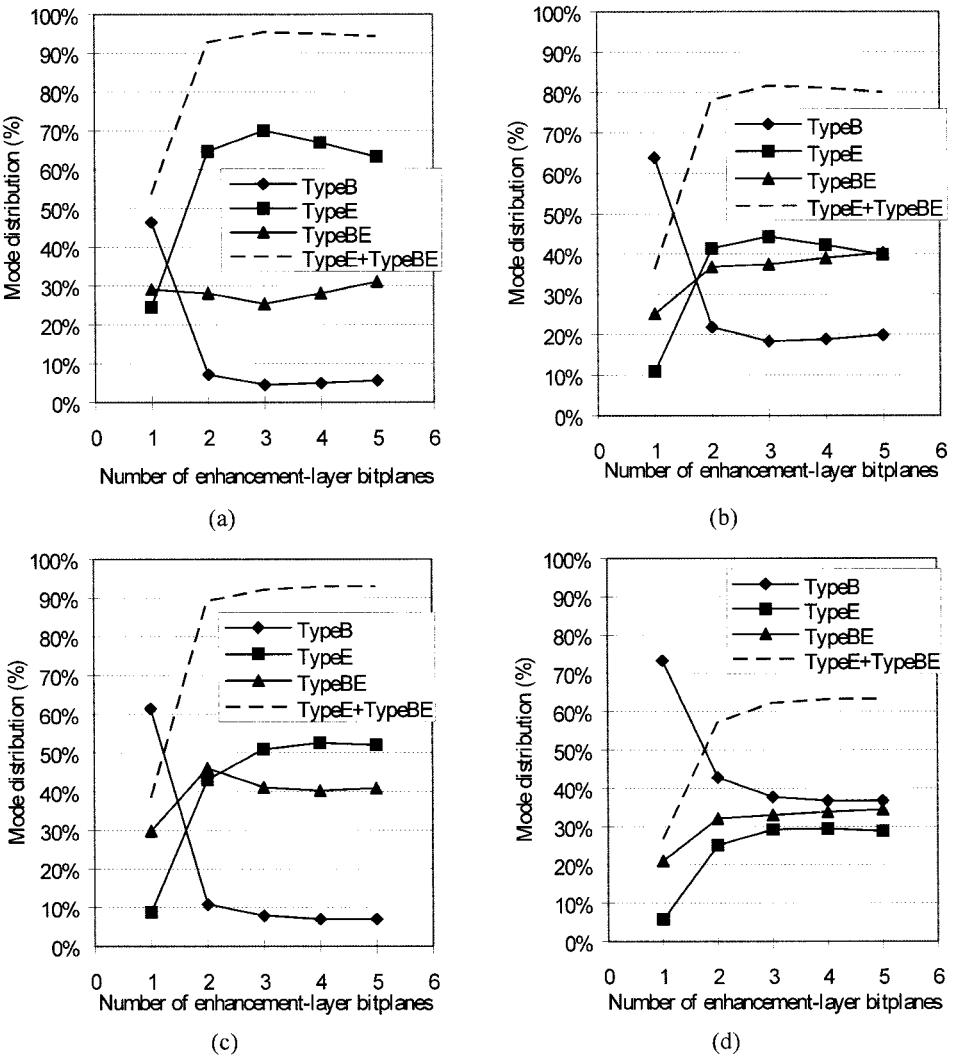

Figure 8. Prediction mode distribution of Coastguard versus frame rate, quantization parameter of base-layer and the number of enhancement-layer bit-planes for prediction. (a) Base-layer Qp = 31 and frame rate = 30 frames/s. (b) Base-layer Qp = 31 and frame rate = 10 frames/s. (c) Base-layer Qp = 15 and frame rate = 30 frames/s. (d) Base-layer Qp = 15 and frame rate = 10 frames/s.

To make a good trade-off between performances at different bit rates, we use no more than three bit-planes to construct Type E and Type BE. Our later experiment results show that using three bit-planes constantly can improve coding efficiency considerably.

E. Overhead of Our Coding Scheme. The flexibility of our predictor adaptation comes at the cost of additional side information. The extra syntax elements required are (1) the number of bit-planes used for enhancement-layer prediction and (2) the prediction mode for each macroblock.

Because the numbers of the bit-planes used for enhancement-layer prediction are transmitted at the frame level. The overhead is minor.

Table V. Best predictor at different scenarios.

Best Predictor	Type B	Type BE	Type E
Motion characteristic	Fast	Fast	Slow
Frame rate	Low	Low	High
Base-layer bit rate	High	Low	Low
Number of bit-planes for enhancement-layer prediction	Less	More	More

On the other hand, the prediction mode for each macroblock is required at macroblock level. Its overhead is considerable. Because our experiment results show that the prediction modes have unequal probabilities. Hence, using an entropy coding can have up to 50% improvement over the binary representation. In this case, the overhead is considerable, but not catastrophic.

V. OUR ENCODER AND DECODER ARCHITECTURES

A. Encoder. Like the MPEG-4 FGS, our EMFGS encoder has a base-layer encoder and an enhancement-layer encoder as in Figure 9. The base-layer encoder simply likes the one in MPEG-4 FGS.

Furthermore, our enhancement-layer encoder inserts an extra motion compensation loop, shown as solid lines in Figure 9, to produce the high-quality enhancement-layer reference frame. The enhancement-layer reference frame is reconstructed by refining the enhancement-layer predictor with the first **n** (a small number, e.g., 2 or 3) most-significant enhancement-layer bit-planes (MSB). Besides, we introduce a switch **M1** to offer adaptive predictors at the macroblock level. Table VI summarizes the switch configuration and its corresponding predictor.

To reduce drifting error, we produce the dummy reference frame by a dummy loop at the encoder as illustrated by the dash lines in

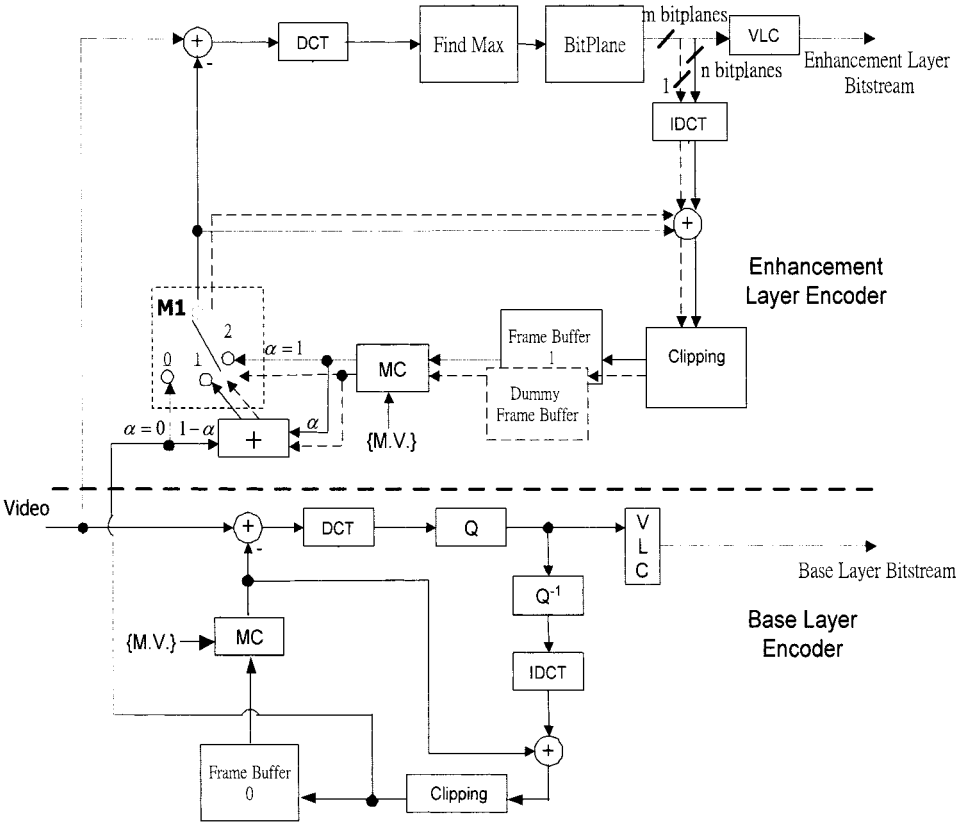

Figure 9. Encoder structure of mode-adaptive fine granularity scalability.

Figure 9. We set the dummy loop as an open loop for drifting error estimation. Table IV specifies the formation of dummy reference frame and dummy predictors.

B. Decoder. The proposed EMFGS decoder contains three parts, which are base-layer, lower enhancement-layer, and higher enhancement-layer decoders, as shown in Figure 10. The base-layer decoder and the higher enhancement-layer decoder are the same as the base-layer and enhancement-layer decoders in MPEG-4 FGS.

Additionally, the lower enhancement-layer decoder is inserted to reconstruct the high-quality enhancement-layer reference frame. There are one extra frame buffer and one extra switch. During decoding, the lower enhancement-layer decoder first constructs the predictor based on the prediction mode received. Then, we exploit the first **n** MSB bit-planes to reconstruct the high-quality enhancement-layer reference frame while using all the decoded bit-planes to construct the final output.

VI. COMPARISONS WITH OTHER ADVANCED FGS

After our proposed scheme has been described in details, this section compares and contrasts the major differences of our proposed EMFGS with two previous works, macroblock-based PFGS (Wu et al., 2001b) and RFGS (Huang et al., 2002). We first describe the algorithmic differences, e.g., the design of the enhancement-layer predictor, the enhancement-layer signal, and the drifting error reduction. Then, from these algorithmic differences, we further show the decoder complexity and implementation issues. Table VII summarizes these differences and shows the decoder complexity based on the number of extra operations per pixel required as compared to MPEG-4 FGS (N3315, 2000).

Table VI. Configurations of encoder/decoder switches and the corresponding prediction/reconstruction modes.

Configuration (M1)	Corresponding Predictor
(0)	Type B
(1)	Type BE
(2)	Type E

A. Enhancement-layer Predictor. To improve the coding efficiency, our EMFGS, PFGS (Wu et al., 2001b), and RFGS (Huang et al., 2002) exploit previous enhancement-layer frame $m.c.\langle I_E(t-1)\rangle$ to construct better predictor. However,

- Our EMFGS and PFGS (Wu et al., 2001b) allow different predictors for different macroblocks while RFGS (Huang et al., 2002) only has one kind of predictor.
- PFGS (Wu et al., 2001b) simply uses the motion-compensated previously reconstructed enhancement-layer frame as better predictor.
- Our EMFGS and RFGS (Huang et al., 2002) further include base-layer frames for predictor construction: RFGS (Huang et al., 2002) takes the motion-compensated previously-reconstructed base-layer frame $m.c.\langle I_B(t-1)\rangle$, and our Type BE predictor uses the current reconstructed base-layer frame $I_B(t)$.

First, it is shown in our prediction mode analysis that different macroblocks need different predictors to minimize the residue. Using one kind for the whole frame misses the opportunities to further

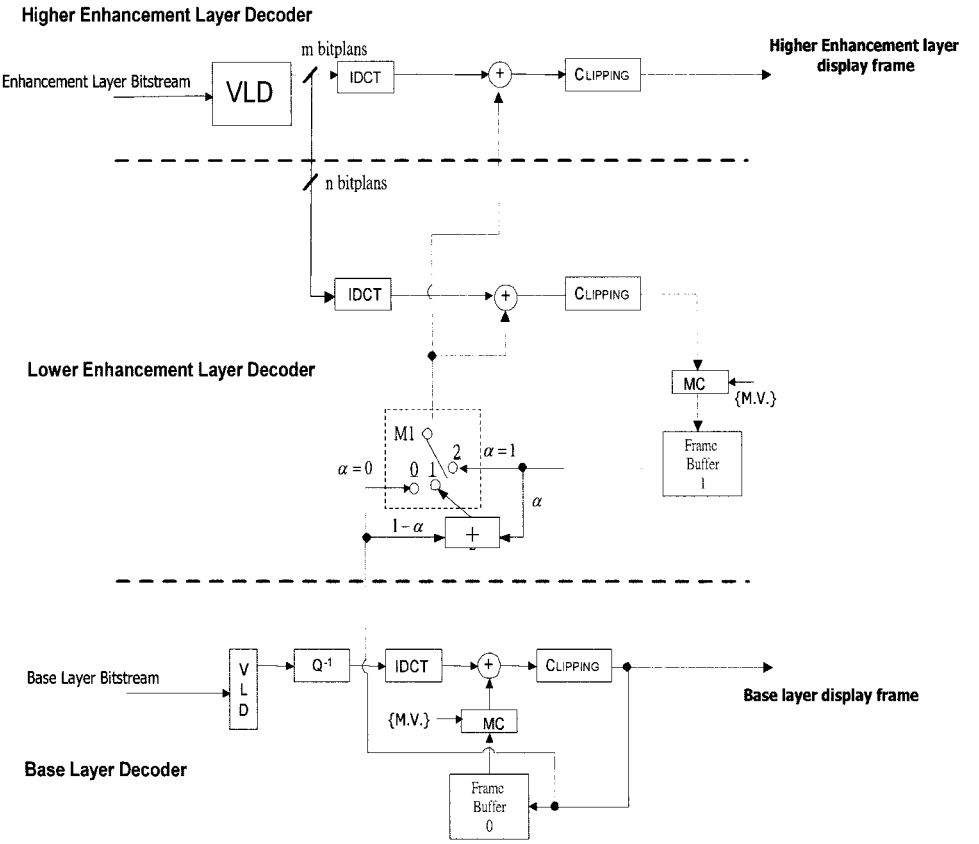

Figure 10. Decoder structure of mode-adaptive fine granularity scalability.

reduce the residue. Second, also as shown in our prediction mode analysis, Type BE predictors are not only better than Type B predictor, but also better than Type E predictors when motion estimation is not very efficient. In this case, it is important to construct the predictor from the reconstructed enhancement-layer frame and the reconstructed base-layer frame. Third, the current

Table VII. Comparison among three different advanced FGS schemes.

	PFGS (Wu et al., 2001b)	RFGS (Huang et al., 2002)	Our Proposed EMFGS
Prediction modes	Macroblock-based	Determined by intra/inter mode of base layer	Macroblock-based
Enh. layer predictor, $P_E(t)$	HPHR: $m.c.\langle I_E(t-1)\rangle$ LPLR: $I_B(t)$	$m.c.\langle \alpha I_E(t-1) + (1-\alpha) I_B(t-1)\rangle$	$m.c.\langle \alpha I_E(t-1)\rangle + (1-\alpha) I_B(t)$ Type B: $\alpha = 0$ Type BE: $\alpha = 0.5$ Type E: $\alpha = 1$
Enh. layer residue signal, $\varepsilon_E(t)$	$I_o(t) - P_E(t) - IQ \cdot Q\langle (I_o(t) - m.c.\langle I_B(t)\rangle)\rangle$	$I_o(t) - P_E(t) - IQ \cdot Q\langle (I_o(t) - m.c.\langle I_B(t)\rangle)\rangle$	$I_o(t) - P_E(t)$
Drifting reduction	Macroblock-based reset mechanism: LPLR mode HPLR mode	Frame-based fading mechanism: predictive leak factor α is chosen from floating point number between 0 and 1.	Macroblock-based reset and fading mechanism: Type B ($\alpha = 0$) Type BE ($\alpha = 0.5$)
Number of floating point or high-precision fixed point multiplications	0	1	0
Number of additions in spatial domain	1	3	1.1–1.4 (1 for Type E and Type B, and 2 for Type BE)
Number of additions in DCT domain	2	2	0
Number of frame buffers	1	1	1
Number of switches	2	0	1

The decoder complexity are summarized as the extra number of multiplications and additions per pixel as compared to the original MPEG-4 FGS decoder.

Table VIII. Typical alpha value used in RFGS (Huang, 2002).

Mother_Daughter CIF 10 Frames/s Base Layer 48 Kbits/s	Coastguard CIF 10 Frames/s Base Layer 48 Kbits/s
0.68750, 0.65625, 0.71875, 0.65625, 0.62500	0.71875, 0.68750, 0.78125, 0.75000, 0.81250, 0.90625, 0.87500, 0.84375, 0.78125

reconstructed base-layer frame has stronger correlation to the current source frame than the motion-compensated previously reconstructed base-layer frame does. Thus, combining the reconstructed enhancement-layer frame with the current reconstructed base-layer frame provides better coding efficiency.

B. Enhancement-layer Signal before Bit-plane Coding. After the enhancement-layer predictor is constructed, our EMFGS simply uses the enhancement-layer prediction residue (the difference between the original source frame and the enhancement-layer predictor) as our enhancement-layer signal. Nonetheless, both PFGS (Wu et al., 2001b) and RFGS (Huang et al., 2002) take the difference between the prediction residue of the base layer ($I_o(t) - P_E(t)$) and the prediction residue of the enhancement layer ($I_o(t) - P_B(t)$) as their enhancement-layer signal. Additional decoder complexity is introduced.

C. Drifting Error Reduction Schemes. Our EMFGS adaptively enables the reset mechanism and the fading mechanism to stop and decay the mismatch error, by using Type B ($\alpha = 0$) and Type BE ($\alpha = 0.5$) predictors. Our predictor-selection algorithm chooses the best predictor according to the performance at high bit rate and at low bit rate.

PFGS (Wu et al., 2001b) introduces an HPLR predictor to stop the mismatch error. The HPLR predictor artificially inserts mismatch error during encoding. In their decision mechanism to enable HPLR predictor (Wu et al., 2001c), the decision considers only the performance at low bit rate. More precisely, if the quality loss at low bit rate is larger than a given threshold, the HPLR predictor is used; otherwise, an HPHR predictor is used to get high coding efficiency.

RFGS (Huang et al., 2002) introduces a frame-based fading mechanism to decay the mismatch error. At each frame, a uniformly predictive leak factor, which is a floating-point number between 0 and 1, is applied to the enhancement-layer predictor. That is, although our approach uses simply one fading factor 0.5, RFGS (Huang et al., 2002) takes a more complex floating-point number to reach the best performance. Table VIII shows the typical α values used in their α-selection algorithm.

D. Decoder Complexity. After knowing the algorithmic differences, we further show that our EMFGS decoder is easier to implement than other advanced FGS schemes.

First, our approach uses the least number of high-precision multiplications:

- Our EMFGS does not require complex multiplications because of our simple fading factor, 0.5.
- PFGS (Wu et al., 2001b) needs no high-precision multiplications.
- On the other hand, the α-selection algorithm in RFGS (Huang et al., 2002) takes arbitrary floating-point numbers between 0 and 1. If their fading factors are quantized to 0, 0.5, or 1, the coding efficiency drops. Thus, RFGS (Huang et al., 2002) does require extra precision in multiplications.

Second, our approach uses the least number of additions in the DCT domain (additions in the DCT domain require a larger dynamic range):

- Our EMFGS does not require DCT-domain additions.
- Both PFGS (Wu et al., 2001b) and RFGS (Huang et al., 2002) need two DCT-domain additions to recover their enhancement-layer prediction residue for the final output and for the prediction.

Third, our approach uses a comparable number of additions in the spatial domain:

- To reconstruct enhancement-layer reference frame, our EMFGS decoder requires one addition for Type B, Type E and two additions for Type BE in spatial domain. Through the prediction mode analysis, our approach requires 1.1–1.4 extra additions on the average.
- PFGS (Wu et al., 2001b) needs one to reconstruct their enhancement-layer reference frame.
- RFGS (Huang et al., 2002) requires extra three additions in the spatial domain.

Fourth, our approach was a comparable number of switches:

- Our EMFGS requires one switch.

Table IX. Test conditions.

Sequences	Foreman, Coastguard, Table Tennis, Mother_Daughter		
Resolution	QCIF (176 × 144)	CIF (352 × 288)	CIF (352 × 288)
Frame rate	10 Hz	10 Hz	30 Hz
Base layer bit rate	32 kb/s	128 kb/s	256 kb/s
Max bit rate	160 kb/s	1024 kb/s	1536 kb/s
Period of I	Whole sequence	Whole sequence	59
Period of P	$M = 0$	$M = 0$	$M = 0$
Quantization	H263	MPEG	MPEG
Advanced MC	True	True	True
Original for ME	True	True	True
MV range	16	32	32
Rate control	TM5	TM5	TM5

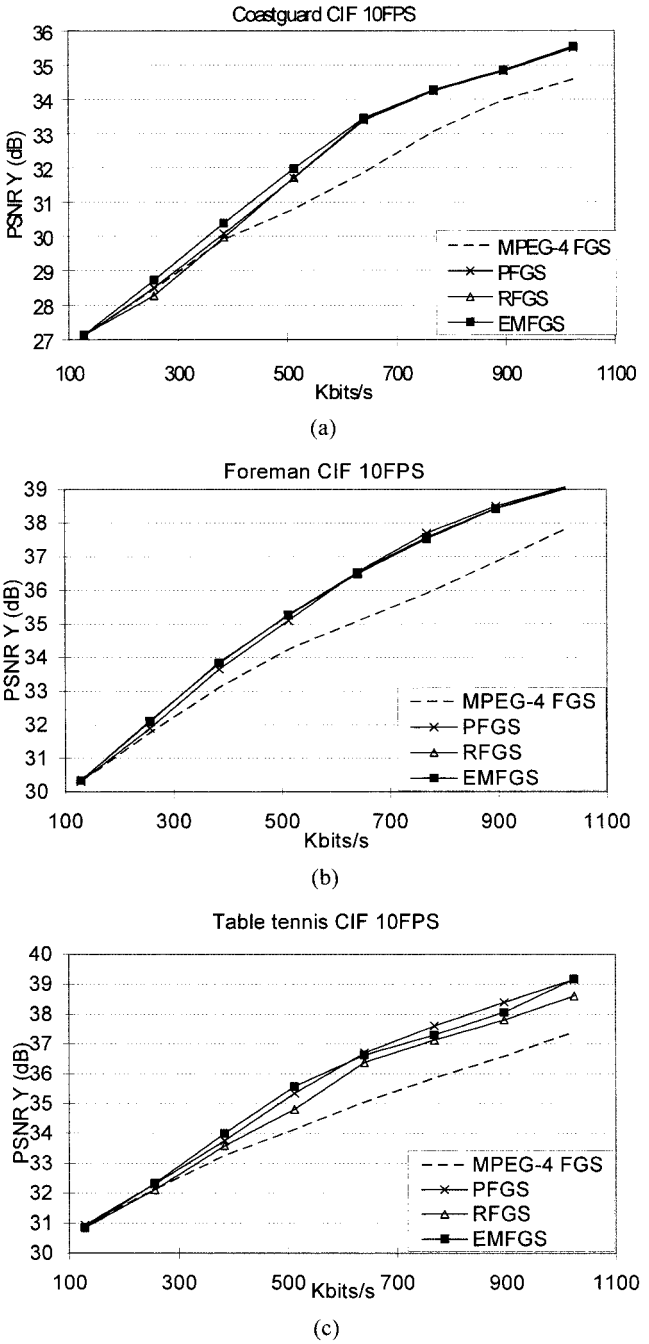

Figure 11. PSNR-Y comparison with PFGS (Wu et al., 2001b), RFGS (Huang et al., 2002) and MPEG-4 FGS using CIF sequences at 10 frames/s. The proposed approach is legend as EMFGS.

- PFGS (Wu et al., 2001b) needs two switches for its HPLR predictor.
- RFGS (Huang et al., 2002) does not have any switch because of using frame-based predictor.

Additionally, all schemes require one extra frame buffer for storing high-quality enhancement-layer reference frame. Table VII summarizes that our EMFGS scheme has a simpler structure.

VII. EXPERIMENT RESULTS

After knowing the details of our EMFGS, in this section, we assess the rate-distortion performance of the proposed codec objectively. In addition to comparing the performance of MPEG-4 FGS (N3315, 2000), we compare the performance of our EMFGS with that of PFGS (Wu et al., 2001b) and RFGS (Huang et al., 2002).

Table IX lists our test conditions, most of which are the same as those used in MPEG-4 committee (Wu et al., 2001b). Our measurement is based on PSNR. During the experiments, we keep only one

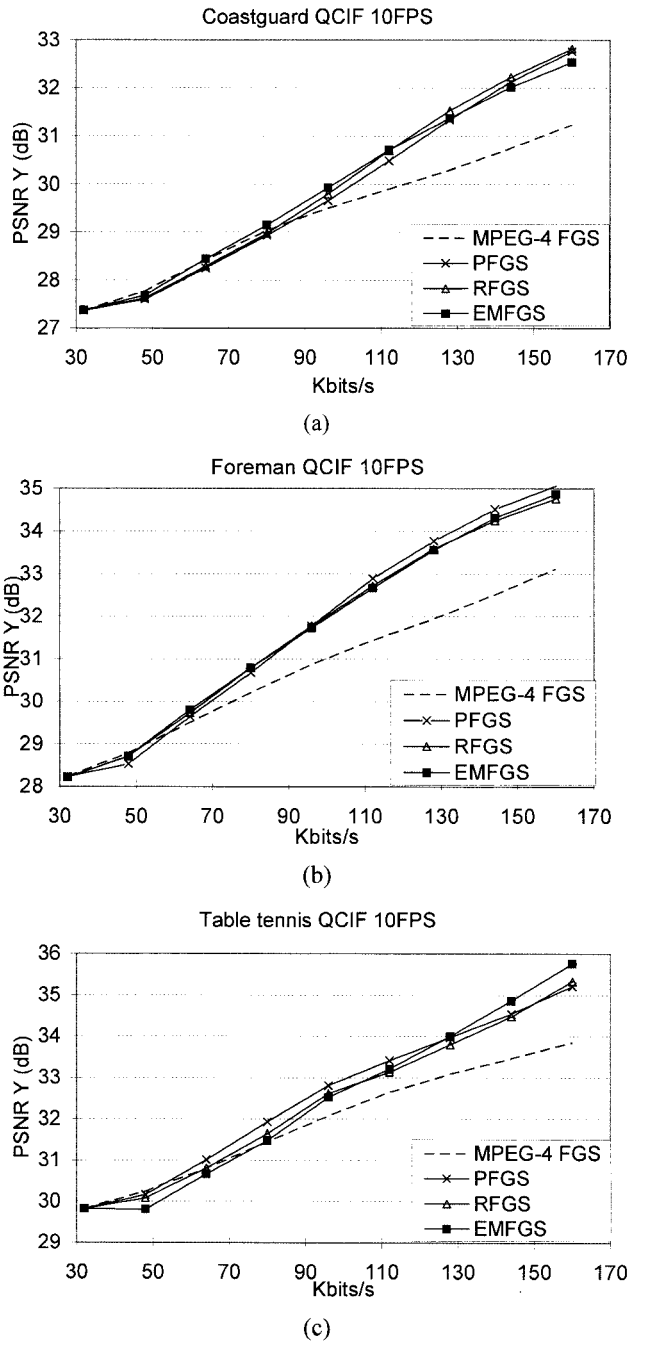

Figure 12. PSNR-Y comparison with PFGS (Wu et al., 2001b), RFGS (Huang et al., 2002) and MPEG-4 FGS using QCIF sequences at 10 frames/s. The proposed approach is legend as EMFGS.

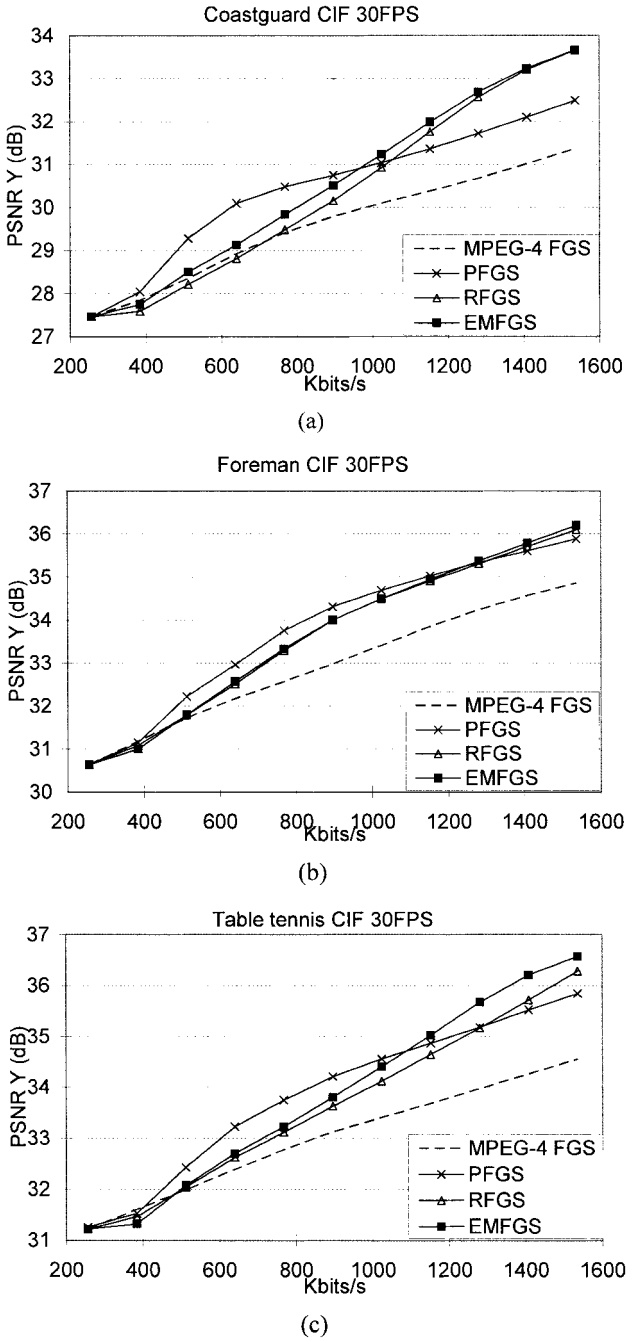

Figure 13. PSNR-Y comparison with PFGS (Wu et al., 2001b), RFGS (Huang et al., 2002) and MPEG-4 FGS using CIF sequences at 30 frames/s. The proposed approach is legend as EMFGS.

enhancement-layer bit-stream at the server side. To simulate the performance while the client is reached through different network bandwidths, our decoder truncates the pre-encoded enhancement layer bit-stream at multiple bit rates and measures the PSNR of decoded video respectively. Such truncation of enhancement layer is performed by the reference software offered by MPEG-4 committee. To comply with MPEG-4 base-layer, there is no prediction from the enhancement layer to the base layer. While α of our predictors, Type B, Type BE, and Type E are 0, 0.5, and 1 respectively, the α values for RFGS (Huang et al., 2002) are from their optimized linear model.

Figures 11–13 show the performance results of all four codecs. When comparing to MPEG-4 FGS, our EMFGS codec averagely improves the PSNR at medium bit rate by 1–2 dB and maintains the same or even better performance at low bit rate.

When comparing to other advanced FGS schemes, our EMFGS reaches the best performance in most cases for CIF sequences at 10 frames per second as shown in Figure 11. Our EMFGS keeps similar or better performance for QCIF sequences as shown in Figure 12. Figure 13 shows that our EMFGS has consistently better performance than RFGS (Huang et al., 2002) by 0.4 dB on the average for CIF sequences at 30 frames per second. Occasionally, PFGS (Wu et al., 2001b) gains more than our EMFGS at median-low bit rate and vice versa at high bit rate. The differences inherently come from different optimization trade-offs.

VIII. CONCLUSIONS

In this article, we proposed a scalable video coding algorithm, enhanced mode-adaptive fine granularity scalability (EMFGS), to deliver higher coding efficiency with less drifting errors. Particularly, we construct three macroblock prediction modes, Type B, Type BE, and Type E, from the previous enhancement-layer frame and the current base-layer frame. We provide a theory to demonstrate that while Type E and Type BE can significantly improve coding efficiency, Type BE and Type B can reduce and stop drifting error via the fading mechanism and the reset mechanism. By creating a dummy reference frame in the encoder, our mode-selection algorithm jointly optimize our performance over different bit rates. Our experiments show that our EMFGS can gain more than 2 dB in PSNR for slow-motion sequences and at least 1–1.5 dB for fast-motion sequences over MPEG-4 FGS.

While providing better quality, the proposed scheme has higher complexity than MPEG-4 FGS. Nevertheless, our complexity analysis shows that the proposed algorithm has limited increases in complexity and has a simpler structure than two other advanced FGS schemes. The computation complexity becomes a smaller issue due to the advance in computational power provided by the modern processors. We can afford to use more computation to yield better performance.

The work proves that the performance gap between the single layer codecs and the scalable codecs can be shortened. To further reduce the performance gap, motion estimation with variable block sizes, as in H.264 (Wiegand, 2003), could be incorporated at our enhancement layer. This is because the performance of EMFGS depends on the efficiency of the motion compensation. Also, depending on the application scenarios, the weighting factor between improvement gain and drifting error can be further adjusted. This leaves lot of spaces for future research activities.

ACKNOWLEDGMENT

We thank the anonymous reviewers for the constructive comments that allowed us to improve our manuscript. We also would like to thank Hsiang-Chun Huang, Chung-Neng Wang, and Tihao Chiang to provide their source code for our performance comparison.

REFERENCES

J.F. Arnold, M.R. Frater, and Y. Wang, Efficient drift-free signal-to-noise ratio scalability, IEEE Trans Circuits Syst Video Technol 11(1) (2000), 332–344.

Y. He, R. Yan, F. Wu, and S. Li, H.26L-based fine granularity scalable video coding, Proc IEEE Int'l Symp Circuits System 4 (2002), 548–551.

H.C. Huang, C.N. Wang, and T. Chiang, A robust fine granularity scalability using trellis based predictive leak, IEEE Trans Circuits Syst Video Technol 12(6) (2002), 372–385.

H. Jiang, Experiments on using post-clip addition in MPEG-4 FGS video coding, ISO/IEC JTC1/SC29/WG11, MPEG00/M5742 (2000).

W. Li, Bit-plane coding of DCT coefficients for fine granularity scalability, ISO/IEC JTC1/SC29/WG11, MPEG98/M4021 (1998).

W. Li, Overview of fine granularity scalability in MPEG-4 standard, IEEE Trans Circuits Syst Video Technol 11(3) (2001), 301–317.

W. Li et al., Advanced fine granularity scalability for high quality video distribution, ISO/IEC JTC1/SC29/WG11, MPEG01/M6766 (2001).

W.-H. Peng and Y.-K. Chen, Mode-adaptive fine granularity scalability, Proc IEEE Int'l Conf Image Process 2 (2001), 993–996.

Proposed Draft Amendment 4, ISO/IEC JTC1/SC29/WG11, MPEG00/N3315 (2000).

K. Rose and S.L. Regunathan, Toward optimality in scalable predictive coding, Proc IEEE Int'l Conf Image Process 3 (1998), 929–933.

K. Rose and S.L. Regunathan, Toward optimality in scalable predictive coding, IEEE Trans Image Process 10(7) (2001), 965–976.

B. Schuster, B. Thebault, W. Li, Y. Chen, and E. Francois, Fine granular SNR scalability: target applications, ISO/IEC JTC1/SC29/WG11, MPEG99/M4426 (1999).

M. van der Schaar and R. Kalluri, A hybrid temporal-SNR fine granular scalability for internet video, IEEE Trans Circuits Syst Video Technol 11(3) (2001), 318–331.

F. Wu, S. Li, and Y.Q. Zhang, A framework for efficient progressive fine granular scalable video coding, IEEE Trans Circuits Syst Video Technol 11(3) (2001a), 332–344.

F. Wu, S. Li, X. Sun, R. Yan, and Y.Q. Zhang, Macroblock-based progressive fine granularity scalable coding, ISO/IEC/JTC1/SC29/WG11, MPEG01/M6779 (2001b).

F. Wu, S. Li, B. Zeng, and Y.Q. Zhang, Drifting reduction in progressive fine granularity scalable video coding, Proc Int'l Picture Coding Symp (2001c).

T. Wiegand, Final Draft International Standard of Joint Video Specification, ITU-T/SG16/06, JVT-G050 (2003).