

# Performance Evaluation of Edge-based Video Error Concealment using H.264 Flexible Macroblock Ordering

Santosh Chapaneri<sup>\*1</sup>, Saurabh Mistry<sup>2</sup>, Sannidhi Dixit<sup>3</sup>

<sup>1</sup>Assistant Professor, <sup>2,3</sup>B.E. Final Year Students

Department of Electronics and Telecommunication Engineering

St. Francis Institute of Technology, University of Mumbai

Mumbai, India

santoshchapaneri@gmail.com

**Abstract**—This paper evaluates the directional interpolation scheme used widely for spatial error concealment in the H.264/AVC video coding standard using Flexible Macroblock Ordering (FMO). A mathematical analysis of FMO is presented to illustrate its effectiveness as an error resilience tool. Compared to the weighted pixel interpolation scheme of the test model, the directional interpolation scheme gives a relatively improved error concealment performance, since it preserves the edge direction of the damaged area by using the information from correctly received neighboring regions. However, the effectiveness of this technique depends on the method used for determining the dominant edge direction for the damaged area. Two methods are discussed, and simulations are conducted to test their performance for varied video content. Also, a variation of the commonly used PSNR quality metric is defined to obtain true performance results based on Jensen's inequality.

**Keywords**—Error concealment; H.264/AVC; FMO; Directional interpolation

## I. INTRODUCTION

In modern real-time communication systems involving video transmissions, the compressed video bit-streams are transmitted on networks that are sometimes unreliable and error-prone. Therefore, packet loss of encoded video over such networks may cause a strong visual degradation in the reproduction of video content at the receiver. Error concealment methods aim to reduce such degradations by making use of the redundancies in the temporal and spatial domain and attempt to recover the lost data based on the correctly received data without increasing the transmission overhead. Error concealment is a non-normative feature of the efficient H.264/AVC video coding standard [1], targeted for the decoder. Since the human eye is sensitive to picture distortions, the decoder must be able to conceal the corrupted area of the picture. Damaged portions of the Intra frames can result in error propagation to the subsequent Inter frames, if not concealed properly.

Spatial error concealment methods assume that the images are smooth in nature, and thus the lost image content can be reconstructed by interpolating from the neighboring pixels as proposed in [2, 3]. However, spatial interpolation approaches often suffer from blurring in the edge regions of the image. To resolve this problem, Suh et al. [4] proposed to find the edges first and then interpolate along the edge direction. Zhu et al. [5] proposed to use a second-order derivative-based method to reduce the blur across the edge while enforcing the smoothness along the edge. Not only interpolation can be applied to the spatial domain, but also to the spectral domain such as the discrete cosine transform (DCT) domain, as proposed in [6]. Some other methods are based on projection onto convex sets (POCS) that iteratively uses the smoothness assumption and pixel or DCT value information for error concealment [7]. Xu et al. [8] proposed the use of gradient filters to determine the dominant edge direction for the missing content and interpolating along that direction. The reference implementation of the H.264/AVC standard uses a simple weighted bilinear interpolation technique [9], but it does not take into account the edge related information.

The rest of this paper is organized as follows. A mathematical analysis of FMO is presented in Section 2 to illustrate why dispersed FMO is widely used for better error concealment. Section 3 explains the weighted bilinear interpolation technique used in H.264 reference software. An existing improved way of concealing the erroneous Intra frame portions using directional interpolation is discussed in Section 4, including two specific methods that have been reported in the literature. Since either of these techniques is adopted by different researchers, there is a need to determine which of these two methods usually performs better for a typical video sequence. We analyze these two methods and present the simulation results to suggest that the directional interpolation scheme performs well only when the dominant edge direction is calculated accurately. Also, a variation of the widely used PSNR metric is presented in Section 4 based on Jensen's inequality. Finally, the conclusion is presented in Section 5.

II. EVALUATION OF FLEXIBLE MACROBLOCK ORDERING

A. Flexible Macroblock Ordering Tool

During a regular encoding procedure, macroblocks (MB) are allocated using a raster scan pattern. Flexible Macroblock Ordering (FMO) is a method which is developed for flexible macroblock allocation [10, 11]. Hence, appropriate allocation pattern is selected according to error characteristics of channel. Using FMO, each MB can be assigned freely to a specific slice group (SG) (i.e. a set of slices) using a macroblock allocation map (MBAMap, also called the slice group map), which is included in the picture parameter set (PPS). The SGs constitute a new level in hierarchy from picture to MB, as shown in Fig. 1.

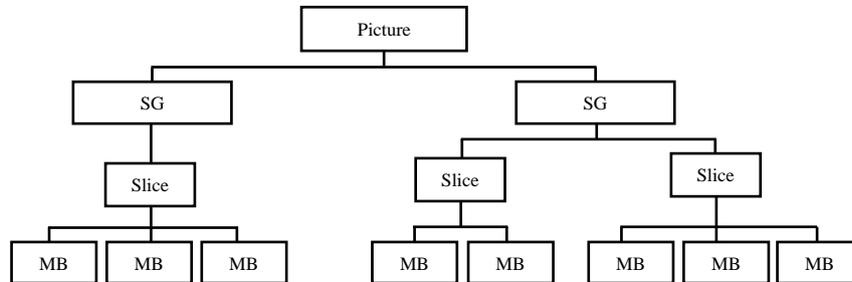


Figure 1. Hierarchy from picture to MB in H.264/AVC

With FMO, there can be up to eight SGs in one picture and within each SG, the MBs are coded in the default raster scan order. Also, the MBs within a certain SG can be grouped into several slices. The case where there is only one SG within a picture is identical to the case that FMO is not used at all. H.264/AVC standard specifies seven types of FMO of which only the first two types (interleaved and dispersed) are analyzed in this paper.

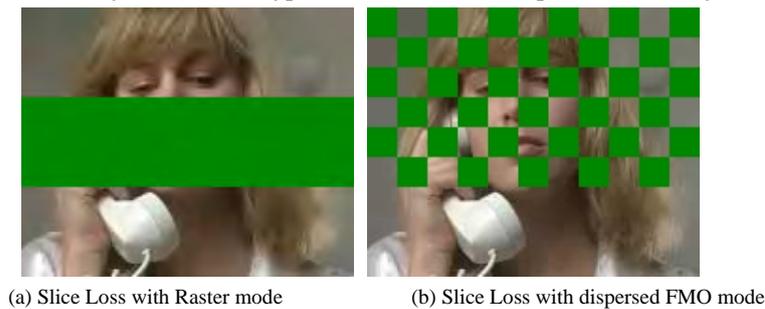


Figure 2. Illustration of FMO as an error resilience tool

Fig. 2 gives an example of the use of flexible macroblock ordering as an error resilience tool. The left image is encoded without FMO, while the right one is encoded with dispersed FMO mode. When transporting both the encoded streams, one of the packets got lost resulting in the rectangles indicating the loss in both images. It is clear that the right version is a lot easier to repair than the left version because of the available extra information around each missing MB. Interleaved and dispersed FMO have been reported widely in the literature to provide improved error concealment as opposed to not using FMO. We next develop a mathematical framework for FMO to demonstrate its use in error concealment. Specifically, we focus on the raster scan (no FMO), interleaved FMO and dispersed FMO cases.

B. Mathematical Analysis of FMO

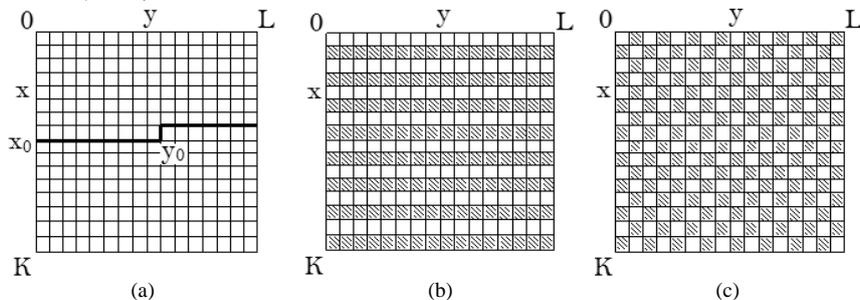


Figure 3. Illustration for FMO analysis. (a) Raster scan (no FMO), two slices. (b) Interleaved FMO, two SGs. (c) Dispersed FMO, two SGs

Let K and L be the number of MBs in the vertical and in the horizontal direction, respectively. Let (x, y) denote the MB coordinate with  $0 \leq x \leq K$ ,  $0 \leq y \leq L$  and  $mb(x, y)$  represent the MB at the coordinate (x, y). If the

size of MB is  $N$  by  $N$ , then  $N$  is 16 and 8 for the luminance and chrominance components, respectively. The macroblock  $mb(x, y)$  has pixels at coordinates  $(i, j)$  such that  $(N-1)x \leq i < Nx$  and  $(N-1)y \leq j < Ny$ . In this formulation, two slices per picture case is presented for comparison with and without FMO. Other cases can be similarly defined and compared. Fig. 3 illustrates the coding pattern for the raster scan, interleaved and dispersed FMO types for a MB. For the raster scan case, consider that the picture has two slices  $S_0$  and  $S_1$ , and let  $mb(x_0, y_0)$  be the last MB of  $S_0$ . The raster scan is defined as

$$S_0 = \{mb(x, y): 0 \leq x < x_0, 0 \leq y < L\} \cup \{mb(x, y): x = x_0, 0 \leq y \leq y_0\} \quad (1)$$

$$S_1 = \{mb(x, y): x_0 < x < K, 0 \leq y < L\} \cup \{mb(x, y): x = x_0, y_0 \leq y < L\} \quad (2)$$

With FMO, we have slice groups consisting of one or more slices. In this formulation, each SG contains only one slice. For the interleaved FMO case, consider that the picture has two slice groups  $S_{0i}$  and  $S_{1i}$ . The interleaved FMO is defined as

$$S_{0i} = \{mb(2x, y): 0 \leq x < K/2, 0 \leq y < L\} \quad (3)$$

$$S_{1i} = \{mb(2x+1, y): 0 \leq x < K/2, 0 \leq y < L\} \quad (4)$$

For the dispersed FMO case, consider that the picture has two slice groups  $S_{0d}$  and  $S_{1d}$ . The dispersed FMO is defined as

$$S_{0d} = \{mb(2x, 2y) \cup mb(2x+1, 2y+1): 0 \leq x < K/2, 0 \leq y < L/2\} \quad (5)$$

$$S_{1d} = \{mb(2x, 2y+1) \cup mb(2x+1, 2y): 0 \leq x < K/2, 0 \leq y < L/2\} \quad (6)$$

The performance of any error concealment scheme depends on the number of correctly received MBs, which in turn depends on the FMO type. Each slice (in raster scan case) or slice group (in FMO case) containing a group of MBs is transmitted in one NAL packet. We theoretically determine the expected number of correctly received neighboring MBs for a given packet loss rate. The following analysis shows that the performance of error concealment is closely related to the FMO type. Let  $n^r(x, y)$ ,  $n^i(x, y)$ , and  $n^d(x, y)$  represent the number of correctly received neighboring MBs when raster scan, interleaved, and dispersed FMO type is used, respectively. Let  $p$  denote the probability of packet loss. If  $S_0$  is lost with probability  $p$ , then  $S_1$  is received with probability  $1-p$ . Let  $E[n^r]$ ,  $E[n^i]$ , and  $E[n^d]$  represent the expected number of correctly received neighboring MBs.

- i) **Raster scan (no FMO)**: Assume that a new slice begins at the first MB in a row i.e.  $y_0 = L-1$ . If  $mb(x, y)$  in  $S_0$  is lost and  $S_1$  is received, then

$$n^r(x, y) = \begin{cases} 1, & \text{if } x = x_0 \\ 0, & \text{otherwise} \end{cases}, \quad E[n^r] = \frac{1-p}{x_0+1} \sum_{x=0}^{x_0} n^r(x, y) \quad (7)$$

When  $mb(x, y)$  in  $S_1$  is lost and  $S_0$  is received,  $n^r(x, y)$  and  $E[n^r]$  can be similarly represented and calculated.

- ii) **Interleaved FMO**: If  $mb(x, y)$  in  $S_{0i}$  is lost and  $S_{1i}$  is received, then

$$n^i(x, y) = \begin{cases} 1, & \text{if } x = 0 \\ 2, & \text{otherwise} \end{cases}, \quad E[n^i] = \frac{1-p}{K/2} \sum_{x=0}^{K/2} n^i(x, y) \quad (8)$$

$E[n^i]$  will be the same as (10) when  $mb(x, y)$  in  $S_{1i}$  is lost and  $S_{0i}$  is received.

- iii) **Dispersed FMO**: If  $mb(x, y)$  in  $S_{0d}$  is lost and  $S_{1d}$  is received, then

$$n^d(x, y) = \begin{cases} 3, & \text{if } x = 0 \\ 4, & \text{otherwise} \end{cases}, \quad E[n^d] = \frac{1-p}{K/2} \sum_{x=0}^{K/2} n^d(x, y) \quad (9)$$

$E[n^d]$  will be the same as (12) when  $mb(x, y)$  in  $S_{1d}$  is lost and  $S_{0d}$  is received.

Consider the following case as an example for the expected number of correctly received neighboring MBs. When the luminance component of CIF video sequence is considered,  $K = 22$  and  $L = 18$ . Assume that the packet loss rate is 10% and the slice size in raster scan case is half, i.e.  $x_0 = 10$ . From (7), (8), and (9), we get the values  $E[n^r] = 0.082$ ,  $E[n^i] = 1.7$ , and  $E[n^d] = 3.5$ . This analysis shows that the expected number of correctly received neighboring MBs becomes larger in FMO cases than in the raster scan case, and dispersed FMO gives larger expected number than the interleaved FMO type. Thus, we can expect that the error concealment performance would be better when using FMO rather than raster scan, and the dispersed FMO type would give better concealment performance than the interleaved FMO type.

### C. Evaluation of FMO

We verify this analysis experimentally using JM ver. 17 reference software of H.264/AVC [9] and the standard *Foreman* CIF video sequence having one hundred frames. Real-time Transport Protocol (RTP) is used as the output file mode for the network abstract layer. Each RTP packet contains one NAL unit. Arbitrary

packets are dropped (lost) from the encoded bitstream file composed of RTP packets at the common loss rates of 0%, 3%, 5%, 10%, and 20% [12]. Fig. 4 shows the concealment performance of the reference software for the *Foreman* sequence coded with the raster scan, interleaved and dispersed FMO types. Only two slices in raster scan case and two slice groups (each having one slice) in FMO case are used to verify the theoretical analysis.



Figure 4. FMO performance on *Foreman* frame 10. (a) Raster scan (no FMO). (b) Interleaved FMO. (c) Dispersed FMO.

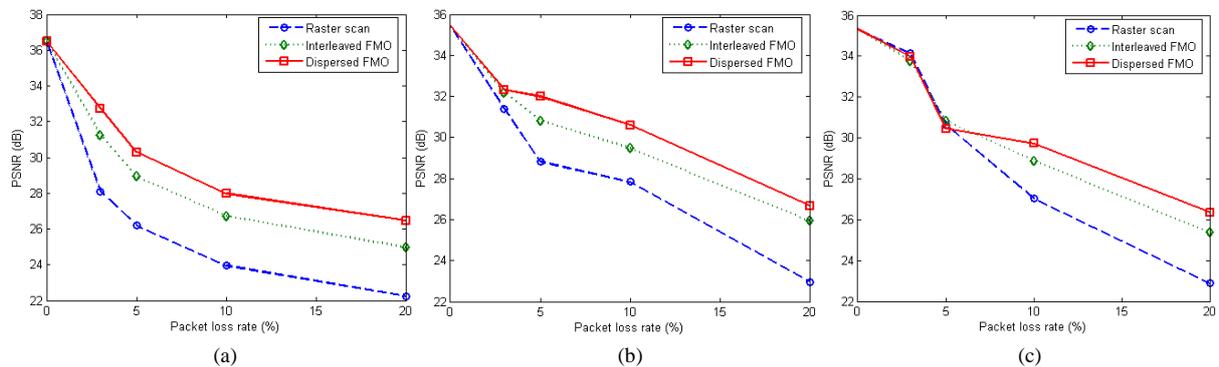


Figure 5. (a) Average PSNR at variable packet loss rates with Intra-frame period of 1, (b) Average PSNR at variable packet loss rates with Intra-frame period of 10, (c) Average PSNR at variable packet loss rates with Intra-frame period of 20

Fig. 5(a) shows the average luminance PSNR performance for variable packet loss rates for the *Foreman* sequence when the Intra-frame period is 1, i.e. every frame is coded as Intra. From this result, we can see that the PSNR performance is very close at 0% packet loss rate, and interleaved and dispersed FMO types give better PSNR than raster scan as the packet loss rate becomes larger. At 20% packet loss rate, both FMO types give about 3 ~ 4 dB higher PSNR than the raster scan, and the dispersed type gives slightly higher PSNR than the interleaved type. Fig. 5(b), (c) illustrates the average luminance PSNR with variable packet loss rates when the Intra-frame period is 10 and 20, respectively, for the *Foreman* sequence. The remaining frames between the Intra-frames are coded as Inter or P-frames. The overall tendency is similar, but we can observe that as the Intra-frame period increases, the impact of FMO is observed significantly at higher packet loss rates. This is because of the severe temporal error propagation effect which occurs in the Inter-frames at higher packet loss rates. Thus, we can conclude both theoretically and experimentally that FMO aids in the error concealment performance and dispersed FMO gives better performance than interleaved FMO. In the rest of this work, we use dispersed FMO type during encoding of the video sequences to achieve a better concealment output.

### III. BILINEAR INTERPOLATION EC

The spatial error concealment algorithm proposed in the H.264 test model is based on weighted-pixel bilinear interpolation [13]. It estimates the value of a lost pixel from the pixels in four adjacent macroblocks on the one-pixel-wide boundary as shown in Fig. 6.

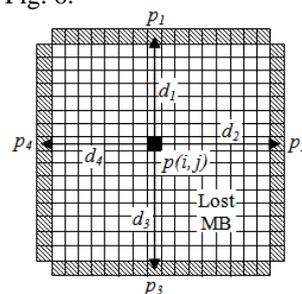


Figure 6. Weighted bilinear interpolation

Consider a lost/damaged macroblock (MB) of an Intra frame. Assuming that the four neighboring MBs are decoded correctly, the pixel value of the damaged MB is estimated as

$$p(i, j) = \frac{\sum_{m=1}^4 p_m(15-d_m)}{\sum_{m=1}^4 d_m} \quad (10)$$

where  $p(i, j)$  is the reconstructed pixel value,  $0 \leq i \leq 15$  and  $0 \leq j \leq 15$ , taking the conventional macroblock size of 16 by 16. The neighboring boundary pixels are denoted by  $p_1, p_2, p_3, p_4$  and their corresponding distances to  $p(i, j)$  are  $d_1, d_2, d_3, d_4$ . Only the correctly received or decoded MBs are used for concealment if at least two such MBs are available. Otherwise, concealed neighboring MBs are also used. The weighted bilinear interpolation performs well only when the missing MB is in a smooth region; otherwise it results in visible blocking or blurring artifacts as illustrated in Fig. 7.



Figure 7. Blocking artifacts due to bilinear interpolation

Also, this technique does not take into consideration the edge direction of the concerned region, even though edge integrity is an important aspect of visual perception. To overcome this drawback, several approaches have been proposed, including projection onto convex sets [7], directional interpolation and its variants [8, 14]. Here, we consider the more popular technique of concealment using directional interpolation based on the information of neighboring edges.

#### IV. EDGE-BASED DIRECTIONAL EC

##### A. Edge Detection

The principle of directional interpolation is to find the correct edge trend through the lost area, thus resulting in smoother concealment. We use the Sobel operator as the gradient filter [15], which has good edge detection characteristics.



Figure 8. Applying Sobel operator to (left) *Foreman* frame, (middle) all detected edges, (right) prominent edges with threshold of 100

An example of the implementation of Sobel operator as edge detector is demonstrated in Fig. 8 where the Sobel operator is applied at each pixel of the frame. In the middle and right figures, the white area shows the maximal edge strength and black area shows the smooth area of the frame. In order to detect only the prominent edges and to weed out false edges, only the pixels having gradient magnitudes above a certain threshold are classified as edge pixels. Consider an area  $A$  comprising the pixels located on the second outer boundary of the lost MB shown as shaded in Fig. 9. The second outer boundary is chosen so as to use all of the eight pixel neighbors for gradient computations.

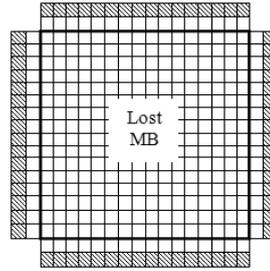


Figure 9. Pixels in shaded area used for gradient computations

The following horizontal and vertical Sobel masks are applied to all the pixels in area A:

$$\mathbf{S}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{S}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (11)$$

The gradient vector  $\vec{G}$  of a pixel is represented by two components, the horizontal gradient  $G_x$  and the vertical gradient  $G_y$ , which are defined as

$$G_x(i, j) = \mathbf{F}(i, j) * \mathbf{S}_x, \quad G_y(i, j) = \mathbf{F}(i, j) * \mathbf{S}_y \quad (12)$$

where  $\mathbf{F}(i, j)$  is the relevant neighborhood for each pixel  $f(i, j)$  in the area A:

$$\mathbf{F}(i, j) = \begin{bmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) \\ f(i, j-1) & f(i, j) & f(i, j+1) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) \end{bmatrix} \quad (13)$$

The gradient magnitude and direction of each pixel  $f(i, j)$  is given by

$$|\vec{G}(i, j)| = \sqrt{G_x^2(i, j) + G_y^2(i, j)}, \quad \theta(i, j) = \arctan\left(\frac{G_y(i, j)}{G_x(i, j)}\right) \quad (14)$$

### B. Dominant Edge Direction

To find the dominant edge direction using gradient computation, two techniques have been used extensively in the literature: i) **Mean** method [8] and ii) **Mode** method [14].

i) In the **Mean** method, the dominant edge direction is computed by averaging the edge directions weighted by the corresponding gradient magnitudes of all the pixels in area A as follows:

$$\theta_d = \frac{\sum \theta(i, j) |G(i, j)|}{\sum |G(i, j)|}; \quad \forall (i, j) \in A \quad (15)$$

ii) In the **Mode** method, the circle is quantized into eight directions as shown in Fig. 10(a). For each such direction, a counter is initialized to zero. For each pixel where the gradient is computed, the counter corresponding to its gradient direction is increased by the corresponding gradient magnitude. Then, the dominant edge direction is the angle yielding the largest counter value. This implies that the dominant direction is the mode of all possible gradient angles obtained with respect to the gradient magnitudes.

Traveling along the dominant edge direction as determined by either method mentioned above, the detected edge is extended from the pixel in the lost MB to the neighboring pixels on the one-pixel-wide outer boundary of the lost MB as shown in Fig. 10(b). The concealment of lost pixels is then done using interpolation along the extended edge as

$$p(i, j) = \frac{p_1 d_2 + p_2 d_1}{d_1 + d_2}; \quad \forall (i, j) \in \text{lost MB} \quad (16)$$

where  $p_1$  and  $p_2$  are the boundary pixel values and  $d_1$  and  $d_2$  are the corresponding distances from the desired pixel  $p(i, j)$ .

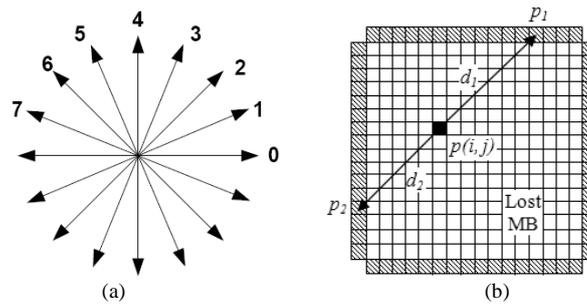


Figure 10. Directional Interpolation. (a) Gradient edge directions, (b) Extending the detected edge for concealment.

C. Objective Quality Metrics

To compare the performance of the *Mean* and *Mode* method, we use two quality metrics: PSNR and a structural similarity index SSIM.

The simplest and most widely used full-reference quality metric is the mean squared error (MSE), computed by averaging the squared intensity differences of distorted and reference image pixels, along with the related quantity of peak signal-to-noise ratio (PSNR). MSE and PSNR are widely used because they are simple to calculate, have clear physical meanings, and are mathematically easy to deal with for optimization purposes. There are different ways of representing PSNR. The most common way is to represent a frame-by-frame versus PSNR graph [16]. Another typical way is to give the information about PSNR e.g. in a table or graph for the overall PSNR of the sequence [17]. In our opinion, it is better option to represent PSNR frame-by-frame because taking the mean of PSNR for every frame can be deceptive. It only gives a reliable result when all frames of the sequence are of similar character (spatial and temporal information) and if the error probability is constant within the whole sequence. If frames differ substantially, then the mean PSNR can become skewed. The H.264/AVC reference software outputs PSNR for every component *c* of the YUV color space (Y-PSNR, U-PSNR and V-PSNR) for every frame *k* as

$$PSNR_k^{(c)} = 10 \cdot \log_{10} \frac{255^2}{MSE_k^{(c)}} [dB] \tag{17}$$

$$MSE_k^{(c)} = \frac{1}{M \cdot N} \sum_{i=1}^N \sum_{j=1}^M [F(i, j) - F_o(i, j)]^2 \tag{18}$$

where *N* by *M* is the size of a frame, *F* is the reconstructed frame and *F<sub>o</sub>* is the original frame (uncompressed and without losses). For the total number of frames *N<sub>fr</sub>*, the reference software also calculates the average over all frames for the luminance and the two chrominance components given by

$$PSNR_{av}^{(c)} = \frac{1}{N_{fr}} \sum_{k=1}^{N_{fr}} PSNR_k^{(c)} \tag{19}$$

We define a variation of calculating the PSNR average value as

$$PSNR_{av} = 10 \cdot \log_{10} \frac{255^2}{MSE_{av}} [dB] \tag{20}$$

$$MSE_{av} = \frac{1}{3 \cdot M \cdot N \cdot N_{fr}} \sum_{c=1}^3 \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{N_{fr}} [F_k^{(c)}(i, j) - F_{o,k}^{(c)}(i, j)]^2 \tag{21}$$

Some definitions are presented to illustrate that the modified metric (20) gives true performance results.

**Definition** A function *f(x)* is said to be convex over an integral (*a, b*) if for every *x<sub>1</sub>, x<sub>2</sub> ∈ (a, b)* and  $0 \leq \lambda \leq 1$ ,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \tag{22}$$

**Corollary** A function *f* is concave if  $-f$  is convex.

The function in calculation of PSNR<sub>av</sub> (20) is  $f(x) = \log(1/x)$ . Since the function  $\log(x)$  is concave and  $\log(1/x) = -\log(x)$  we can assume that our function is convex.

**Jensen’s Inequality** If *f* is a convex function and *X* is a random variable, then the expectation operator satisfies

$$E\{f(X)\} \geq f\{E(X)\} \quad (23)$$

In (19), the reference software performs average using logarithmic values, which does not make sense as it leads to systematic error – Jensen’s inequality is the reason for it. In (20), the average is performed using a linear value  $MSE_{av}$ , which is more logical. Also, with this variation, we have one value which groups luminance and chrominance components of the sequence. We can observe the difference in the values using (19) and (20) in Table I where  $PSNR_{av}$  (20) is applied only to the luminance component for fair comparison. As expected from Jensen’s inequality, the reference software PSNR values are optimistical, they are always higher than the true values. All work in this paper is thus based on the modified PSNR metric given by (20).

TABLE I. COMPARISON OF PSNR USING (19) &amp; (20)

Video Sequence	JM Metric [dB] (19)	Proposed Metric [dB] (20)
<i>Akiyo</i>	38.8579	38.8533
<i>Foreman</i>	35.6605	35.5505
<i>Salesman</i>	34.4275	34.3669
<i>Stefan</i>	36.4250	36.3576

The PSNR quality metric has also been widely criticized in recent years for not correlating well with perceived quality measurement [18, 19, 20]. Therefore, a distortion measure that is based on human perception is more appropriate for picture quality estimation. The SSIM index, proposed in [21], is a measure of deviations in luminance (estimated by the mean), contrast (estimated by the standard deviation) and structure (estimated by the normalized covariance) between the two signals  $\mathbf{x}$  and  $\mathbf{y}$ . Luminous, contrastive and structural degradations are represented by the following, respectively:

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{y}}}{\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2}; c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2}; s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}}}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \quad (24)$$

The SSIM index is essentially a product of these three distortions given by

$$SSIM(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2)} \quad (25)$$

where  $C_1=(K_1L)^2$  and  $C_2=(K_2L)^2$  are added constants to ensure the stability of the system.  $L$  is typically set to 255 for gray scale images with 8 bits per pixel. As in [21], we set  $K_1$  and  $K_2$  to 0.01 and 0.03, respectively. We used the average SSIM value (MSSIM) over the desired video frame as the final quality measure. The maximum value of MSSIM is 1, and a value closer to 1 indicates that the concealed frame quality is closer to the quality of the error-free frame. MSSIM has been shown to correlate well with the mean opinion score obtained from subjective quality assessments [21, 22]. In this work, we compute MSSIM for each Intra frame of the video sequence and calculate an average MSSIM as a quality index for the test video sequence.

#### D. Performance Evaluation of Mean and Mode Methods

We demonstrate that the **Mode** method usually gives an improved performance compared to the **Mean** method for finding the dominant edge direction. Intuitively, the **Mode** method should give an accurate edge direction since we are interested in an edge direction having the maximum gradient magnitude. The **Mean** method averages all the obtained directions with their corresponding magnitudes and tends to favor even those angles which occur only once with a high gradient magnitude. For example, consider the example gradient directions and magnitudes shown in Figure 11, obtained on the second outer boundary of the lost MB, after using the Sobel masks. The **Mean** method gives the dominant edge direction as  $39.68^\circ$ , which after quantization results in  $45^\circ$ . However, the **Mode** method gives the dominant edge direction as  $67.5^\circ$ , which is accurate here.

$\theta$	$22.5^\circ$	$67.5^\circ$	$0^\circ$	$22.5^\circ$	$67.5^\circ$	$0^\circ$	$67.5^\circ$
$ G $	100	244	164	40	139	100	80

Figure 11. Example gradient directions and magnitudes

The test model used for simulation is JM 17 [9]. The data set for simulation consisted of seven standard test sequences: *Carphone*, *Foreman1*, *Mother-Daughter*, *News*, *Salesman*, *Foreman2*, and *Hall*. The first five

sequences are in QCIF (176×144) format, and the last two are in CIF (352×288) format. These sequences were selected since they have good edge characteristics, and thus are useful to evaluate the performance of the *Mean* and *Mode* methods. All the sequences are encoded as Intra-only frames at 128 kbps, 25 fps and using dispersed FMO, to utilize the maximum information of all four neighbors for each lost MB. To simulate the channel loss, we dropped the macroblocks at a loss rate of 20% [12].

Fig. 12 shows the visual analysis for the *Hall* Intra frame, which illustrates that the *Mode* method gives more accurate angles compared to the *Mean* method and hence better concealment performance. The objective performance of the seven test sequences of our data set with PSNR as the metric is listed in Table II. Average PSNR was computed for each sequence using the *Mean* and *Mode* method relative to the error-free decoded sequence, and the difference was obtained as

$$\Delta\text{PSNR} = \text{PSNR}_{\text{Mode}} - \text{PSNR}_{\text{Mean}} \quad (26)$$

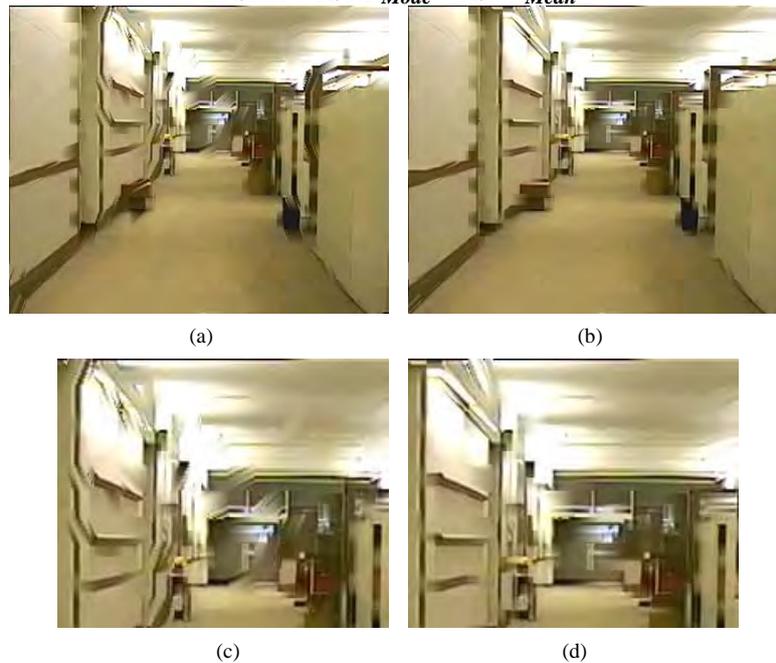


Figure 12. *Hall* CIF Intra frame (a) Concealment with *Mean* method (PSNR = 31.215 dB, MSSIM = 0.7365), (b) Concealment with *Mode* method (PSNR = 32.887 dB, MSSIM = 0.8626), (c) & (d) zoomed portions of (a) and (b)

TABLE II. SIMULATION RESULTS: PSNR & MSSIM

Test Sequence	# Frames	$\Delta\text{PSNR}$ (dB)				$\Delta\text{MSSIM}$			
		Avg.	$\sigma$	Max.	Min.	Avg.	$\sigma$	Max.	Min.
<i>Carphone</i>	200	0.7752	0.3918	1.9992	-0.1425	0.1241	0.0108	0.1519	0.1001
<i>Foreman1</i>	100	0.9743	0.3543	1.9894	-0.0780	0.1101	0.0088	0.1306	0.0896
<i>Mother</i>	200	0.6201	0.3884	1.8664	-0.1508	0.1175	0.0085	0.1392	0.0921
<i>News</i>	200	0.3365	0.1551	0.7259	-0.1118	0.1365	0.0080	0.1580	0.1143
<i>Salesman</i>	200	0.4029	0.1977	0.8596	-0.1960	0.1243	0.0074	0.1450	0.1033
<i>Foreman2</i>	200	0.7728	0.3448	1.5918	-0.1044	0.1129	0.0052	0.1286	0.0991
<i>Hall</i>	300	0.7439	0.1406	1.1224	0.3545	0.1287	0.0038	0.1402	0.1174

Table II indicates the average PSNR difference, the corresponding standard deviations, and the maximum and minimum PSNR difference between the two methods. We can observe that the *Mode* method yields better PSNR values for most of the frames. The maximum improvement obtained with the *Mode* method is 1.9992 dB. Also, for the *Hall* sequence which has the most amount of straight edges, the *Mode* method outperforms the *Mean* method for all the frames. In a few cases where detailed texture is present in the frames (eg. *Salesman* video sequence), the *Mode* method shows a poor performance compared to the *Mean* method since in such cases, very few prominent edges are found. A similar analysis is presented with the quality index and the values are listed in Table II where

$$\Delta\text{MSSIM} = \text{MSSIM}_{\text{Mode}} - \text{MSSIM}_{\text{Mean}} \quad (27)$$

The quality index with the *Mode* method is better for all the test sequences under consideration, with a maximum quality improvement of 0.1519. For the total 1400 Intra frames of our data set, we obtain histograms of  $\Delta$ PSNR and  $\Delta$ MSSIM, as shown in Fig. 13. Overall, the *Mode* method gives better error concealment performance in terms of PSNR and MSSIM relative to the *Mean* method.

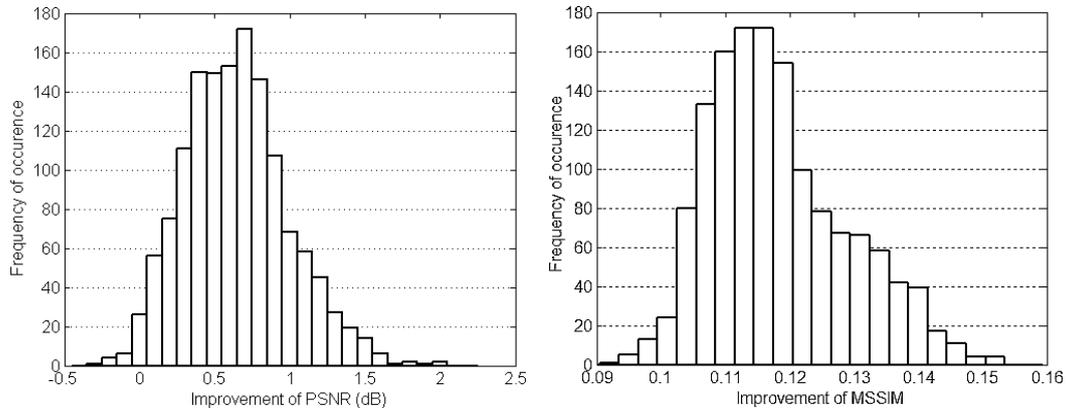


Figure 13. Histogram of  $\Delta$ PSNR (left) and Histogram of  $\Delta$ MSSIM (right)

## V. CONCLUSION

A spatial error concealment scheme is required at the H.264 decoder to compensate for the degradation from lossy channels and improve the visual quality of the decoded picture. Using dispersed FMO during encoding of the video sequence, the error concealment performance at the decoder can be greatly improved since more neighboring information is available as demonstrated mathematically. The bilinear interpolation method of the test model suffers from blurring artifacts in smooth regions, and this drawback is overcome by using the directional interpolation scheme, which preserves the local edge integrity of the damaged regions. To determine the dominant edge direction for a damaged macroblock, the *Mode* method should be used since it usually outperforms the *Mean* method by enhancing the visual perception as well as improving the objective quality metrics of the reconstructed picture.

## REFERENCES

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 13, no. 7, pp. 560-576, July 2003
- [2] Y. Wang, Q. Zhu, and L. Shaw, "Coding and cell-loss recovery in DCT based packet video," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 3, no. 3, pp. 248-258, Jun. 1993
- [3] S. Hemami, and T. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 1023-1027, Jul. 1995
- [4] J. W. Suh, and Y. S. Ho, "Error concealment based on directional interpolation," *IEEE Trans. Consum. Electron.*, vol. 43, no. 3, pp. 295-302, Aug. 1997
- [5] W. Zhu, Y. Wang, and Q. F. Zhu, "Second-order derivative-based smoothness measure for error concealment," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 8, no. 6, pp. 713-718, Oct. 1998
- [6] Z. Alkachouh, and M. G. Bellanger, "Fast DCT-based spatial domain interpolation of blocks in images," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 729-732, Apr. 2000
- [7] G. S. Yu, M. M. K. Liu, and M. W. Marcellin, "POCS-based error concealment for packet video using multiframe overlap information," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 8, no. 4, pp. 422-434, Aug. 1998
- [8] Y. Xu, and Y. Zhou, "H.264 video communication based refined error concealment schemes," *IEEE Trans. Consum. Electron.*, vol. 50, no. 4, pp. 1135-1141, Nov. 2004
- [9] H.264/AVC Software Coordination, "Joint Model Software," *Joint Video Team*, ver. 17 [Online] Available: <http://iphome.hhi.de/suehring/tml/>
- [10] P. Baccichet, A. Chimienti, E. Quacchio, E. Magli, and G. Olmo, "Performance evaluation of FMO to improve the error resilience of H.264/AVC," *ST Journal of Research (Multimedia Streaming Technologies)*, vol. 3, no. 2, pp. 46-60, Dec 2006
- [11] Y. Dhondt, P. Lambert, S. Notebaert, and R. Van de Walle, "Flexible macroblock ordering as a content adaptation tool in H.264/AVC," *Multimedia Systems and Applications VIII, Proceedings of the SPIE*, vol. 6015, Oct. 2005
- [12] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing," ITU-T SG16/Q6, Doc. VCEG-N79r1, Sep. 2001
- [13] M. T. Sun, and A. R. Reibman, *Compressed Video over Networks*, New York: Marcel Dekker Inc., 2000
- [14] O. Nemethova, A. Al-Moghrabi, and M. Rupp, "Flexible error concealment for H.264 based on directional interpolation," *IEEE Intl. Conf. Wireless Networks, Communications and Mobile Computing*, vol. 2, pp. 1255-1260, Jun. 2005
- [15] R.C. Gonzalez, and R. E. Woods, *Digital Image Processing*, Addison-Wesley, USA, 1992
- [16] T.P. Chen, and T. Chen, "Second-generation error concealment for video transport over error-prone channels," *Proc. of IEEE Intl. Conf. on Image Processing*, vol. 1, pp. 25-28, Sept. 2002
- [17] Y. Ma, and A. Cai, "A new spatial interpolation method for error concealment," *Proc. of IEEE 6th Int. Symp. on Multimedia Software Engineering*, pp. 65-69, Dec. 2004

- [18] S. Chapaneri, and J. Rodriguez, "Low complexity error concealment scheme for Intra frames in H.264/AVC", *Proc. of IEEE Intl. Conf. Image Processing*, pp. 925-928, Nov. 2009
- [19] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it?", *IEEE Signal Processing Magazine*, vol. 26, pp.98-117, Jan 2009
- [20] A. Hore, and D. Ziou, "Image quality metrics: PSNR vs. SSIM", *IEEE Intl. Conf. Pattern Recognition*, pp. 2366-2369, Aug 2010
- [21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 1-14, Jan. 2004
- [22] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, special issue on "Objective Video Quality Metrics", vol. 19, no. 2, pp. 121-132, Feb. 2004