

# Reference Frame Selection for Loss-Resilient Depth Map Coding in Multiview Video Conferencing

Bruno Macchiavello<sup>a</sup>, Camilo Dorea<sup>a</sup>, Edson M. Hung<sup>a</sup>, Gene Cheung<sup>b</sup>, Wai-tian Tan<sup>c</sup>

<sup>a</sup>Universidade de Brasilia;  
<sup>b</sup>National Institute of Informatics;  
<sup>c</sup>Hewlett-Packard Laboratories

## ABSTRACT

Multiview video in “texture-plus-depth” format enables decoder to synthesize freely chosen intermediate views for enhanced visual experience. Nevertheless, transmission of multiple texture and depth maps over bandwidth-constrained and loss-prone networks is challenging, especially for conferencing applications with stringent deadlines. In this paper, we examine the problem of loss-resilient coding of depth maps by exploiting two observations. First, different depth macroblocks have significantly different error sensitivities with respect to the reconstructed images. Second, unlike texture, the relative overhead of using reference pictures with large prediction distance is low for depth maps. This motivates our approach of assigning a weight to represent the varying error sensitivity of each macroblock and using such weights to guide selection of reference frames. Results show that (1) errors in depth maps in sequence with high motion yields significant drop in quality in reconstructed images, and (2) that the proposed scheme can efficiently maintain the quality of reconstructed images even at relatively high packet loss rates of 3-5%.

**Keywords:** Multiview video, depth-image-based rendering, loss resiliency

## 1. INTRODUCTION

Multiview video extends traditional single-view video by providing a choice of view captured by multiple synchronous cameras that are spaced closely apart. In addition to captured images (texture), if depth maps of per-pixel distances between the camera and captured physical objects are also available\*, then the transmission of both texture and depth maps of multiple viewpoints—a format known as “texture-plus-depth”—can enable a decoder to synthesize novel intermediate views beyond those captured via depth-image-based rendering (DIBR) techniques such as 3D warping.<sup>1</sup> The ability to generate any desired viewpoint—commonly called *free viewpoint*<sup>2</sup>—smooths transition between neighboring captured views and improves a viewer’s perception of depth via motion parallax.<sup>3</sup>

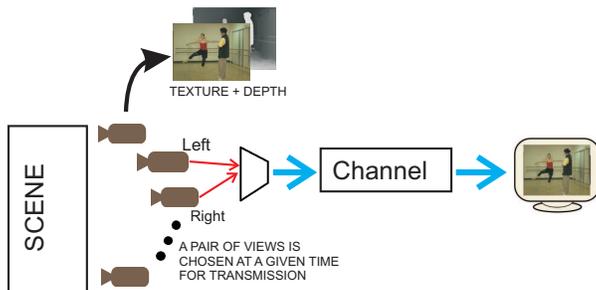


Figure 1. A bandwidth-efficient multiview video conferencing system dynamically selects two views for transmission.

In an interactive multiview video system, it is often not necessary to transmit all captured views. In contrast, by selectively transmitting only two neighboring views, it is possible to support stereoscopic viewing, as depicted

\*Depth maps can be estimated from texture maps using stereo matching algorithms, or captured explicitly using time-of-flight cameras.

in Fig. 1. Free viewpoint can also be supported if depth maps are additionally transmitted. Nevertheless, reliable transmission of two textures and depth maps over bandwidth-constrained and loss-prone networks is challenging, especially for conferencing applications with stringent video playback deadlines.

The subject of error-resilient streaming of video (texture) in a low-delay environment has been extensively studied. In this paper, we focus on resilient streaming of depth maps. Our basic observation is that different depth macroblocks (MB) with different contents have different error sensitivity to the final synthesized images, which calls for unequal protection. We apply reference picture selection (RPS) techniques to provide the necessary unequal protection since it is an effective technique supported by H.264 standards. Specifically, a coded depth macroblock has best chance of being decodable if it references an early depth frame that is known to be error free, though at the expense of coding efficiency. Conversely, the highest coding efficiency is achieved by using the immediate preceding frame as reference, though likely with more distortion due to packet losses. To support optimized selection of reference for a depth macroblock, we develop a method of assigning weights or relative importance to each depth MB, and also a method to model distortion in a candidate reference block taking into account loss and motion compensation. As we will show in the results section, using RPS for depth maps is especially attractive given the relatively low overhead compared to using RPS with texture.

The outline of the paper is as follows. We first discuss related works in Section 2. Then, we discuss how to determine the weight or importance of different depth MBs in Section 3. We derive recursive equations to model depth MB distortion under motion compensation, and formulate the optimization problem of choosing the right reference frames for each MB in Section 4. An efficient algorithm for the optimization is presented in Section 5. Preliminary results and conclusions are given in Sections 6 and 7, respectively.

## 2. RELATED WORK

Due to its ability to synthesize novel views at decoder, multiview video in “texture-plus-depth” format has been a popular research topic.<sup>4,5</sup> Most works, however, focus on compression performance of multiview frames, exploiting correlations in frames across time and view. In this paper, we address the orthogonal and practical problem of loss resilient coding of multiview video (albeit depth maps only). To the best of the authors’ knowledge, this is the first rigorous study on loss resiliency of depth video streaming.

Compression schemes tailored specifically for depth maps have also been studied recently.<sup>6,7</sup> These studies<sup>6,7</sup> observed that depth maps are encoded solely for the purpose of view synthesis and designed specialized coding optimization accordingly. In one work,<sup>6</sup> coding modes in H.264 are selected based on individual MBs’ impact on synthesized view distortion. In the *transform domain sparsification* work,<sup>7</sup> the values of less important depth pixels in a MB are manipulated so that the MB’s transform domain representation becomes more sparse (for coding gain). We leverage the recent results by Cheung et al.<sup>7</sup> to derive weights of importance for individual MBs in a depth frame.

Using the flexibility of RFS in H.264 to control error propagation in motion compensated frames have been previously studied for *single-view* video at the frame level.<sup>8</sup> Our contribution in this paper is to optimize RFS for depth maps at the MB level according to importance of each MB.

## 3. DETERMINING MB IMPORTANCE

We now derive weights or importance of pixels in a depth map<sup>†</sup>. The weight of a MB is computed as a sum of its constituent pixels. A pixel  $I_l(m, n)$  in the left texture map can be mapped to a shifted pixel  $I_r(m, n - D_l(m, n) * \gamma)$  in the right texture map, where  $D_l(m, n)$  is the disparity value in the left depth map corresponding to left texture pixel  $I_l(m, n)$ , and  $\gamma$  is the camera-shift scaling factor for this camera setup. We express a synthesized view’s distortion sensitivity to disparity  $D_l(m, n)$  by the error function  $E_l(k; m, n)$ , which is the difference in texture pixel values between left pixel  $I_l(m, n)$ , and incorrectly mapped right pixel  $I_r(m, n - (D_l(m, n) + e) * \gamma)$  due to disparity error  $e$ .

$$E_l(e; m, n) = |I_l(m, n) - I_r(m, n - (D_l(m, n) + e) * \gamma)| \quad (1)$$

---

<sup>†</sup>Technically, we are encoding *disparity* rather than *depth* values for DIBR at decoder. Since it is clear from context which quantity is intended, we will use these terms interchangeably in this paper.



Figure 2. Error and weight functions for one pixel are shown in (a) for image in (b). The resulting weights for entire image are shown in (c).

Error function  $E_r(e; m, n)$  for the right depth map is similarly defined. The error function  $E_r(e; m, n)$  for view 6 of the multiview image sequence **Teddy** is shown in blue in Fig. 2(a). We see that as the depth value deviates from the true disparity value  $D_r(m, n)$  (red circle), the error increases generally.

We now fit a per-pixel quadratic weight function  $g_i(s_i)$  to the error function, as done in Cheung et al.:<sup>7</sup>

$$g_i(s_i) = (1/2)a_i s_i^2 + b_i s_i + c_i \quad (2)$$

where  $s_i$  is the disparity value corresponding to pixel location  $i$ , and  $a_i$ ,  $b_i$  and  $c_i$  are the quadratic function parameters determined as follows. Given threshold  $\rho$ , we first seek the nearest disparity  $D_l(m, n) - e$  value below ground truth  $D_l(m, n)$  that results in error  $E_l(-e; m, n)$  exceeding  $\rho + E_l(0; m, n)$ . Using only two data points at  $D_l(m, n) - e$  and  $D_l(m, n)$ , and assuming  $g_i(s_i)$  has minimum at ground truth depth value  $D_l(m, n)$ , we can construct one quadratic function. Similar procedure is applied to construct another weight function using two data points at  $D_l(m, n) + e$  and  $D_l(m, n)$  instead. The sharper of the two constructed functions (larger  $a$ )—reflecting the larger of the two sensitivities in the plus and minus error directions—is the chosen weight function for this pixel.

Fig. 2(a) shows the two quadratic functions, the narrower of which is chosen as the weight function. The weight of a pixel is simply the curvature (parameter  $a$ ) of the chosen weight function. In Fig. 2(b) and (c), the original image 6 of **Teddy**<sup>9</sup> and the per-pixel curvatures of the weight functions of the corresponding depth map are shown. We can clearly see that larger curvatures (larger weights in white) occur at object boundaries, agreeing with our intuition that a synthesized view is more sensitive to depth pixels at object boundaries.

## 4. PROBLEM FORMULATION

The goal of our optimization is, for each MB  $i$  in a current depth frame  $F_t$ , select a prediction block in a previous depth frame  $F_{\tau_{t,i}}$ ,  $\tau_{t,i} < t$ , for motion compensation (MC), so that the expected distortion of an interpolated view, synthesized via DIBR at decoder using the decoded depth frame  $F_t$ , is minimized, subject to a total transmission rate constraint. We first derive a recursive formula to estimate depth distortion of a block following similar derivation in a previous work.<sup>10</sup>

### 4.1 A Recursive Distortion Model

In a single view, let  $d_{t,i}(\tau_{t,i}, c_{t,i})$  be the distortion of MB  $i$  of frame  $F_t$ , given it is motion compensated using a block identified by motion vector (MV)  $c_{t,i}$  inside frame  $F_{\tau_{t,i}}$ ,  $\tau_{t,i} < t$ . Let  $p$  be the probability that MB  $i$  of  $F_t$  is correctly received. We can now write  $d_{t,i}(\tau_{t,i}, c_{t,i})$  in terms of  $d_{t,i}^R(\tau_{t,i}, c_{t,i})$  and  $d_{t,i}^L$ , the distortion of MB  $i$  of  $F_t$  if coded MB  $i$  is correctly received and lost, respectively:

$$d_{t,i}(\tau_{t,i}, c_{t,i}) = p d_{t,i}^R(\tau_{t,i}, c_{t,i}) + (1 - p) d_{t,i}^L \quad (3)$$

If MB  $i$  of  $F_t$  is correctly received, then the corresponding distortion  $d_{t,i}^R(\tau_{t,i}, c_{t,i})$  depends on the quality of the reference block  $b$  identified by MV  $c_{t,i}$  in frame  $F_{\tau_{t,i}}$ . In general, reference block  $b$  can be interpolated using several neighboring MBs  $k \in c_{t,i}$  at integer pixel coordinates, due to sub-pixel accuracy in H.264's MC. We can hence write  $d_{t,i}^R(\tau_{t,i}, c_{t,i})$  as a weighted sum of distortions of these MBs:

$$d_{t,i}^R(\tau_{t,i}, c_{t,i}) = \gamma \sum_{k \in c_{t,i}} \alpha_k d_{\tau_{t,i},k}(\tau_{\tau_{t,i},k}, c_{\tau_{t,i},k}) \quad (4)$$

where  $\alpha_k$ 's are the weights, and  $\gamma < 1$  is the attenuation factor that reflects the dissipating effect of distortion in an earlier frame over a sequence of motion compensated frames.

If MB  $i$  of  $F_t$  is lost, then we assume a simple block copy procedure is used for loss concealment, where MB  $i$  of previous frame  $F_{t-1}$  is used in its place. In this case, the distortion  $d_{t,i}^L$  will be the previous MB's distortion  $d_{t-1,i}(\tau_{t-1,i}, c_{t-1,i})$  plus block difference between MB  $i$  of frame  $F_{t-1}$  and MB  $i$  of frame  $F_t$ ,  $\delta_{t-1,i}$ :

$$d_{t,i}^L = d_{t-1,i}(\tau_{t-1,i}, c_{t-1,i}) + \delta_{t-1,i} \quad (5)$$

The recursive definitions above computes channel-induced distortions given inter-frame dependencies established during MC of previous frames. To provide a base case for the recursion, we assume there exists an *acknowledged MB* (ACKed MB) in every dependency chain, one where the receiver has indicated it has been decoded without error, so that its channel-induced distortion is 0. Correctly received MBs from a periodically inserted I-frame, for example, can serve as ACKed MBs.

The extension of these recursive definitions to a stereo system are trivial. The two selected views of our video conferencing scenario can be encoded as described in Fig. 3, where the left view is coded using only temporal predictions (similar to standard single-view video), and the right view can be predicted from its previous frames or from frames of the adjacent view at the same instant. Therefore, the recursive definition for the right view includes an extra reference frame (from the adjacent view) from which to choose.

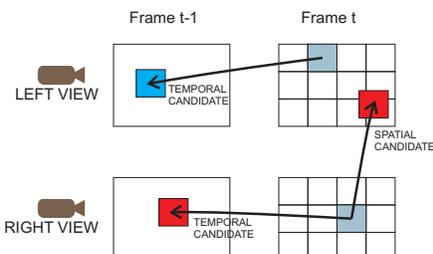


Figure 3. A coding structure for a stereo system

## 4.2 Problem Definition

We are now ready to formally define our optimization problem. Our objective is to find a set of MVs  $c_{t,i}$ 's of previous frames  $F_{\tau_{t,i}}$ 's for all MBs  $i$ 's in current  $F_t$  that minimizes the expected weighted distortion:

$$\min_{\{\tau_{t,i}\}, \{c_{t,i}\}} \sum_i w_{t,i} d_{t,i}(\tau_{t,i}, c_{t,i}) \quad (6)$$

where  $w_{t,i}$  is the weight (relative importance) of depth MB  $i$  of  $F_t$ , as computed in Section 3. In other words, instead of raw depth MB distortions, we scale each one by its weight  $w_{t,i}$  to reflect its degree of impact in synthesized view distortion.

Each chosen MV  $c_{t,i}$  in previous frame  $F_{\tau_{t,i}}$  will lead to a coding rate overhead  $r_{t,i}(\tau_{t,i}, c_{t,i})$  to encode block  $i$  of  $F_t$ . The optimization will hence be subject to the following total rate constraint:

$$\sum_i r_{t,i}(\tau_{t,i}, c_{t,i}) \leq R_t \quad (7)$$

where  $R_t$  is the bit budget for encoding of frame  $F_t$ .

## 5. ALGORITHM

Instead of solving the constrained optimization problem (6) subject to rate constraint (7), we solve the Lagrangian unconstrained equivalent:

$$\min_{\{\tau_{t,i}\},\{c_{t,i}\}} \sum_i [w_{t,i} d_{t,i}(\tau_{t,i}, c_{t,i}) + \lambda r_{t,i}(\tau_{t,i}, c_{t,i})] \quad (8)$$

where  $\lambda$  is the Lagrangian multiplier that trades off distortion with coding rate.

To solve (8) efficiently, we first note that the individual Lagrangian cost for each MB  $i$  in (8) can be minimized separately without loss of global optimality. Second, to further reduce complexity, for each MB  $i$ , we optimize reference frame  $\tau_{t,i}$  and MV  $c_{t,i}$  separately as follows. Given reference frame  $\tau_{t,i}$ , the MV  $c_{t,i}$  pointing to the most similar block in  $F_{r_{t,i}}$ —the one leading to the smallest sum of absolute difference (SAD)—is first searched, as done in normal motion estimation operation in H.264. Then among all the MVs  $c_{t,i}$ 's for all reference frames back to the previous ACKed frame, we select the combo  $(\tau_{t,i}, c_{t,i})$  that leads to the smallest Lagrangian cost (8) for MB  $i$ .

## 6. RESULTS

We first motivate why differential application of reference picture selection according to macroblock importance is a preferred method for robust transport of depth maps. Fig. 4 shows the PSNR of reconstructed pictures

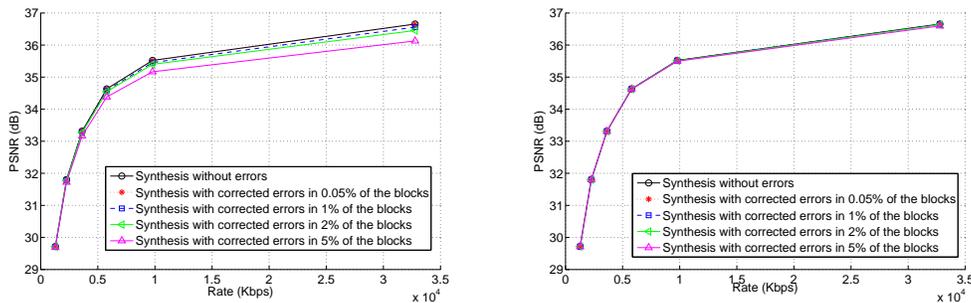


Figure 4. PSNR for reconstructing pictures with only errors in depth macroblocks co-located with object borders (left) and non-border depth macroblocks (right) for Ballet sequence.

when losses only occur in depth macroblocks along the borders of objects (left), and in the interior of the objects (right) for Ballet sequence. The key observation is that the reconstructed pictures are insensitive to losses in the “interior” depth macroblocks, and any extra bits for protection should be spent on “edge” macroblocks. Differential protection of depth macroblocks can be realized by many methods. We are interested in reference picture selection due to its effectiveness and its rate efficiency when applied to depth maps. While we observe similar behavior for many sequences, we illustrate the efficiency of using RPS for depth maps using the rate-distortion curve for view 40 of Rena in Fig. 5 for texture (left) and depth (right). It is obtained using H.264 encoder from Intel’s Unified Media Classes software. The key observation is that the percentage increase in rate caused by using larger prediction distance is significantly smaller for coding depth maps than texture. This makes it practical to apply reference picture pro-actively rather than only in response to losses.

Our proposed scheme in this paper applies reference picture selection both reactively and proactively. Specifically, we proactively choose longer prediction distance for select macroblocks to improve their resilience, while reacting to loss feedback by avoiding prediction from pictures known to be impaired by losses. We call our scheme “Modified H.264”. It is implemented using JM reference software v18.0.<sup>11</sup> For results in this paper, only P16×16 mode is used. For comparison, we also implement a scheme based on intra-refresh where an encoder can choose between constrained intra blocks or P16×16 modes. A constrained intra block does not use inter blocks for Intra prediction and are inserted to improve error resilience. We call this scheme “Conventional H.264+MBIntra”.

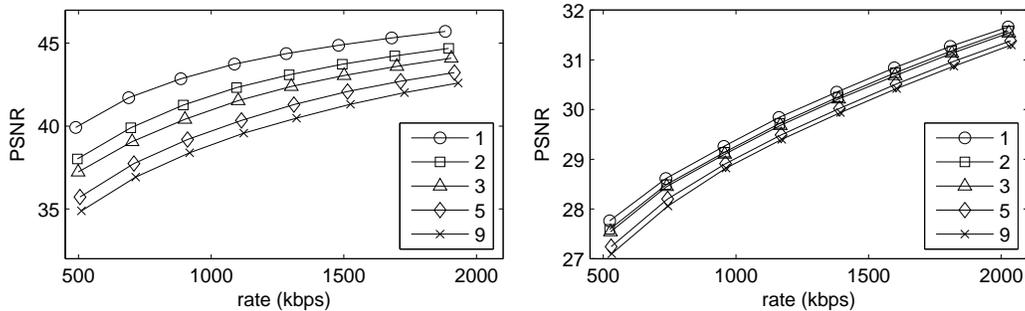


Figure 5. Rate-distortion tradeoff for texture (left) and depth map (right) at different prediction distance for view 40 of Rena sequence.

During encoding of depth maps each frame was divided into three packets. For texture encoding, fifteen packets were used due to higher associated bit rates. Packet loss is only simulated for depth maps. In our experiments MB partitions and sub-partitions are not allowed, we use only P-type  $16 \times 16$  MBs. Simulations include losses of 3% and 5% of the MBs. Given these percentages, the probability that a MB is correctly received in equation 3 is calculated and introduced as an encoding parameter. In order to provide meaningful comparisons, the same packets are lost in all tests, independently of the coder used.

Encoding configurations that are applied for both depth maps and texture include 64-pixel search window, CABAC entropy encoder and *IPPP*... encoding mode. For view interpolation we used the MPEG View Synthesis Reference Software (VSRS v3.5).<sup>12</sup>

We perform view synthesis of view 2 of the Kendo sequence ( $1024 \times 768$  pixels)<sup>13</sup> using views 1 and 3, and compare the resulting PSNR against the actual view 2. The results are shown in Fig. 6 for QPs 22 and 27. The simulation is for network with round-trip-delay of 4 frames or 133 ms. Texture is coded with conventional H.264, and the depth maps are coded using either “Conventional H.264+MBIntra”, our optimized “Modified H.264”, or “Conventional H.264”, which refers to H.264 compression without any application of RPS. We also included as reference the curve that represent the synthesis obtained when no errors occurred. In the first 25 frames of Fig. 6 there are no significant PSNR differences due to the lack of motion for these frames. As motion in the sequence increases, we see that “Conventional H.264” starts to suffer from drift errors and quality degrades significantly. In contrast, applications of RPS to depth map (green and black) are able to recover from losses and avoid drift due to appropriate adaptation to losses. We also see that traditional application Intra blocks improves the performance compare to “Conventional H.264”. In this configuration the intra blocks are inserted randomly, therefore it is possible that particular frame can have a similar (or even better) PSNR than our scheme, however as can be seen our method outperforms the random introduction of constrained intra blocks in most frames. The associated bitrate, considering depth maps of both views, are: (i) for QP 22, 4.9 Mbps for the “Conventional H.264”, 7.4 Mbps for “Conventional H.264+MBIntra” and 7.47 or 7.38 Mbps for “Modified H.264” (depending on the error rate) and (ii) for QP 27, 2.8, 4.3 and 4.3 (or 4.27) Mbps, respectively. Corresponding results for the Akko and Kayo sequence are presented in Fig. 7 for a simulated delay of 8 frames or 266 ms. As before, use of RPS avoids accumulation of drift, and significant transient drops in quality can be efficiently avoided by our method of proactive application of RPS. Note that the appropriate  $\lambda$  for equation 8 in all simulations was selected empirically.

Subjective improvements of our method can be appreciated in the detail crop of Kendo in Fig. 8 and Fig. 9. Due to better protection of edge macroblocks, our proposed method are less prone to objectionable artifacts. Similar results are observed for Akko and Kayo in Fig. 10 and 11. Finally, in Fig. 12 we show the RD curves for quantization parameter equal to 22, 27, 32 and 37 using Bjontegaard<sup>14</sup> method. A positive PSNR difference ( $\Delta$  PSNR) means that the proposed method outperforms the conventional H.264 plus the random insertion of constrained intra blocks. We see that our scheme consistently out-performs intra-refresh. Even though the difference in average PSNR is not large, usually less than 0.3 dB, we have seen earlier than the difference in

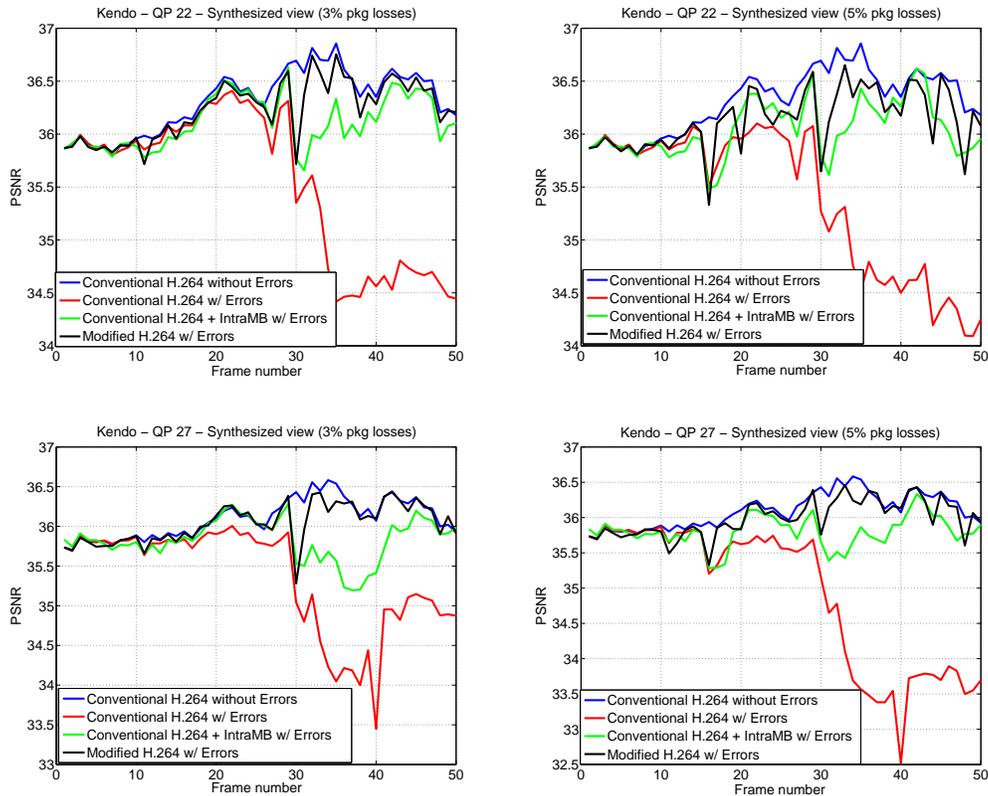


Figure 6. PSNR results of synthesized view 2 of Kendo sequence, with packet losses in the Depth Maps: (a) QP 22 and 3% packet were lost, (b) QP 22 and 5% packet were lost, (c) QP 27 and 3% packet were lost, (d) QP 27 and 5% packet were lost

visual appearance can be significant, as can be seen in the frame examples of Figs. 8 to 11. The rate in the graphs is the rate for the depth maps of both views.

## 7. CONCLUSION

In this paper we have: (i) formulated a recursive distortion model to estimate the impact of a lossy channel during depth maps encoding, (ii) given this model, defined an optimization problem with a rate constraint for coding MBs of a depth map; and (iii) presented an algorithm to solve the constrained optimization problem. This algorithm was implemented in H.264/AVC, and the results show that it is superior to intra-refresh in both achieving a superior PSNR and maintaining less quality variation due to losses. More importantly, the subjective quality also increases, since during encoding the most important MBs are protected. This is achieved by determining the relative MB importance during the encoding process and representing this importance as a weight for the block distortion.

## REFERENCES

- [1] Mark, W., McMillan, L., and Bishop, G., "Post-rendering 3D warping," in [*Symposium on Interactive 3D Graphics*], (April 1997).
- [2] Kubota, A., Smolic, A., Magnor, M., Tanimoto, M., Chen, T., and Zhang, C., "Multi-view imaging and 3DTV," in [*IEEE Signal Processing Magazine*], **24**, no.6 (November 2007).
- [3] Zhang, C., Yin, Z., and Florencio, D., "Improving depth perception with motion parallax and its application in teleconferencing," in [*IEEE International Workshop on Multimedia Signal Processing*], (October 2009).

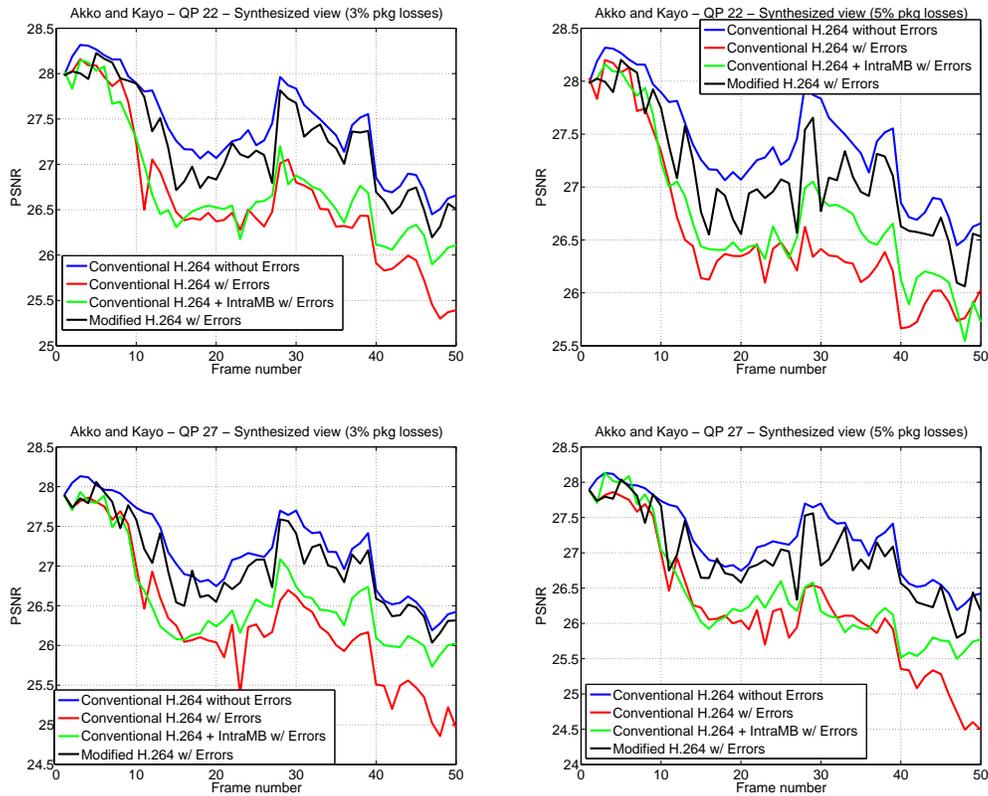


Figure 7. PSNR results of synthesized view 48 of Akko and Kayo sequence, with packet losses in the Depth Maps: (a) QP 22 and 3% packet were lost, (b) QP 22 and 5% packet were lost, (c) QP 27 and 3% packet were lost, (d) QP 27 and 5% packet were lost

- [4] Oh, H. and Ho, Y.-S., “H.264-based depth map sequence coding using motion information of corresponding texture video,” in [*The Pacific-Rim Symposium on Image and Video Technology*], (December 2006).
- [5] Daribo, I., Tillier, C., and Pesquet-Popescu, B., “Motion vector sharing and bit-rate allocation for 3D video-plus-depth coding,” in [*EURASIP: Special Issue on 3DTV in Journal on Advances in Signal Processing*], **2009** (2009) (January 2009).
- [6] Kim, W.-S., Ortega, A., Lai, P., Tian, D., and Gomila, C., “Depth map distortion analysis for view rendering and depth coding,” in [*IEEE International Conference on Image Processing*], (November 2009).
- [7] Cheung, G., Ishida, J., Kubota, A., and Ortega, A., “Sparse transform domain representation of depth maps using iterative quadratic programs,” in [*IEEE International Conference on Image Processing*], (September 2011).
- [8] Cheung, G., Tan, W.-T., and Chan, C., “Reference frame optimization for multiple-path video streaming with complexity scaling,” in [*IEEE Transactions on Circuits and Systems for Video Technology*], **17**, no.6, 649–662 (June 2007).
- [9] “2006 stereo datasets.” <http://vision.middlebury.edu/stereo/data/scenes2006/>.
- [10] Zhou, Y., Hou, C., Xiang, W., and Wu, F., “Channel distortion modeling for multi-view video transmission over packet-switched networks,” in [*accepted to IEEE Transactions on Circuits and Systems for Video Technology*], (2011).
- [11] “Jm h.264 reference software v18.0,” in [<http://iphome.hhi.de/suehring/tml/>],
- [12] Tanimoto, M., Fujii, T., Suzuki, K., Fukushima, N., and Mori, Y., “Reference softwares for depth estimation and view synthesis,” in [*ISO/IEC JTC1/SC29/WG11 MPEG2008/M15377*], (April 2008).
- [13] “Nagoya university ftv test sequences,” in [<http://www.tanimoto.nuee.nagoya-u.ac.jp/>],

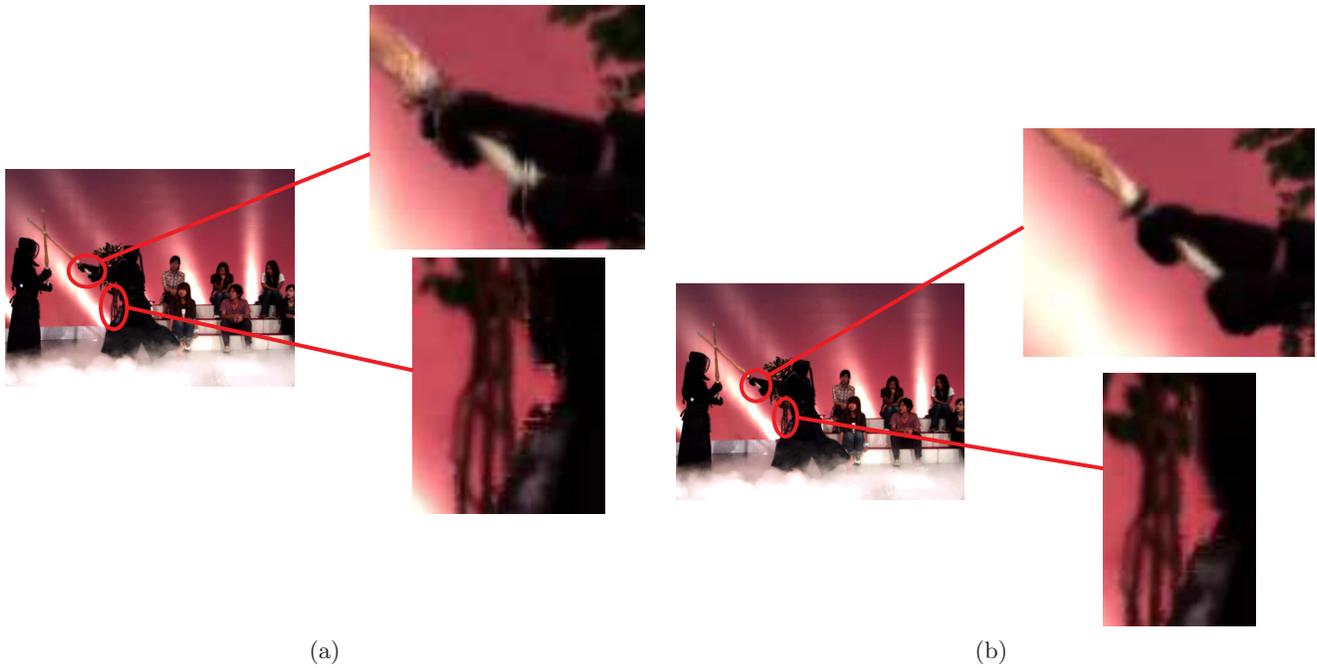


Figure 8. Example of synthesized errors due to depth map losses in frame 30 of Kendo sequence for QP 22 and 5% packet loss (last ACKed frame was frame 29). (a) Conventional H.264 + MB Intra Refresh, (b) proposed encoding mode (Modified H.264)



Figure 9. Example of synthesized errors due to depth map losses in frame 35 of Kendo sequence for QP 27 and 3% packet loss (last ACKed frame was frame 33). (a) Conventional H.264 + MB Intra Refresh, (b) proposed encoding mode (Modified H.264)

[14] Bjontegaard, G., "Calculation of average psnr differences between rd curves," in [*PITU-T SC16/Q6, 13th VCEG Meeting Doc. VCEG-M33*], (2001).

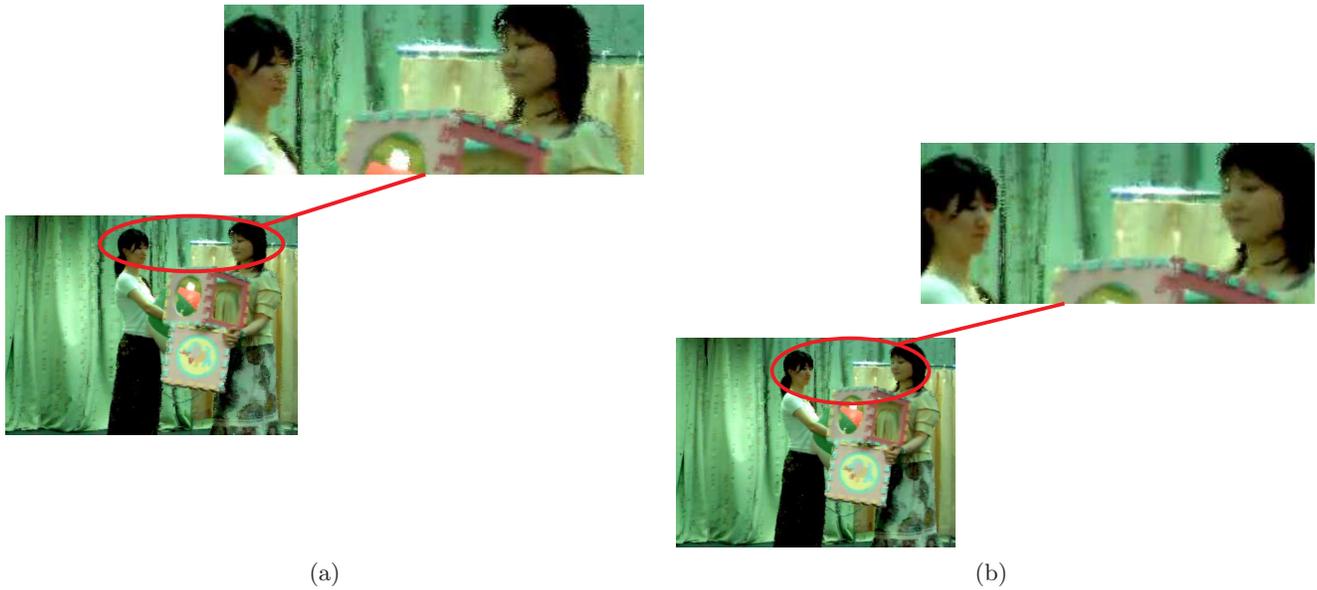


Figure 10. Example of synthesized errors due to depth map losses in frame 19 of Akko and Kayo sequence for QP 22 and 3% packet loss (last ACKed frame was frame 17). (a) Conventional H.264 + MB Intra Refresh, (b) proposed encoding mode



Figure 11. Example of synthesized errors due to depth map losses in frame 7 of Akko and Kayo sequence for QP 22 and 5% packet loss (last ACKed frame was frame 1). (a) Conventional H.264 + MB Intra Refresh, (b) proposed encoding mode

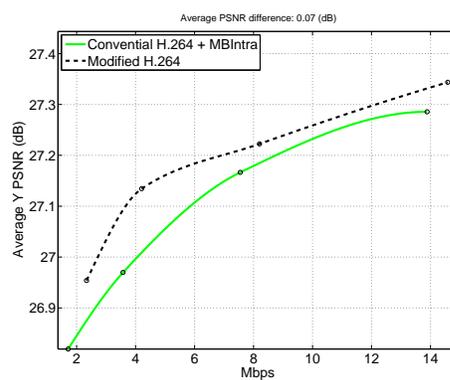
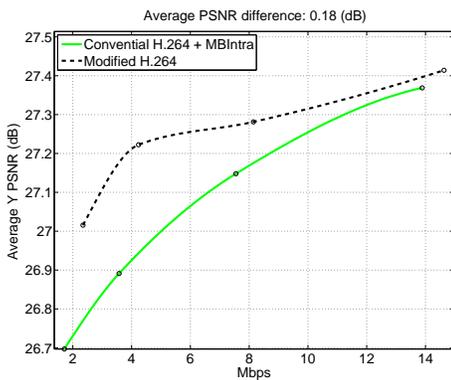
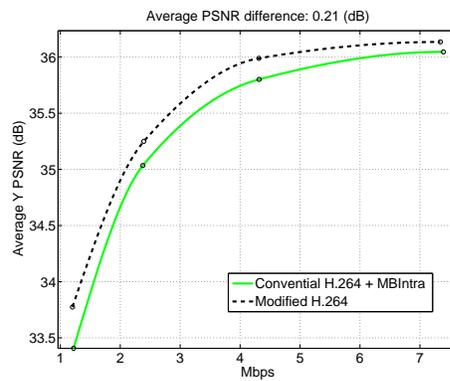
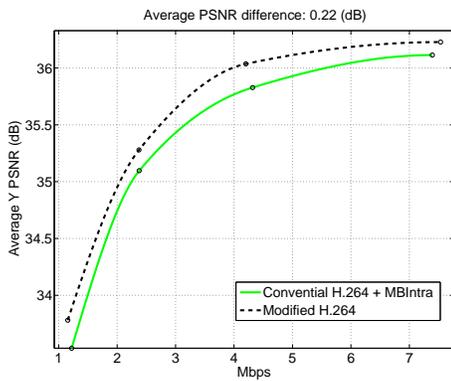


Figure 12. RD curves for QPs 22, 27, 32 and 32. (a) Kendo with 3% packet loss ( $\Delta$  PSNR 0.22), (b) Kendo with 5% packet loss ( $\Delta$  PSNR 0.21), (c) Akko and Kayo with 3% packet loss ( $\Delta$  PSNR 0.18), (d) Akko and Kayo with 5% packet loss ( $\Delta$  PSNR 0.07),