©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-234

VPx Error Resilient Video Coding Using Duplicated Prediction Information

Neeraj Gadgil and Edward J. Delp
Video and Image Processing Laboratory
School of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana, USA.

Abstract

Most real-time video applications typically demand low end-to-end latency and faithful reconstruction of the video sequence. Many popular video coding standards (e.g. VP8, VP9, H.264 and HEVC) aim at achieving high compression efficiencies by exploiting spatial and temporal redundancies. This makes the encoded bitstream vulnerable to errors. Thus, applications especially on mobile phones, tablet PCs and other portable devices that use WiFi or 3G/4G/LTE networks typically suffer from low quality of service typically characterized by frequent delays, jitter, frozen picture, partial/no picture and total loss of connection. Similar scenarios are also often observed while watching live streaming accompanied by service interruptions and a blank screen. Our approach is to investigate error resilient coding control for the VPx encoder to make the bitstream more error resilient for streaming applications under lossy channel conditions. In this paper, we describe an error resilient coding system that uses duplication of frame prediction information. Our "error resilience packet" consists of this prediction information of several frames, that can be used for error concealment in the case of packet loss.

Introduction

There has been a dramatic increase in the volume of video traffic over the Internet over the last decade. With the development of 3G/4G/LTE and WiFi networks, a further increase in the demand for video delivery over these channels is projected. A recent statistical study revealed that, in 2014, 64% of global Internet traffic was used for video delivery and that number is predicted to increase to 80% by the year 2019 [1]. In 2014, wired devices accounted for the majority of IP traffic (54%). Traffic from wireless devices will exceed traffic from wired devices by 2019, with WiFi and mobile devices accounting for 66% of IP traffic [1]. This rapid increase in traffic over error-prone wireless channels and among heterogeneous clients has raised some significant challenges for developing efficient coding techniques for this purpose.

A defining characteristic of a wireless channel is the variation of the channel strength over time and frequency [2]. This can cause packet loss during signal transmission. In real-time applications such as video chat or live streaming, retransmission of lost packets is not feasible. As a result, only a subset of total packets is available at the receiver, which must reconstruct the signal from the available information.

The goal of video coding is typically to represent a source sequence by the lowest data rate (bits/pixel or bits/second) for a given reconstruction quality. Video compression is achieved by removing the redundancies from an original sequence [3, 4]. Statistical correlations within a video frame are used to reduce the spatial redundancy and statistical correlations among the neighboring video frames can be used to reduce the temporal redundancy. In addition, orthogonal transformations such as the discrete cosine transform (DCT) are used to further reduce the redundancy. To ensure the inter-operability between different manufacturers and devices, a series of video coding standards have been developed with the growing requirements of applications [4]. Many well-established video compression standards such as MPEG-2 [5], VP8 [6], VP9 [7], H.264 [8] or HEVC [9] are mainly aimed at achieving better compression efficiencies. Due to the use of spatial-temporal correlations for compression, compressed bitstreams typically become vulnerable to errors. The encoder uses the error free frame as a reference when encoding future frames however the reference frame at the decoder has errors. The errors may propagate to the future decoded frames until the next instantaneous decoding refresh (IDR) or key frame. In real time applications, reconstructing video contents using a lossy bitstream becomes quite challenging because of the strict constraints on the deadline of displaying the contents [10]. Error resilience and concealment methods are indispensable especially for video delivery over unreliable channels such as wireless networks.

Related Work

Many standards such as VPx and H.26x offer tools for error resilience in their encoder profile. Specifically, H.264 uses data partitioning (DP), flexible macroblock ordering (FMO) and switching P (SP) and switching I (SI) slices [11]. VP9 offers error resilient encoding mode that employs entropy coding context reset and constraints on the motion vector (MV) reference selection. Scalable video coding (SVC) [12] and multiple description coding (MDC) [13] are two popular error resilient coding techniques. A detailed review of error resilient coding methods is presented in [10]. The goal of error concealment is to minimize the amount of distortions at the decoder caused due to packet loss. Various methods such as spatial pixel interpolation, frequency domain reconstruction and temporal motion compensated concealment are proposed. A popular temporal approach boundary matching algorithm (BMA) is proposed to recover the lost motion vectors to avoid error propagation [14]. In [15], a two-stage error concealment is proposed that makes use of available motion vectors, an image continuity preserving method and MAP estimation-based refinement. A method based

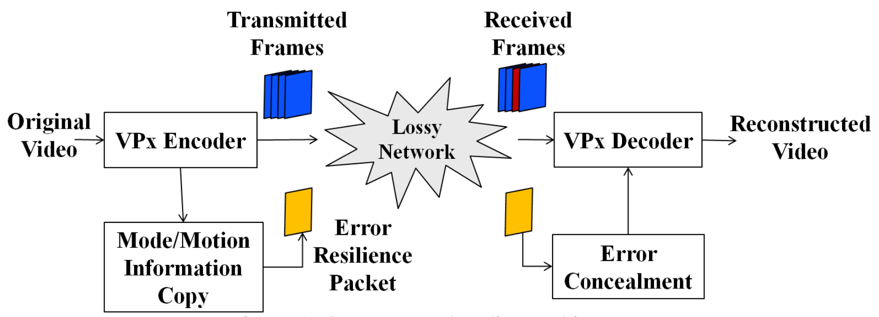


Figure 1: Our proposed coding architecture

on a two-step spatial-temporal extrapolation is described in [16]. Due to high computational complexity of block-matching, BMA and other methods, efforts are made to develop a practical yet effective methods [17]. Many concealment methods are developed for specific SVC and MDC error resilient coding architectures. In [18], an overview of SVC in the context of error resilience and concealment is presented. MDC-based resilience is advantageous because it also allows access to both past and current coding units from other descriptions which may have undergone a different packet loss when transmitted via different channels [19]. Due to its inherent nature of generating more than one descriptions of the source signal, the joint decoding of the received packets includes utilizing the redundancy to conceal the losses [20].

As described in [7], VP9 offers a few resilience tools for communication of conversational video with low latency over an unreliable network. When arbitrary frames lost, it becomes necessary to support a coding mode where decoding can still continue in spite of inconsistencies in the received bitstream. This mode reported a performance drop in the order of 4-5%.

A key assumption is that the drift between the encoder and the decoder is still manageable until a key frame is received. It is important that the arithmetic encoder must be able to decode symbols correctly in frames subsequent to the lost one, in spite of corrupt frame buffers leading to a mismatch. In the current VP9 implementation, a flag "error_resilient_mode" is used to achieve error resilience while encoding a source sequence. This mode restricts encoder in the following ways. First is that the entropy coding context probabilities are reset to defaults at the beginning of each frame. Another restriction is on the MV reference selection, where the colocated MV from previously encoded reference frame cannot be included in the candidate list and sorting of the initial list of MV reference candidates based on search in the reference frame buffer is disabled. However, this cannot prevent drift between the encoder and decoder and a key frame is required for resetting the buffers. It also causes a significant drop in compression efficiency and is not recommended when there is no packet loss. We investigate new error resilience tools for VP9 that can enhance the system performance.

In a typical video coding system, each square block of pixels is predicted from previously decoded set of pixels. Each frame is divided into partitions (and blocks) by the encoder. This prediction can be *INTRA* (same frame) or *INTER* (previous frame) and generally represented in terms of *INTRA* direction mode or motion vectors. Prediction error is transform-coded to provide additional information to the prediction signal [21]. A "good" encoder generates partition and prediction signal such that it minimizes the prediction error. Therefore, in case of frame-loss, it is important to recover its prediction signal that mainly consists of partition, mode and motion information.

In this paper, we propose a VPx-based video coding system that uses duplication of frame-level macroblock prediction information to provide error resilience.

Our Proposed System

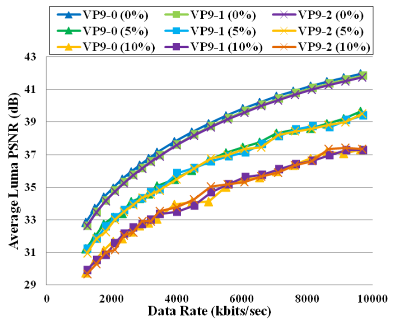
System Architecture: As shown in Figure 1, a video sequence is encoded using a standard VPx encoder with each encoded frame shown in blue. Our proposed system is designed to form an "error resilience packet" (shown in yellow) for a given interval of time. This packet consists of only the prediction information of each frame that was transmitted from the occurrence of last error resilience packet. This packet follows syntax specific to VPx standard. These packets are sent either embedded in the bitstream or over a separate channel. When the VPx decoder receives the bitstream from a lossy network (lost frames in red) it uses the corresponding error resilience packet (when available) to conceal the lost frames to produce a reconstruction signal.

In VPx standard [6], every compressed frame has three or more parts. It begins with an uncompressed data chunk comprising 10 bytes in the case of key (Intra) frames and 3 bytes for Interframes. This is followed by two or more blocks of compressed data known as partitions. The first compressed partition consists of two subsections: (a) Header information that applies to the frame as a whole and (b) Per-macroblock information specifying how each macroblock is predicted from the already-reconstructed data that is available to the decompressor. The rest of the partitions contain, for each block, the quantized DCT/WHT coefficients of the residue signal to be added to the predicted block values [6]

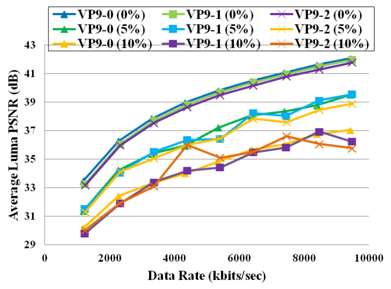
In our proposed system, we duplicate, for each frame, the uncompressed data chunk and the first compressed partition e.g. (a) header and (b) per-macroblock information. We concatenate this information from N frames to form our error resilience packet. This error resilience packet is sent after every N frames. The bitstream produced by our system is not compliant with the current VPx standard.

Error Concealment: When a packet loss occurs, our proposed method the decoder uses the encoded prediction information obtained using the error resilience packet to reconstruct the prediction signals for that frame. This is different than a conven-

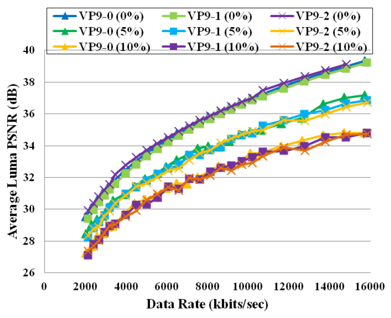
©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-234



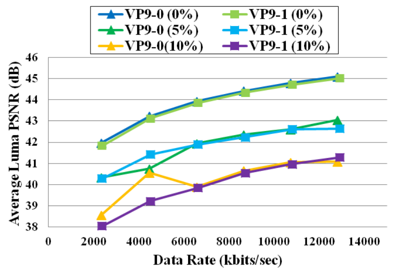
(a) *BasketBallDrill*



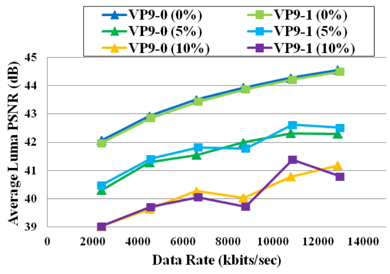
(b) *RaceHorses*



(c) *PartyScene*



(d) *KristenAndSara*



(e) *Johney*

Figure 2: Packet loss performance for test sequences

tional VPx decoder that uses previous frame's encoded or pixel information to estimate the lost frame. For Interframe, the decoder forms motion compensated prediction signal using the motion information: mode, motion vectors and reference frame to reconstruct the Interframe prediction signal. Residue information cannot be recovered because it is not duplicated and sent via the error resilience packet. In case of a lost keyframe, the decoder forms Intra prediction signal for each coded macroblock. However, recovering a lost keyframe using this method is not very effective because often the keyframe relies on the transform coefficients to reconstruct the initial block.

Preliminary Experiments

We modified the VPx software available on the WebM website [22] for our preliminary experiments. We used the VP9 encoding options [7], mainly "codec", "good", "error-resilient", "cpu-used", "target-bitrate", "kf-max-dist" to obtain different encoded bitstreams [23]. Details of these parameters are listed in Table 1.

The following are our encoding commands:

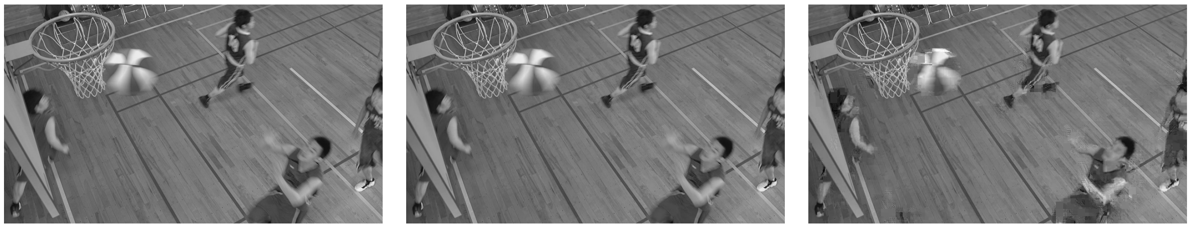
```
./vpxenc -w <Width> -h <Height> --i420 --verbose
 --psnr -o <out.webm> --codec=vp9 --good --cpu-used
=<0/1/2> --end-usage=cbr --fps=<fps>/1 --passes=1
 --target-bitrate=<500-15000> --kf-min-dist=0
 --kf-max-dist=<15/25> --error-resilient=1 <in.yuv>
```

We assume one encoded frame is sent per packet. We also assume that the error resilience packet is always loss-free. In our experiments we set "N" to `--kf-max-dist`. We used a Gilbert model as a packet loss simulator [24]. When the packet loss rate is small, burst length is large; and vice versa [25]. We used burst length of 5 for 5% and 10% packet loss rate. The test sequences used for our experiments are listed in Table 2.

Parameter	Description	Value
`-w, -h`	Spatial dimensions of a frame (width and height) of the video sequence in .yuv format	832×480 or 1280×720
`--fps`	Output frame rate, expressed as a fraction	$30/1$, $50/1$ or $60/1$
`--i420`	Input file uses 4:2:0 subsampling	–
`--good`	`--good` quality and `--cpu-used=0` typically gives quality that is usually very close to and even sometimes better than that obtained with `--best` with the encoder running approximately twice as fast.	–
`--cpu -used`	This sets target cpu utilization $= 100\times\frac{(16-\text{cpu-used})}{16}\%$	0, 1 or 2
`--codec`	Codec to use VP8 or VP9	`vp9`
`--end -usage`	Rate control mode specifying constant bitrate, variable bitrate or constrained quality	`cbr`
`--target -bitrate`	Target bitrate in kbps	500 – 15000
`--error -resilient`	Specifies usage of video conferencing mode	1
`--kf-min -dist`	Minimum keyframe interval (frames)	0
`--kf-max -dist`	Maximum keyframe interval (frames)	15, 25 or 30
`--passes`	Specifies one-pass or two-pass encoding	1
`--psnr`	Shows PSNR in status line	–
`--verbose`	Shows encoder parameters	–

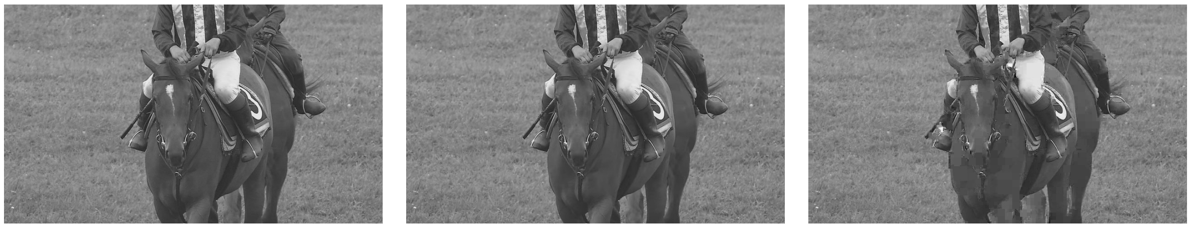
Table 1: VPx encoding parameters used in our experiments

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-234



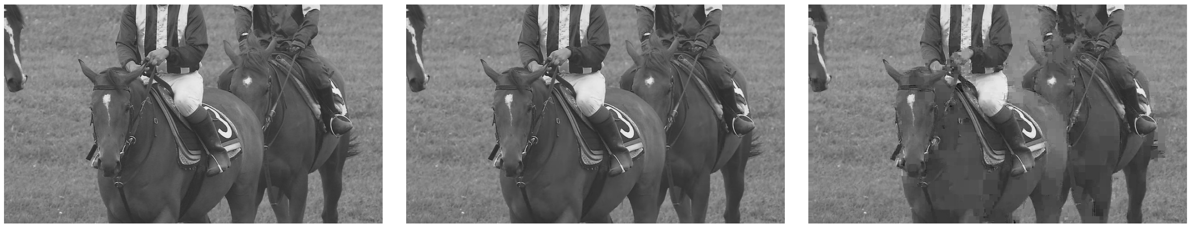
(a) Original (b) No Loss (c) Loss Concealed

Figure 3: *BasketBallDrill* sequence (frame no. 284)



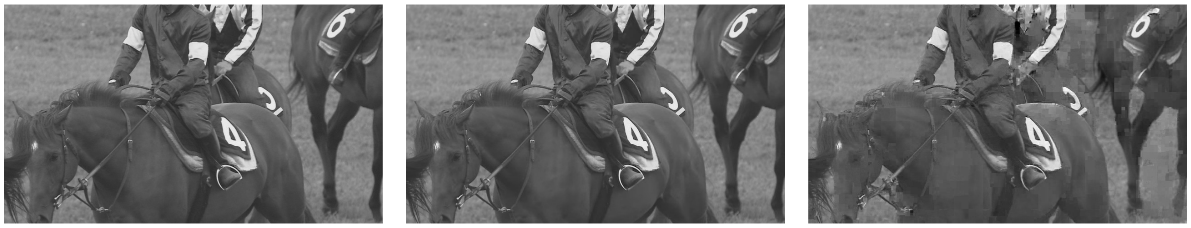
(a) Original (b) No Loss (c) Loss Concealed

Figure 4: *RaceHorses* sequence (frame no. 148)



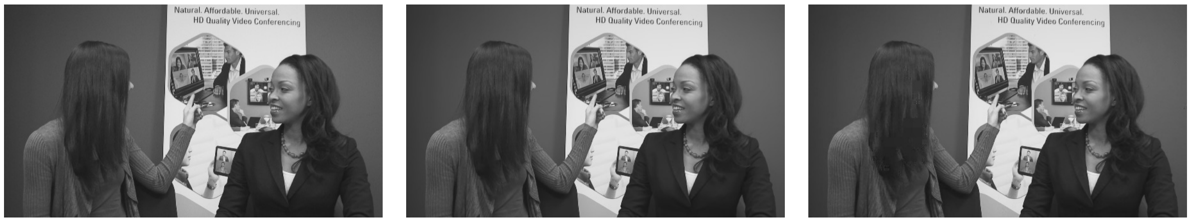
(a) Original (b) No Loss (c) Loss Concealed

Figure 5: *RaceHorses* sequence (frame no. 177)



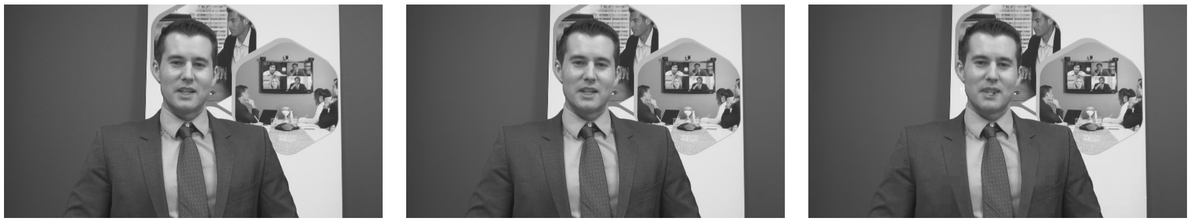
(a) Original (b) No Loss (c) Loss Concealed

Figure 6: *RaceHorses* sequence (frame no. 280)


Natural. Affordable. Universal.
HD Quality Video Conferencing

(a) Original (b) No Loss (c) Loss Concealed

Figure 7: *KristenAndSara* sequence (frame no. 164)



(a) Original (b) No Loss (c) Loss Concealed

Figure 8: *Johney* sequence (frame no. 53)

DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-234

Sequence	Spatial Resolution	Frame Rate	Number of frames	`--kf-max -dist`
BasketBallDrill	832×480	50	500	25
RaceHorses	832×480	30	300	15
PartyScene	832×480	50	500	25
KristenAndSara	1280×720	60	600	30
Johney	1280×720	60	600	30

Table 2: Test sequences used for our experiments

Figure 2 (a) - (e) show the packet loss performance of our proposed method for test sequences. For each test sequence, PSNR vs. data rate is depicted for lossless, 5% and 10% packet loss cases. "VP9-m" indicates our proposed method using the encoding parameter `--cpu-used`, where "m" can take values "0", "1" or "2". Encoded bitstreams for different data rates are obtained using the parameter `--target-bitrate`. Note that, due to the error resilience packets, our bitstreams have a higher data rate (as reported in Figure 2 (a) - (e)) than the actual value specified using `--target-bitrate` for each data point. For example, for *KristenAndSara*, using `--target-bitrate=2000` and `--cpu-used=1` actually produced 2387.78 kbits/sec using our proposed method. The additional data rate accounts for an error resilience packet sent after every *N* frames. Our reported data rate numbers obviously include the additional data rate due to error resilience packets.

For each sequence, our method produces acceptable PSNR values in presence of packet loss. As packet loss increases, our method shows a graceful degradation in performance. When the `--cpu-used` parameter is increased, PSNR performance is also slightly degraded. According to Table 1, `--cpu-used=0` means the highest CPU usage among the values we used ("0", "1" and "2"). Therefore, as this number increases, PSNR typically decreases. This is because the amount of CPU used to encode the bitstream becomes smaller as the value of `--cpu-used` is increased, indicating less efforts taken for encoder-controlled operations such as motion estimation.

Figure 3 - 8 show visual performance of our proposed method using the luma component of some example frames of the test sequences. Each figure contains (a) original frame, (b) decoded frame without any packet loss and (c) decoded frame when it was lost during transmission and concealed at the decoder using our proposed method. As shown in examples from Figure 3 - 8, each lost frame is successfully concealed in most parts of the frame. For *BasketBallDrill* example shown in Figure 3, the areas near moving parts (e.g. the ball, players' hands and legs) contain blocky artifacts. In the example shown in Figure 4, 5 and 6 using *RaceHorses* sequence, the darker moving areas with relatively uniform pixel intensity (e.g. horse) contains blockiness. Figure 7 contains small artifacts in the high-texture area (e.g. the hair of the woman on the left) of *KristenAndSara* sequence. An example of *Johney* sequence, as shown in Figure 8, contains a small artifact in the moving area (e.g. man's face) of the frame. This is because the decoder only has the mode and motion information of that frame and lacks any pixel-wise residue information.

Figure 9 and 10 show examples of failure cases we encountered, when only the prediction information is not sufficient to produce an acceptable image quality at the decoder. However, sending pixelwise residue information means a significant increase in the data rate, which can be forbidden considering the bandwidth limitation imposed by the transmission media. Therefore, our method can generally produce an acceptable concealment outcomes both in terms of PSNR and the visual quality in the case of lost packets.

Conclusions and Future Work

In this paper, we presented a VPx-based error resilient video coding system that uses frame-level macroblock prediction information duplication in the form of "error resilience packet." Our encoder generates an error-resilient bitstream, non-compliant with the existing VPx standard. Based on our preliminary experiments, our system can produce a graceful degradation of video quality in presence of packet loss caused during the video transmission. We plan to compare our results with other error resilience methods such as frame-copy, MV copy, MDC and also our proposed method implemented using the H.264/HEVC standards. We plan to improve our method by providing an effective keyframe concealment strategy and by combining our proposed prediction-based method with pixel-based method such as pixel copy from previous frame.

Acknowledgement

This work was supported by a Google Faculty Research Award.

References

[1] "Cisco visual networking index: Forecast and methodology, 2014-2019," *White Paper*, May 2015, Cisco Systems Inc., San Jose, CA.

[2] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[3] A. M. Tekalp, *Digital Video Processing*. Prentice Hall, 1995.

[4] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, January 2005.

[5] P. Tudor, "MPEG-2 video compression," *Electronics Communication Engineering Journal*, vol. 7, no. 6, pp. 257–264, December 1995.

[6] J. Bankoski, J. Koleszar, L. Quillio, J. Salonen, P. Wilkins, and Y. Xu, "VP8 Data Format and Decoding Guide: RFC 6386," 2013, URL: http://datatracker.ietf.org/doc/rfc6386/, Last accessed: 01/14/2016.

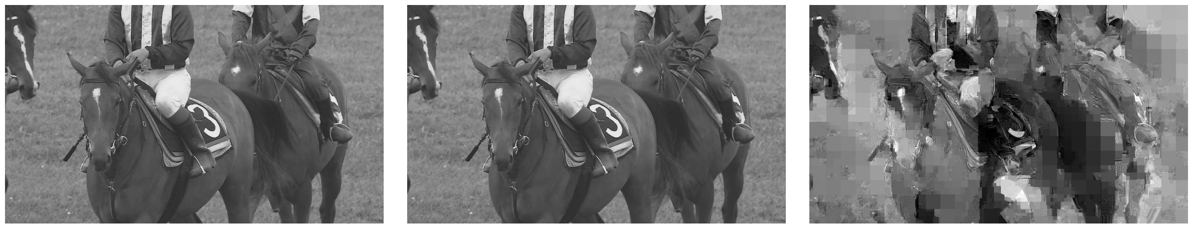
[7] J. Bankoski, R. Bultje, A. Grange, Q. Gu, J. Han, J. Koleszar, D. Mukherjee, P. Wilkins, and Y. Xu, "Towards a next generation open-source video codec," *Proceedings of the SPIE/IS&T Electronic Imaging: Visual Information Processing and Communication IV*, vol. 8666, no. 866606, pp. 1–13, February 2013, Burlingame, CA.

[8] T. Weigand, G. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[9] G. Sullivan, J. Ohm, W. Han, and T. Weigand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, December 2012.

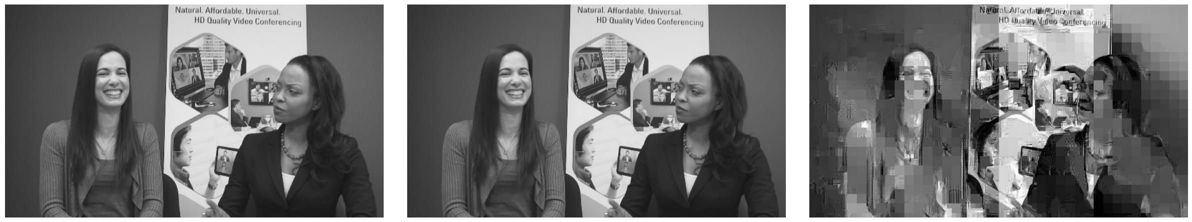
[10] Y. Wang and Q.-F. Zhu, "Error control and concealment for

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-234



(a) Original (b) No Loss (c) Loss Concealed

Figure 9: *RaceHorses* sequence (frame no. 183)


Natural. Affordable. Universal.
HD Quality Video Conferencing

(a) Original (b) No Loss (c) Loss Concealed

Figure 10: *KristenAndSara* sequence (frame no. 360)

video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[11] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, July 2003.

[12] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[13] N. Gadgil, M. Yang, M. Comer, and E. Delp, "Multiple description coding," *Academic Press Library in Signal Processing Vol. 5*, R. Chellappa and S. Theodoridis, Eds. Oxford, UK: Elsevier Ltd., 2014.

[14] W. Lam, A. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, no. 12, pp. 417–420, April 1993, Minneapolis, MN.

[15] S. Shirani, F. Kossentini, and R. Ward, "A concealment method for video communications in an error-prone environment," *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 6, pp. 1822–1833, June 2000.

[16] J. Seiler and A. Kaup, "Adaptive joint spatio-temporal error concealment for video communication," *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pp. 229–234, October 2008, Cairns, Australia.

[17] W-Y.Kung, C.-S. Kim, and C.-C. Kuo, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE Transactions on circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 789–802, July 2006.

[18] Y. Guo, Y. Chen, Y.-K. Wang, H. Li, M. Hannuksela, and M. Gabbouj, "Error resilient coding and error concealment in scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 781–795, June 2009.

[19] W.-J. Tsai and J.-Y. Chen, "Joint temporal and spatial error concealment for multiple description video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1822–1833, December 2010.

[20] N. Gadgil, H. Li, and E. J. Delp, "Spatial subsampling-based multiple description video coding with adaptive temporal-spatial error concealment," *Proceedings of the Picture Coding Symposium*, pp. 90–94, May 2015, Cairns, Australia.

[21] I. E. G. Richardson, *H.264 and MPEG-4 video compression*. Wiley Online Library, 2003, vol. 20.

[22] "The WebM Project," URL: http://www.webmproject.org/, Last accessed: 01/14/2016.

[23] "VP8 Encode Parameter Guide," URL: http://www.webmproject.org/docs/encoder-parameters/, Last accessed: 01/14/2016.

[24] M. Yang, M. L. Comer, and E. J. Delp, "A four-description MDC for high loss-rate channels," *Proceedings of the Picture Coding Symposium*, pp. 418–421, December 2010, Nagoya, Japan.

[25] U. Horn, K. Stuhlmller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, pp. 77–94, September 1999.

Author Biography

Neeraj J. Gadgil received his B.E. (Hons.) in Electrical and Electronics Engineering from Birla Institute of Technology and Science (BITS), Pilani, India in 2009. He is currently pursuing a Ph.D. at the Video and Image Processing Laboratory (VIPER), School of Electrical and Computer Engineering, Purdue University, USA. His research interests are video/image processing.

Edward J. Delp was born in Cincinnati, Ohio. He is currently The Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering at Purdue University. His research interests include image and video compression, multimedia security, medical imaging, multimedia systems, communication and information theory.