

Error Resilient Video Coding with Tree Structure Motion Compensation and Data Partitioning

Yong Wang , Mandyam D. Srinath
Dept. of Electrical Engineering
Southern Methodist University
Dallas, TX 75275
Tel: (214) 7681471, (214)7683114
{ ywang , mds }@seas.smu.edu

Abstract

This paper introduces a new scheme for robust video coding and transmission to reduce the effect of error propagation. In this scheme, the linear structure of motion estimation and compensation in the H.263/H.26L is rearranged to a tree structure. With this structure, the length of error propagation in most of the frames is confined to a specified number and the Macro-Block (MB) intra-refreshing rate is implicitly increased. Since different frames in this structure are of different importance in terms of error propagation, frames are sorted to different priorities according to their importance in confining the error propagation. This property is used to design a new data prioritization scheme, in which different levels of error protection are assigned to packets from different types of frames. Furthermore, it is found that the proposed scheme is complimentary with data partitioning. Thus, a combination of data partitioning with the proposed scheme is also proposed in this paper.

1. Introduction

One inherent problem of video communications is the sensitivity of the compressed video to network error. The most commonly used video compression techniques today are based on MPEG and H.26x standards, where block motion compensation (BMC) is employed. While this technique is very efficient in exploiting inter-frame redundancies, it is very sensitive to channel errors, due to error propagation and accumulation.

Previous schemes to reduce error propagation can be roughly sorted to three categories: Forced Intra Refresh (FIR), Prediction Confinement (PC), and Feedback-Based Error Control (FBEC). FIR tries to reduce error propagation by forcing macro-blocks or frames to be intra-coded based on some strategies [2,4,5,6]. PC restricts the scope of motion estimation either spatially or temporally [2,8], thus restricting the scope of error propagation. FBEC deals with this problem with feedback information from the decoder about the corrupted MB/frame [1,3]. The encoder will avoid using that MB/frame in encoding the following frames. Reference Picture Selection (RPS) [2] and Selective Recovery [12] are all belong to this category. The FIR and PC based techniques are simple to implement. But it is hard to control the trade-off between coding gain and error resilience in such schemes. FBEC based techniques are very efficient, but require a feedback channel which is impossible in many applications or it is impossible for the source to respond to all the receiver requirements, e.g. video broadcasting and multicasting.

Generally, for packet video, video transmission schemes are based on two types of techniques: Forward Error Correction (FEC) and Automatic Retransmission reQuest (ARQ).

But, for real-time video streaming, especially in video broadcasting or multicasting, FEC based schemes are more appropriate to protect the video data from network errors. However, FEC code has the effect of increasing transmission overhead, and hence reduces usable bandwidth for the video data. One way to solve this problem is to assign unequal protection to different video data with different importance. Layered video coding and data partitioning have been used widely to generate video data with different priorities. However, neither of them adopts a mechanism to stop or alleviate error propagation. Thus, when an error occurs at the base layer or most important partition, the video quality degrades rapidly.

In this paper, we propose a new scheme to deal with error propagation. In this new scheme, we rearrange the motion estimation and compensation structure of the H.263 to a tree structure. It is also noticed that different types of frames in this structure have different importance in confining the error propagation. This lead to a new prioritization scheme defined at the frame layer. Compared with data partitioning, which sorts the data within a frame to different priorities, this scheme provides an extra priority at the frame layer. A method of combining the proposed scheme with data partitioning is also proposed to achieve more error robustness.

The rest of the paper is organized as follows. In Section 2, we analyze error propagation in H.263. In Section 3, we propose a tree structure using motion estimation and compensation. In Section 4, we proposed a new data prioritization and unequal error protection scheme. The method to combine the proposed scheme with data partitioning is also presented in this section. Section 5 presents some simulation results. Finally, in Section 6, we summarize the work in this paper and discuss its application.

2. Error Propagation in H.263

In H.263, four types of frames, I-frame, P-frame, PB-frame and B-frame, are defined. While I-frames are totally intra-frame coded, P, B and PB frames employ BMC to exploit temporal redundancy. In BMC, a macro-block in the current frame is compared with MB-size blocks in the reference frame within a specified search region. The minimum value of an appropriate cost function is compared with a threshold to determine the coding mode of the MB. If it is inter-coded, the difference between the macro-block and its best matching block is computed pair-wise for further transform and entropy coding. If it is intra-coded, the original data is coded without motion compensation.

Given the difference in encoding of inter and intra coded macro-blocks, the probability of error recovery of the MB (P_{error}) is different. In inter-coded macro-blocks, this probability is:

$$P_{error} = P_r + P_e \quad (1)$$

In intra-coded macro-blocks, this probability is:

$$P_{error} = P_e \quad (2)$$

Here, P_r is the probability that the reference block is incorrectly recovered and P_e is the probability that the MB itself is corrupted in transmission. To make the analysis tractable, it is assumed that:

- 1) Only I and P frames are considered.
- 2) The error in a macro-block (MB) affects only those MBs in the succeeding frames that are at the same spatial location as the original erroneous MB.

Assumption 1) is based on the fact that the B-frame and B part of PB frame are not used as reference frame in motion compensation and thus do not affect error propagation. So, only I and P frames need to be considered in evaluating error propagation. Assumption 2) is reasonable in video-conferencing applications, in which most parts of the picture are static from frame to frame.

With the above assumptions, P_{error} can be analyzed using a Markov chain model introduced in [7]. It was shown in [9] that P_{error} is a decreasing function of intra-coding frequency and is an increasing function of channel error rate and error propagation length. Therefore, video quality can be improved by reducing the error propagation length and/or increasing the intra-coding frequency.

3. Tree Structure Motion Estimation Compensation (TSMEC)

Motivated by above analysis, we rearrange the linear structure of motion estimation and compensation in H.263 to a tree structure as shown in Figure 1.

- 1) Three types of frames are defined in this structure: root frame, stem frame and branch frame. Root frame is defined as the intra-coded frames of sequence. Stem frame is the frame whose distance from the root frame is multiple of predefined branch length (K). Branch frames are frames other than root frame and stem frame.
- 2) In the tree structure, a branch frame is predicted with reference to the previous frame as in traditional linear structure. If it is a stem frame, it is predicted with reference to its previous stem frame/ root frame. This frame itself becomes a stem frame.

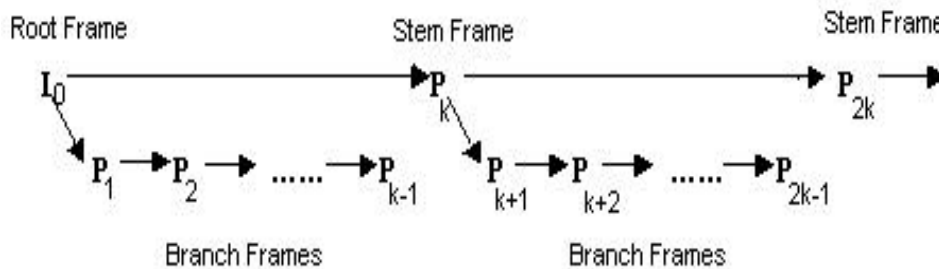


Figure 1: The tree structure motion estimation and compensation

It can be seen that, compared with the linear structure in traditional H.263, the error propagation length is greatly reduced in such a tree structure. For example, given a highly correlated scene of 100 frames, with a branch length of 10, the maximum error propagation length is 10 in the new structure, while it is 100 in traditional H.263. Moreover, since correlation between frames decreases with the distance between them, compared with the traditional linear structure, an implicit increase in intra-coding frequency is expected with the new structure, due to the motion prediction between the stem frames. Finally, with this scheme, the errors in branch frames will not propagate to the next stem frame or beyond. This greatly reduces the probability of error .

Therefore, the new scheme is expected to be more robust to network errors. This new scheme is standard compliant with the application of Reference Picture Selection (RPS) in H.263 and multiple reference frames in H.26L.

4. Data Prioritization

Data prioritization and unequal error protection are efficient techniques to assign error protection to the data which is to be transmitted through error prone networks. By assigning different levels of error protection to data belonging to different priorities, the redundancy budget for error protection can be used more wisely than with equal error protection. Layered coding and data partitioning have been used widely to generate data with different priorities. Here, we propose a new data prioritization scheme based on TSMEC.

4.1 Data Prioritization in TSMEC

In layered coding and data partitioning, the priorities are defined based on a one-dimensional data dependency. Generally, data belonging to the highest priority can be decoded independently. However, data belonging to the lower priorities depend on data at higher priorities for them to be useful for reconstruction of picture.

We note that there is one-directional data dependency within the three defined types of frames in the proposed structure too. That is, the root frames are encoded and decoded independently; stem frames are encoded/decoded depending on root frames; and branch frames are encoded/decoded depending on stem frames. This data dependency is obvious in Figure 1. In the same way as in layered coding, we can define data priorities based on such data dependency. That is, root frames are the most important frames; stem frames are of middle importance; the branch frames are of the lowest importance. The importance of the three types of frames can also be viewed in terms of error propagation confinement. In the tree structure, an error occurring at the branch frames will not propagate beyond the next stem frame. But an error occurring at the stem/root frame will propagate to the following frames without confinement. So, the root frames and stem frames are more important than branch frames. Since the root frames are intra-coded, there are more significant coefficients in the root frames than that in stem frames, which are inter-coded. Loss of root frame causes more severe error than loss of stem frame. Thus, root frames are more important than stem frames.

Therefore, we have a hierarchical data structure with three priorities: the root frames, the stem frames, and the branch frames, from high to low.

4.2 Combining the New Prioritization Scheme with Data Partitioning

Data partitioning is one of the most popular techniques to provide a means of partitioning coded video data into different priorities. With prioritized transmission, data partitioning is very efficient to combat network errors. Data partitioning has been adopted by MPEG-2/4, H.263+ and emerging H.26L. Various types of partition schemes have been proposed. In this paper, we based our discussion on a data partition scheme adopted by H.26L Testing model version 5.2[11]. In this scheme, several syntax elements in one frame are grouped into one partition. The priority of the partition is determined by the dependency of elements. The dependencies between the syntax elements is that, erroneous or missing syntax elements only effect decoding of syntax elements which are in the current branch, away from the root of the dependency tree. Therefore, the impact on the decoded image quality of syntax elements closer to the root of the tree have higher importance for picture reconstruction and

should therefore be assigned to higher priority classes. Thus, the following six partitions are defined:

- Partition 1: Psync, Ptype, MBtype, REFframe
- Partition 2: IntraPredMode, MVD
- Partition 3: CBP_Intra, LUM_DC_Intra, CHR_DC_Intra
- Partition 4: LUM_AC_Intra, CHR_AC_Intra
- Partition 5: CBP_Inter, LUM_DC_Inter, CHR_DC_Inter
- Partition 6: LUM_AC_Inter, CHR_AC_Inter

Figure 2 shows the priority assignment in this data-partitioning scheme. From this figure, we can find that data partitioning partitions data within each frame into six priority groups (priorities). Besides, we may also find that this scheme treats each frame equally. That is, the same partition from all the frames is sorted to the same priority as shown in Figure 2. This is also true with data partitioning schemes defined in other video coding standards. However, in practice, different frames are of different importance. For example, in MPEG, I, P and B frames are of different importance. In the TSMEC scheme, root frame is more important than stem frame, which in turn is more important than branch frame.

To further improve the efficiency of data partitioning, we introduce frame layer priority into data partitioning by combining the TSMEC scheme with data partitioning. In combining the two schemes, priorities are determined based not only on the dependencies of elements within each frame, but also depend on the frame itself. Therefore, not only are the different partitions of different priorities, but also the same partition from different types of frames has different priorities. It is possible that a low priority partition belonging to a high priority frame is assigned a higher priority than a higher priority partition from a low priority frame. This idea is shown in Figure 3, in which the three frame layer priorities are combined with priorities defined in data partitioning. The branch length in this example is defined as K .

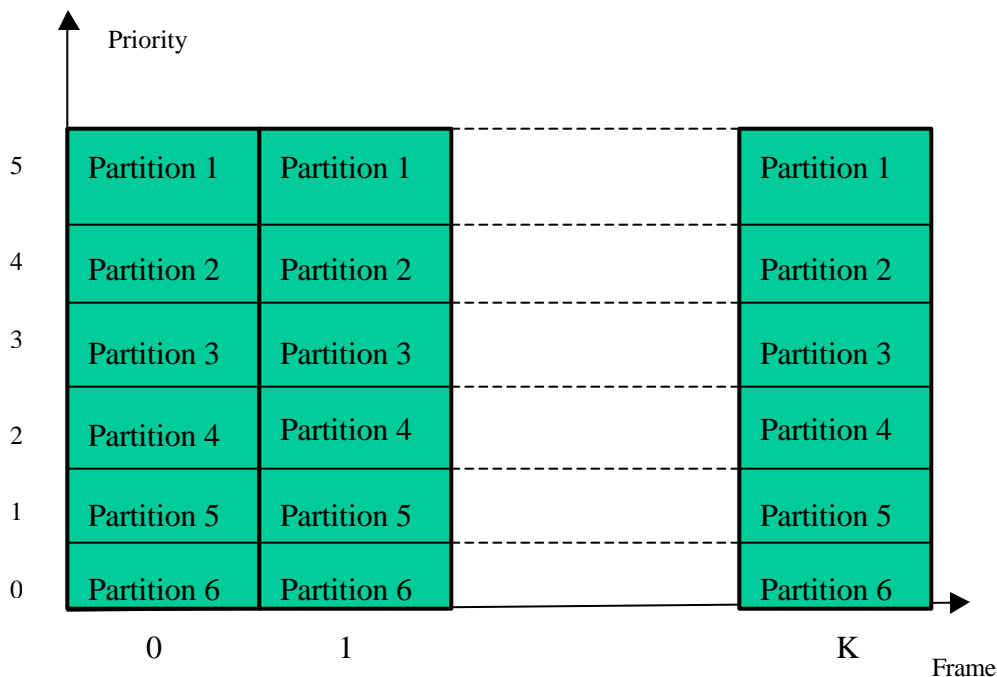


Figure 2: Priority assignment in data partitioning

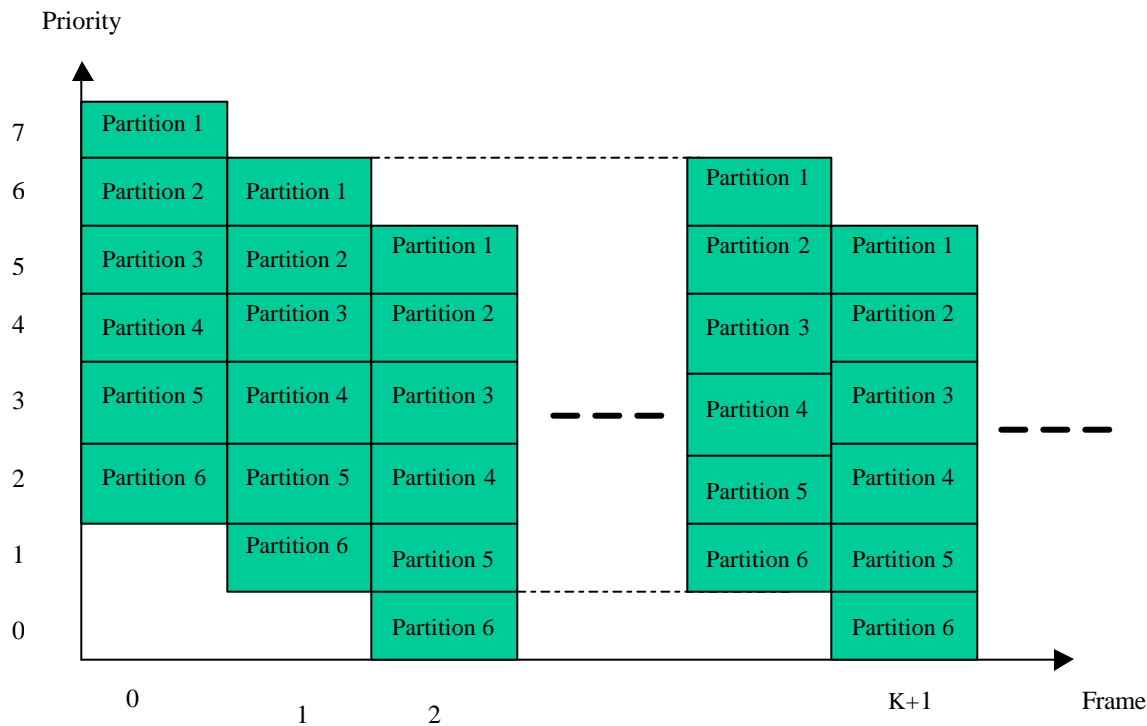


Figure 3: Priority assignment in the proposed scheme

5. Simulation Results

To test the error resilience of the proposed scheme, we did simulations with a TSMEC (Branch length equals 4) Codec and a H.26L Codec based on H.26L Testing Model version 5.2. Data partitioning was employed in both cases.

The network simulator was a pseudo-mux software emulating H.223 Annex D. Six error-pattern files, as shown in table 1, were used for emulating the bit error in mobile networks. 64KBPS, 128KBPS and 384KBPS channel rates were assumed according to the error pattern files. All of them were proposed and commonly accepted by ITU – Telecommunications Standardization Sector, STUDY GROUP 16, Video Coding Experts Group (Question 15) for emulating errors in mobile networks [13][14].

We first use the packetizer to obtain the packetized bitstream and apply unequal error protection to the packets based on their priorities according to table 2. The error protection was based on Forward Error Correction (Reed-Solomon code). Then the packets were fed into network simulator, which was driven by one of the error pattern files to apply corresponding bit errors to the packets. The output of the network simulator was first fed into Reed-Solomon decoder and then to the video decoder to recover the video signal. For fair comparison of the two schemes, the bit rates of compressed video, as well as the total error protection applied was the same for both schemes.

The mapping of partitions to the applied error protection capability for these resulting partitions are provided in table 2. Here, DP denotes data partitioning scheme. TSMEC+DP denotes the combination of the proposed scheme with data partitioning. The number in the table is the amount of FEC codes assigned to the corresponding H.223 multiplexing packets. This number also defines the error correction ability for the bit-error in the packet.

Three testing sequences, Foreman (QCIF), Silent (QCIF) and Paris (CIF), were used in all simulations. For all sequences, 4000 coded frames were used to avoid the influence of distribution errors in the error patterns. For this requirement, those sequences were looped. All sequences were coded with a fixed frame-rate. In the simulation, 75% of the channel rate was assumed available for video. The quantization parameters used in the simulation were chosen to meet the target bitrate.

Simulation results are listed in Tables 3 – 8. The experiment results are also plotted in Figures 4 -- 6. From the results, we can find that the proposed scheme achieves better results in terms of error resilience, when error rate is high. One trend is clear from the result is that the video quality degrades slowly with the proposed scheme. In comparison, data partitioning degrades rapidly when bit error keeps increasing. This is because that there is no error propagation confinement mechanism adopted in data partitioning. Thus, when the network error becomes higher, better performance is achieved by the proposed scheme. For example, in the foreman sequence, when the bit error rate is 1×10^{-3} , the TSMEC+DP outperforms DP 4.16 dB in average PSNR, while it is only 1.68 dB when bit error rate is 5×10^{-4} , with Foreman sequence. We may also find several cases, in which the proposed scheme does not outperform. For example, with Silent sequence, when the error rates are at 1.6×10^{-4} and 1.3×10^{-4} , the data-partitioning scheme achieves better results. This is because the proposed scheme introduces redundancy at source encoder and sacrifices coding gain at this stage. When network error is low, data partitioning alone is enough to combat transmission error, without sacrificing coding gain in. One way to improve performance at low error rates in the proposed scheme is to adaptively change the branch length depending on the network error rate. That is, when the error rate is low, we can change to the linear structure in motion compensation. This can be achieved by simply forcing the branch length to be one.

Table 1: Error pattern files for bit error environment

Error-Pattern File	Bandwidth (KBPS)	Bit Error Rate
3_64	64	5×10^{-4}
4_64	64	1.6×10^{-4}
5_64	64	1×10^{-3}
6_64	64	1.3×10^{-4}
7_128	128	4.7×10^{-4}
8_384	384	4.7×10^{-4}

Table 2: Mapping of partitions to error protection capability

Partition	TSMEC+DP		DP
	STEM FRAME	Branch Frame	
1	32	16	32
2	32	8	16
3	16	4	8
4	16	0	4
5	8	0	2
6	8	0	0

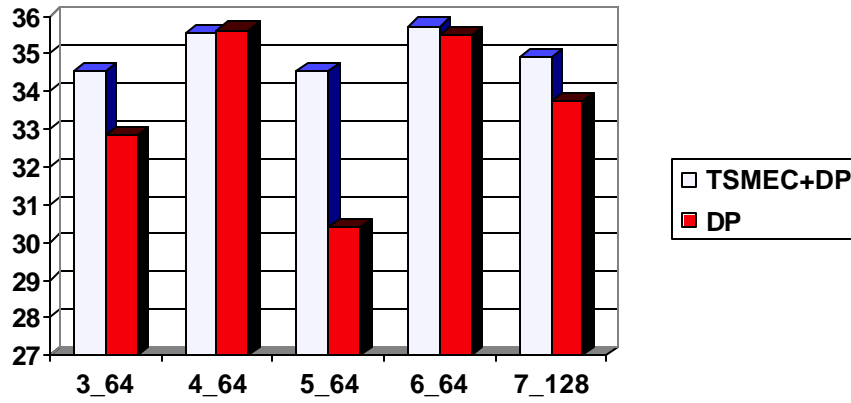


Figure 4: Comparison of average PSNR from two schemes in bit error (Foreman)

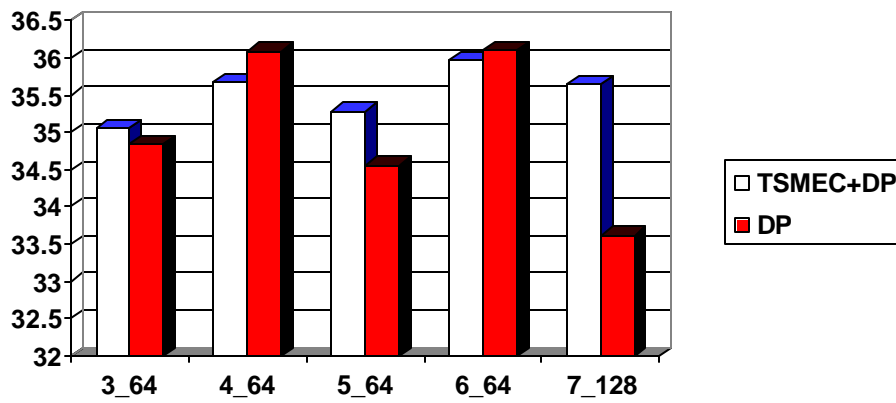


Figure 5: Comparison of average PSNR from two schemes in bit error (Silent)

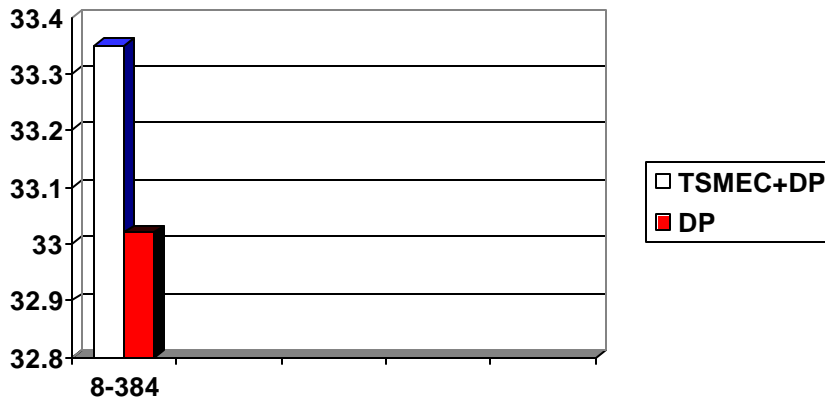


Figure 6: Comparison of average PSNR from two schemes in bit error (Paris)

Table 3: Simulation result with Error Pattern file 3-64

Mode	quant	Sequence	video+overhead (BPS)	Videobr (BPS)	psnr-y	psnr-u	psnr-v	Psnr
TSMEC+DP	25	Foreman	56161	42588.61	28.216	37.53	37.844	34.53
DP	23	Foreman	58654	44605.66	25.392	36.25	36.9	32.85
TSMEC+DP	21	Silent	47823	35952.35	30.258	36.662	38.252	35.06
DP	19	Silent	50208	37394.64	29.582	36.646	38.29	34.84

Table 4: Simulation result with Error Pattern file 4-64

Mode	quant	Sequence	video+overhead (BPS)	Videobr (BPS)	psnr-y	psnr-u	psnr-v	Psnr
TSMEC+DP	25	Foreman	56161	42588.61	30.136	38.038	38.45	35.54
DP	23	Foreman	58654	44605.66	30.128	38.052	38.714	35.63
TSMEC+DP	21	Silent	47823	35952.35	31.715	37.005	38.52	35.75
DP	19	Silent	50208	37394.64	32.44	37.44	38.695	36.19

Table 5: Simulation result with Error Pattern file 5-64

Mode	quant	Sequence	video+overhead (BPS)	Videobr (BPS)	psnr-y	psnr-u	psnr-v	Psnr
TSMEC+DP	25	Foreman	56161	42588.61	28.292	37.498	37.88	34.56
DP	23	Foreman	58654	44605.66	22.45	33.86	34.89	30.40
TSMEC+DP	21	Silent	47823	35952.35	30.73	36.76	38.335	35.28
DP	19	Silent	50208	37394.64	28.725	36.715	38.235	34.56

Table 6: Simulation result with Error Pattern file 6-64

Mode	quant	sequence	video+overhead (BPS)	Videobr (BPS)	psnr-y	psnr-u	psnr-v	Psnr
TSMEC+DP	25	foreman	56161	42588.61	30.368	38.208	38.646	35.74
DP	23	foreman	58654	44605.66	30.146	37.864	38.554	35.52
TSMEC+DP	21	silent	47823	35952.35	32.175	37.155	38.55	35.96
DP	19	silent	50208	37394.64	33.09	37.39	38.765	36.42

Table 7: Simulation result with Error Pattern file 7-128

Mode	quant	sequence	Video+overhead (BPS)	Videobr (BPS)	psnr-y	psnr-u	psnr-v	Psnr
TSMEC+DP	25	foreman	116210	76838.86	29.092	37.648	38.032	34.92
DP	23	foreman	111680	75831.59	27.494	36.558	37.216	33.76
TSMEC+DP	21	silent	109470	74366.13	31.718	36.926	38.292	35.65
DP	19	silent	110820	73238.29	27.09	35.44	38.342	33.62

Table 8: Simulation result with Error Pattern file 8-384

Mode	quant	sequence	Video+overhead (BPS)	Videobr (BPS)	psnr-y	psnr-u	psnr-v	Psnr
TSMEC+DP	22	Paris	249680	211419.3	28.13	35.54	36.37	33.35
DP	20	Paris	247020	216742.06	27.17	35.48	36.41	33.02

6. Conclusions

In this paper, we have proposed a tree structure motion compensation scheme based to combat error propagation in network video. The proposed scheme introduces a new prioritization scheme at the frame layer so that unequal error protection can be applied. A method of combining this scheme with data partitioning is also proposed in order to achieve more error robustness. On commonly used testing sequences, the scheme is found to give superior error resilience results compared with data partitioning, especially when network error rate is high. It is a very efficient scheme of error resilient video coding for transmission over error prone environments, such as wireless networks.

7. Acknowledgement

This research was supported, in part by Nokia Mobile Phones. We are thankful to Viktor Varsa of Nokia Research Center, Irving, Texas, for his suggestions and help during this research.

8. References

- [1] Yao Wang, Qin-Fan Zhu, "Error Control and Concealment for Video Communication: A Review", Proceedings of the IEEE, Vol.86, No.5, May 1998, PP.974-997.
- [2] "Video Coding for Low Bitrate Communication", International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.263, 1998.
- [3] Jong-Tzy Wang and Pao-chi Chang, "Error-Propagation Prevention Technique for Real-Time Video Transmission over ATM Networks", IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No. 3, Apr. 1999, PP.513-523.
- [4] P. Haskell and D. Messerschmitt, "Resynchronization of Motion Compensated Video Affected by ATM Cell Loss", in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), San Francisco, CA, USA, vol.3, March 1992, PP.545-548.
- [5] J. Liao and J. Villasenor, "Adaptive Intra Update for Video Coding over Noisy Channels", in Proc. IEEE Int. Conf. On Image Processing (ICIP), Lausanne, Switzerland, Vol.3, Sept. 1996, PP. 763-766.
- [6] Koji Imura and Yutaka Machida, "Error Resilient Video Coding Schemes for Real-time and Low bitrate Mobile Communications", Signal Processing: Image Communication, 14(1999) 519-530.
- [7] Madhukar Budagavi and Jerry Gibson, "Multiframe Video Coding for Improved Performance Over Wireless Channels", IEEE International Conference on Image Processing, Oct., 1997, Santa Barbara CA, PP.89-92.
- [8] S. Wenger, "Video redundancy coding in H.263+", In proc. AVSPN, Aberdeen, UK, Sept. 1997.
- [9] Yong Wang and Mandyam D. Srinath, "Robust Video Coding Based On Tree Structure Motion Compensation", IEEE DSP2000 Workshop, Hunt, Texas, USA, Oct. 2000.
- [10] K. Stuhlmuller, M. Link and B. Girod, "Scalable Internet Video Streaming With Unequal Error Protection", Packet Video workshop 99, April 1999, New York, USA.
- [11] ftp://standards.pictel.com/q6-site/0008_Por/q15k18.doc
- [12] Masahiro Wada, "Selective recovery of video packet loss using error concealment", IEEE Journal on Selected Areas in Communications, Vol. 7, No. 5, June 1989, PP. 807-814.
- [13] <ftp://standards.pictel.com/q6-site/Q15-I-53.doc>
- [14] <ftp://standards.pictel.com/q6-site/Q15-I-60.doc>