Title
Comparison of algorithms for concealing packet losses in the transmission of compressed video

Author
Bustamante, Maria Jose

Publication Date
2010

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Comparison of Algorithms for Concealing Packet Losses in the Transmission of Compressed Video

A Thesis submitted in partial satisfaction of the requirements for the degree Master of Science

in

Electrical Engineering (Signal and Image Processing)

by

Maria Jose Bustamante

Committee in charge:

Professor Pamela C. Cosman, Chair
Professor Truong Nguyen
Professor Bhaskar Rao

2010

The Thesis of Maria Jose Bustamante is approved and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2010

DEDICATION

I would like to dedicate this thesis to my beloved mother, Maria L. Bagne, whose unconditional love and support have been essential in helping me get this far in my life journey. I cannot express in words the love, respect, and admiration that I have for my mother.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Pamela Cosman for her help and support in the past two years that I have spent at UCSD. I would like to express my gratitude to her for letting me be a part of her team and for her guidance at different stages of my research project.

I would also like to acknowledge Professor Truong Nguyen for always welcoming questions, and for providing me with physical space in his lab even though I am not part of his research group.

I would like to acknowledge Jihyun Shin for her significant contribution to this research experiment, and Ting-Lan Lin for his help.

Finally, I would like to thank my family and friends for supporting me in these toughest years of my academic studies.

ABSTRACT OF THE THESIS

Comparison of Algorithms for Concealing Packet Losses in the Transmission of Compressed Video

by

Maria Jose Bustamante

Master of Science in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2010

Professor Pamela C. Cosman, Chair

Information loss in video tends to reduce its visual quality since some glitches or artifacts can be detected by humans. Over the past few years, great efforts have been made to reduce packet loss and its visibility in videos. Forward error correction (FEC) methods can be used at the encoder to prevent packet loss. Error concealment methods can be used at the decoder to reduce the visibility of lost packets if the error correction algorithms have failed. One way to prevent the decrease of quality in videos due to packet loss is to tag packets at the encoder according to their priority levels. A problem with this prioritization method is that typically the encoder has no information about what kind of error concealment algorithm the decoder uses. This

thesis presents the experiment that we performed in order to determine to what extent the encoder prediction of packet importance is independent of the error concealment method used at the decoder. In this experiment, four different types of packet loss were introduced into nine different videos. Each lossy version of each video was then decoded using five types of error concealment methods. A Visual Quality Metric was then used to rate the concealed videos. Results show that for each type of loss, the error concealment methods did not differ significantly, thus indicating that it is possible to create a packet priority tagging algorithm that is largely independent of the error concealment method used at the decoder.

1. INTRODUCTION

Packet loss in video transmission has been a problem since the implementation of networks and compression algorithms. Packets are frequently dropped in a congested network causing the appearance of glitches or artifacts in a video. Several aspects of packet loss have been studied, which range from priority tagging for packet dropping at the encoder to error concealment algorithms at the decoder.

Despite the vast amount of research that has been done in the field of packet loss, there is still a lot of room for improvement in its different aspects. An interesting aspect of packet loss is error concealment (EC), which is used to reduce the visibility of packet loss in a video. There are many different methods for error concealment that operate in the spatial, temporal, frequency domains, or different combinations of domains [1].

The authors of [2] present a novel post-processing shape error concealment algorithm. The proposed approach is based on matching boundary segments in the current frame to the boundary in the previous frame and reconstructing the missing boundary pieces using motion compensation. This approach has good subjective performance; however, it has a high computational complexity.

Kwok and Sun [3] proposed a spatial interpolation algorithm that uses spatially correlated edge information and performs directional interpolation to conceal the lost block. This algorithm performs well with blocks that contain edges in different directions. However, when multiple edges are present in a single block, the algorithm is not as effective.

Orientation-adaptive interpolation is presented in [4]. This technique uses sequential recovery in such a way that previously recovered pixels are used to recover other missing pixels. The orientation-adaptive interpolation algorithm used is derived from a pixel-wise (instead of block-wise) statistical model. This algorithm performs well on smooth-textured blocks. Nevertheless, for high-detail blocks, this concealment technique introduces blurring artifacts and discontinuity.

Countless other error concealment techniques exist that range from hybrid and adaptive methods [5, 6], to second generation error concealment [7]. Different methods produce different results, but they all have the same goal: to improve the quality of the video in such a way that the viewer does not perceive or barely perceives packet loss.

Previous research done by our group [8, 9] involved human subjective experiments in which packets were lost in a video and were then concealed at the decoder by a specific concealment method. Viewers then indicated whether or not they could see the packet losses, which appear as glitches in videos. If the concealment was good enough, the viewers would not notice the artifact and would not respond to it. On the other hand, if the concealment was not good enough, then the viewers would see the glitch and respond by hitting the space bar. The goal of this experiment was to produce a model which could predict, for each packet, the probability that that packet would be noticed by a viewer if it were lost. The results of this research were therefore dependent on the concealment algorithm.

In the second phase of our error concealment research, we wanted to examine how much effect the error concealment has on the results. In general, when transmitting a video over a network, the encoder does not know what error concealment strategy the decoder is going to use. To create a prediction model for packet loss visibility that is independent of the error concealment algorithm at the decoder, we need to analyze different results obtained when the decoder uses distinct error concealment algorithms.

This thesis describes the second phase of our research project. We used five different error concealment algorithms to decode videos with four different types of packet loss. A computable quality metric (VQM) is then used to evaluate the decoded videos. VQM (Visual Quality Metric) assigns a score to the entire video or to a segment of video, such as a group of pictures (GOP), as opposed to an individual slice or frame. The score has been shown to correlate well with how humans perceive the quality of the video [10].

It should be noted that the purpose of this research project is not to create an error concealment algorithm that surpasses the quality of already existing ones, but rather to implement and compare already existing algorithms and categorize them according to their perceived quality. If all algorithms result in comparable quality, then that indicates that we can create a packet prioritization algorithm at the encoder independent of the error concealment method used at the decoder.

This thesis is organized as follows: Section 2 describes the experiment setup. Section 3 presents the results and experimental comparisons. Finally, Section 4 is the conclusion.

2. EXPERIMENT SETUP

This research project consisted of five steps: 1) packet loss simulation, 2) decoding, 3) segmentation, 4) yuv to avi conversion, and 5) VQM. Figure 1 presents the diagram showing the steps involved in processing a single video in this project. A total of nine different original videos were used in this experiment.

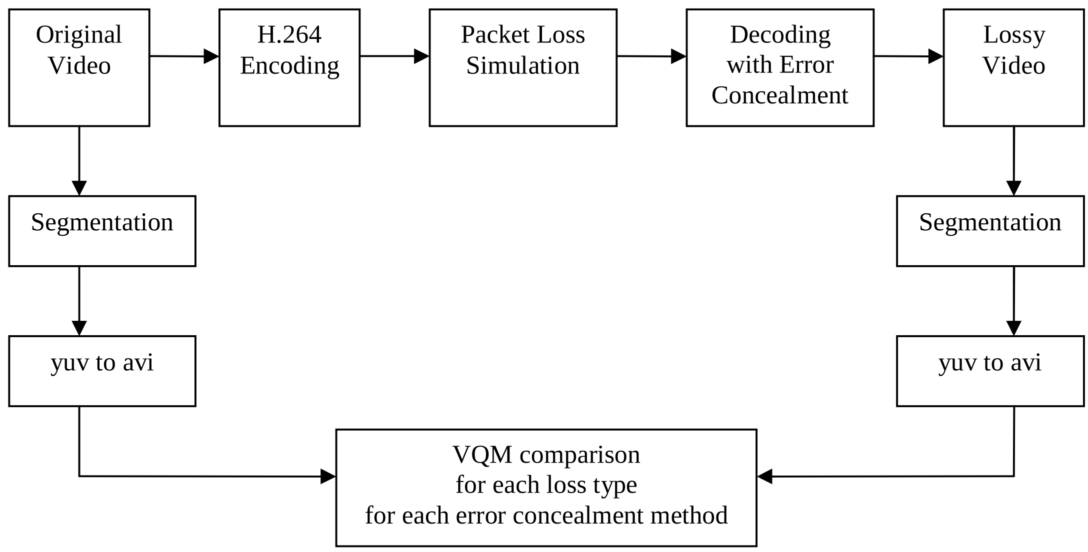

Figure 1: Steps involved in this research project.

The videos were encoded and decoded using the H.264/AVC video coding standard. The JM12.1 decoder from HHI was used for four error concealment algorithms and was modified accordingly [11]. The FFmpeg decoder was used for the remaining error concealment algorithm [12]. These methods are described in Section 2.2.

All frames in each video to be tested were divided in slices. A slice consisted of one horizontal row of macroblocks (MB). Each macroblock had a fixed size of sixteen by sixteen pixels in the luma component, and eight by eight pixels in the chroma components.

The videos were divided into groups of pictures (GOP). A GOP is a collection of frames in which the first frame is an I (intracoded) frame, and the remaining frames are P and B frames ordered in a set pattern. The first and last group of pictures (GOP) of each video were not used for placing errors (packet losses), but all other GOPs contained introduced errors. The settings for the decoder and encoder used in our experiment are listed in Table 2. The details of the videos used are listed in Table 3.

Table 1: Settings used by the decoders.

Decoders	1. HHI JM 12.1 2. FFmpeg
Resolution	720 x 480
Bit rate	2.1 Mbps
Frame Rate	30 fps
Type	Progressive
GOP sequence	IBBPBBPBBPBBPBB
Error Concealment Methods	1. Intra EC 2. Default EC 3. Motion Compensated EC 4. Zero Motion EC 5. FFmpeg EC

Table 2: Videos used in the experiment.

Video Name	Number of frames	GOPs used
Sony_golf_telus_world_skins	2538	2 to 168
Philips_formula1	2780	2 to 184
Philips_airshow	2976	2 to 197
Sony_indanapolis_racing	3551	2 to 235
Sony_beach_soccer_world_cup	3961	2 to 263
JVC_stories_of_life	4411	2 to 293
Pioneer_hawaiian_style	5300	2 to 352
JVC_new_york	5661	2 to 376
Hitachi_symphony_of_the_Earth	8056	2 to 536

2.1 Step 1: Packet Loss Simulation

To simulate packet loss, we created four different types of error patterns in the video. The dropping patterns were not combined in a single video; that is, four lossy versions of each video were created, one for each pattern. Only full (not partial) slice losses were simulated; that is, the loss spanned the full width of a frame.

In order to calculate a VQM score, the number of frames in the original video must be equal to the number of frames in the lossy video; therefore, the first slice of each frame could be not dropped since it contains the header information and loss of that slice would entail loss of the frame. The four types of loss are described below.

2.1.1. Single Slice (SS):

Idea: A single slice is dropped in one frame per GOP in a video.

Implementation: One frame per GOP is chosen randomly, and one slice (other than the first slice) is randomly dropped from that frame.

2.1.2 Whole Frame (WF):

Idea: A single entire frame is dropped per GOP.

Implementation: One frame per GOP is chosen randomly, and all slices except for the first slice are dropped. Because the first slice is retained, the decoder will be aware that there is a frame, but it will have to conceal almost the entire frame since it only retained the first slice. No I frames were dropped since most error concealment methods use some kind of spatial interpolation to conceal the loss in an I frame.

2.1.3. Multiple Slices Single Frame (MSSF):

Idea: Multiple slices are dropped in one frame per GOP.

Implementation: One frame per GOP is chosen randomly, and several slices are dropped in the chosen frame. A minimum of two slices and a maximum of ten slices are dropped per frame. Both the number of slices and the choice of which slices are random.

2.1.4. Multiple Slices Multiple Frames (MSMF):

Idea: A slice is dropped in different frames per GOP.

Implementation: Several frames are randomly selected per GOP, and a slice is dropped in each selected frame. A minimum of two frames and a maximum of five frames are chosen per GOP. The number of frames is chosen randomly, as are the specific frames and the slice within the frame.

2.2 Step 2: Decoding

A lossy version of each video was decoded using each of the five error concealment algorithms. The five error concealment methods are described in detail below.

2.2.1. Intra-frame Error Concealment

This error concealment algorithm is an implementation of the one patented by WebTV Networks, Inc. [13]. This is an intra-frame error concealment which means that it uses spatial interpolation of pixels within the same frame to cover the loss. This algorithm selectively rotates, weights, and adds uncorrupted slices to conceal lost slices.

Specifically, if lost slices are detected, the decoder checks each slice in the frame to find which slices have been lost. It is then determined if the frame lost the first or last slice. When packet loss in either the first or last slices is detected, only the slice below or above is selected, respectively. If the lost slice is not the first or last slice in the frame, both the slices above and below it are selected to help with the concealment of the lost slice. The slice above the lost slice will be referred to as slice A, and the slice below will be referred to as slice B.

After the decoder has selected the slices that will be used for concealment, the selected slice(s) are flipped toward the lost slice resulting in slices A' and B'. That is, the last pixel row in A becomes the first pixel row A'. The second-to-last pixel row in A becomes the second pixel row in A', and so on until the first pixel row in A becomes the last pixel row in A'. Similarly, the first pixel row in B becomes the last pixel row B’. The second pixel row in B becomes the second-to-last pixel row in B’, and so on until the last pixel row in B becomes the first pixel row in B’.

After the flipping process, each pixel row in A’ and B’ is weighted by a number from zero to one resulting in slices A’’ and B’’. The weighting is done in such a way that there is a smooth transition between the lost slice and the slices above and/or below, so pixels closer to the edges get higher weights than pixels farther away from the edges. When only the first or last slice is lost (i.e. only the slice above or below is used for concealment), the weighting is uniform throughout all pixel rows in the slice.

The final step in this error concealment algorithm is to add the weighted flipped slices (i.e. A’’ and B’’). Since there is only one slice that is used for error concealment when the loss is in the first or last slice, the final step is the weighting process, and that single slice replaces the lost slice.

It should be noted that since this algorithm uses information from the same frame to conceal loss, videos with whole frame loss could not use this intra-frame type of concealment since all slices (except for the first slice) are lost in whole frame loss.

2.2.2. Default JM12.1 Error Concealment

For I frames: This method performs a pixel-based interpolation of intact or concealed neighboring macroblocks. It uses a weighted average according to distance for the interpolation [11].

For P and B frames: It conceals the missing blocks using either the “by copy” or the “by trial” method. If the average motion vector of the correctly received macroblocks is less than a set threshold, concealment by copy is used; otherwise, concealment by trial is used. Concealment by copy is simply zero motion error concealment (described below). Concealment by trial conceals a given macroblock by using the motion vector of one intact or concealed neighboring macroblock. The selected motion vector is the one that gives the lowest pixel difference at the edges of the concealed macroblock compared to the surrounding macroblocks [11].

2.2.3. Motion Compensated Error Concealment

The Motion Compensated Error Concealment (MCEC) method estimates a motion vector to conceal a lost macroblock [14]. Each macroblock can have from one to sixteen motion vectors depending on the level of motion compensation used, which can range from the macroblock level to the smallest 4x4 block level. A set of motion vectors from the macroblocks that surround the lost macroblock is created. The motion vectors for a specific macroblock reference different frames, but only the frame that is referenced the most in the whole set of motion vectors is selected for concealment. The median of motion vectors in the set that refer to the selected reference frame is calculated. The estimated motion vector to use for concealment is the calculated median of motion vectors.

2.2.4. Zero Motion Error Concealment

For I frames: Concealment was performed as in the default mode (pixel-based interpolation).

For P and B frames: Zero Motion Error Concealment (ZMEC) conceals a given macroblock by copying from the reference frame the pixel area that is at the same spatial location as the lost macroblock [11]. Only I and P frames are used as reference frames for concealment.

2.2.5. FFmpeg Error Concealment

In FFmpeg error concealment [12], the first step is to determine how many macroblocks are intact (not lossy). If the number of intact macroblocks is greater than a set threshold, then a type of Intra error concealment is performed. Otherwise, a type of Inter error concealment is performed.

For Intra-error concealment, the FFmpeg decoder does a process called FillDC. In this process, for each 8x8 block in the luma component, or each 16x16 block in the chroma components, the decoder looks at the four directions surrounding the block (top, bottom, left, right) to find uncorrupted blocks. It then finds the pixel average of each uncorrupted neighboring block. Finally, it takes a weighted average of the uncorrupted averaged blocks and the result is the block that is used for concealment.

For Inter-error concealment, the decoding is done differently for I, P, and B frames. For I and P frames, several possible motion vectors are calculated for each damaged macroblock based on motion vectors of neighboring undamaged and concealed macroblocks. Several methods are used to calculate possible motion vectors, including median and mean of surrounding motion vectors, and zero-motion. The motion vector that produces the most continuous edges between the concealed macroblock and the neighboring macroblocks is selected for concealment. For B frames, the decoder uses a forward weighted version and a backward weighted version of the motion vector of the collocated nearest P reference frame.

2.3 Step 3: Segmentation

After the lossy versions of the nine videos were decoded using each of the five error concealment methods, each decoded video was then segmented into groups of pictures (GOPs). As indicated in Table 2, each GOP consists of fifteen frames and has sequence IBBPBBPBBPBBPBB. All but the first and last GOPs of each video were used for this research experiment.

2.4 Step 4: yuv to avi Conversion

Once all the videos were segmented into GOPs, each GOP was then converted from yuv format to avi format. This conversion was necessary because VQM can only process videos in avi format.

2.5 Step 5: Visual Quality Metric (VQM)

The last step in this research project was to assign a score to each GOP of each processed video using VQM, which compares the original lossless video and a processed version of the video. Scores range from zero to one, where a lower score means higher quality. A score of zero indicates no difference between the original lossless video and the processed video [10].

Each original video was also segmented into GOPs. Then each lossy GOP was compared with the corresponding original GOP using VQM, which assigned a score to the lossy GOP. Scores varied depending on the types of loss and the types of concealment method, as well as the underlying video content. A total of 2596 GOPs per error type per error concealment algorithm were compared using VQM.

3. RESULTS

To help the reader visualize what lossy and concealed versions of a frame would look like, Figures 2 to 8 display the decoded frame with no packet loss, the decoded frame with an unconcealed packet loss, and the concealed frames using the five different error concealment methods implemented in this experiment. The frame displayed in Figures 2-8 is a P frame from a video that has very little motion. Figures 9 to 15 also display the decoded frame with no packet loss, the decoded frame with an unconcealed packet loss, and concealed frames using the five different error concealment methods of a video. In contrast to Figures 2 to 8, Figures 9 to 15 display a P frame of a fast moving video.

Figure 2: Decoded lossless frame of slow motion video

Figure 3: Single slice loss in frame of slow motion video.

Figure 4: Frame of slow motion video decoded using Intra EC method.

Figure 5: Frame of slow motion video decoded using Default EC method.

Figure 6: Frame of slow motion video decoded using MCEC method.

Figure 7: Frame of slow motion video decoded using ZMEC method.

Figure 8: Frame of slow motion video decoded using FFmpeg EC method.

Figure 9: Decoded lossless frame of fast motion video.

Figure 10: Single slice loss in frame of fast motion video.

Figure 11: Frame of fast motion video decoded using Intra EC method.

Figure 12: Frame of fast motion video decoded using Default EC method.

Figure 13: Frame of fast motion video decoded using MCEC method.

Figure 14: Frame of fast motion video decoded using ZMEC method.

Figure 15: Frame of fast motion video decoded using FFmpeg EC method.

From the images above, it is obvious that for slow motion videos, inter-frame error concealment methods (Default EC, MCEC, ZMEC, and FFmpeg) are significantly better perceptually than the intra-frame error concealment method. On the other hand, for fast motion videos, the intra frame error concealment method is better perceptually than the inter-frame error concealment methods.

After decoding all lossy videos with the different error concealment methods, VQM scores were collected according to loss type, thus forming four categories: Single Slice, Whole Frame, Multiple Slices Single Frame, and Multiple Slices Multiple Frames. For each loss type, the five error concealment methods were compared. Figures 16 through 19 show the results for the four different loss types. The y-axis shows the VQM score. In each figure what is plotted is the mean score together with the 95% confidence interval for each particular error concealment method. As mentioned before, lower values are better. A difference of 0.1 points or less between two VQM scores is considered insignificant and thus visually imperceptible by most humans.

Single slice loss is shown in Figure 16. Observing the graph, we can see that for the single slice loss case, MCEC did the best concealment, but in general, all scores are fairly low (good). The 95% confidence intervals for some of the methods are non-overlapping. For example, Default EC is better than FFmpeg and Intra EC because the confidence bars are non-overlapping; but it is not better than ZMEC or MCEC because the confidence bars overlap. Even though the cases where confidence intervals do not overlap are considered statistically significant, they are still visually insignificant because the differences in VQM scores are substantially less than 0.1.

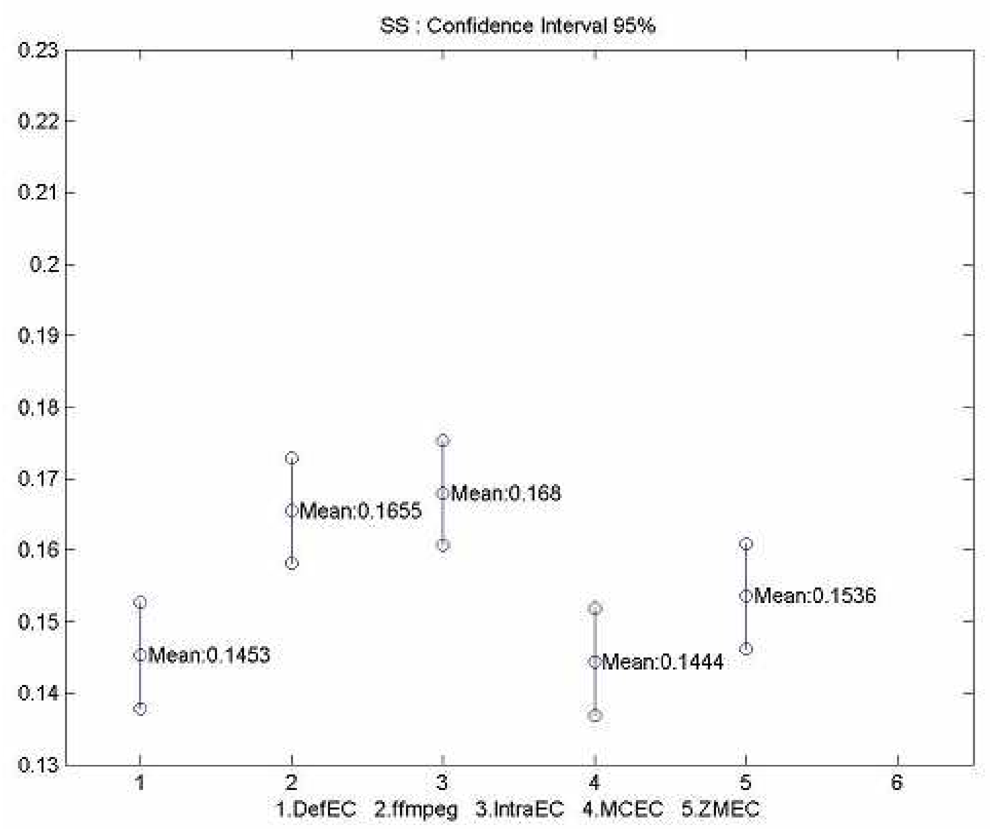

Figure 16: VQM scores showing mean and 95% confidence intervals for the case of Single Slice loss.

Whole frame loss is shown in Figure 17. Due to the fact that in the whole frame loss type an entire frame (except for the first slice) is lost, it is impossible to use the Intra EC since there is not enough information available within the frame to conceal the loss. Therefore, only the four remaining EC algorithms were used.

In the whole frame loss case, DefEC can only use the conceal by copy method since there are no surrounding macroblocks from where to get motion vectors, thus, DefEC simply copies the reference frame, which is the same method as ZMEC. In the graph in Figure 17, we can see that the means of DefEC and ZMEC are identical, as expected for the whole frame case. For the case of MCEC, since only the first slice is available, the second slice uses estimated motion vectors from the first slice of the same frame. For the remaining slices, however, only the reference frame is used, resulting in a method identical to ZMEC for slices three and beyond. Since for the whole frame case, the difference between MCEC and ZMEC is extremely small, their means resulted to be almost identical (0.0002 points), as seen in the graph.

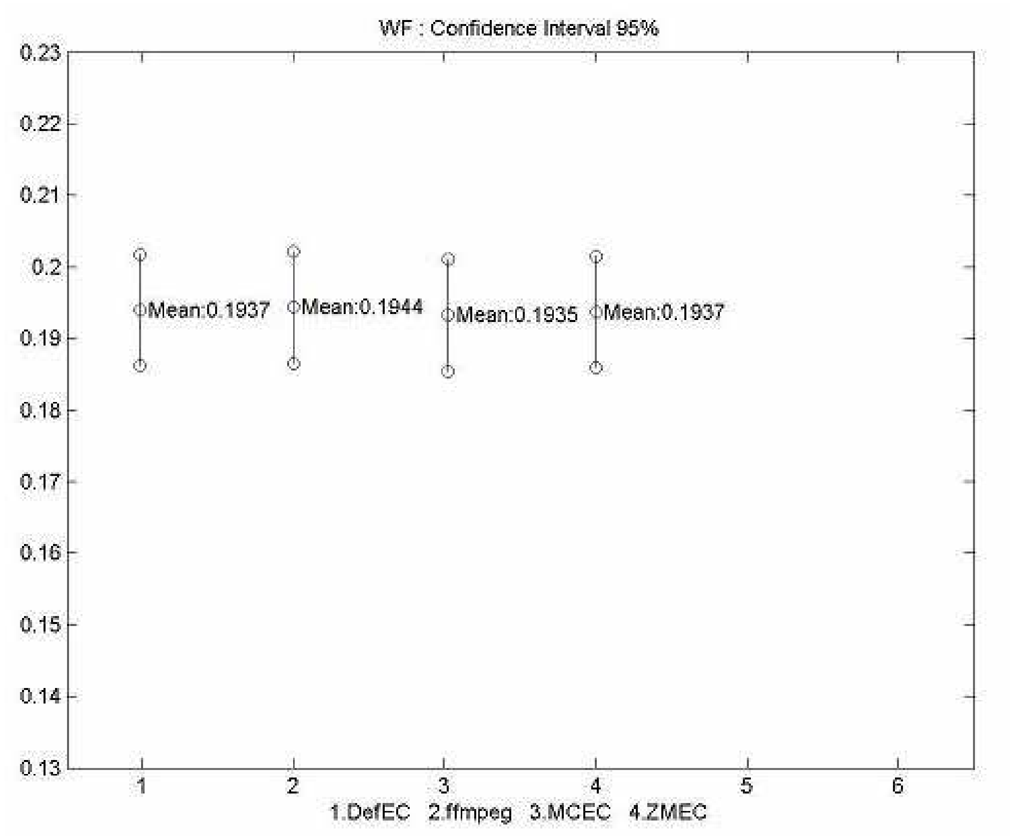

Figure 17: VQM scores showing mean and 95% confidence intervals for the case of Whole Frame loss.

Multiple slices single frame loss is shown in Figure 18. The Intra EC method got the worst score, without even overlapping the 95% confidence interval with any other EC method used for this type of loss. The reason why the Intra EC method did worse than the other methods is that since multiple slices are lost in the same frame, Intra EC used already concealed (imperfect) slices to conceal the loss in other slices resulting in lower quality concealment than when using perfect slices.

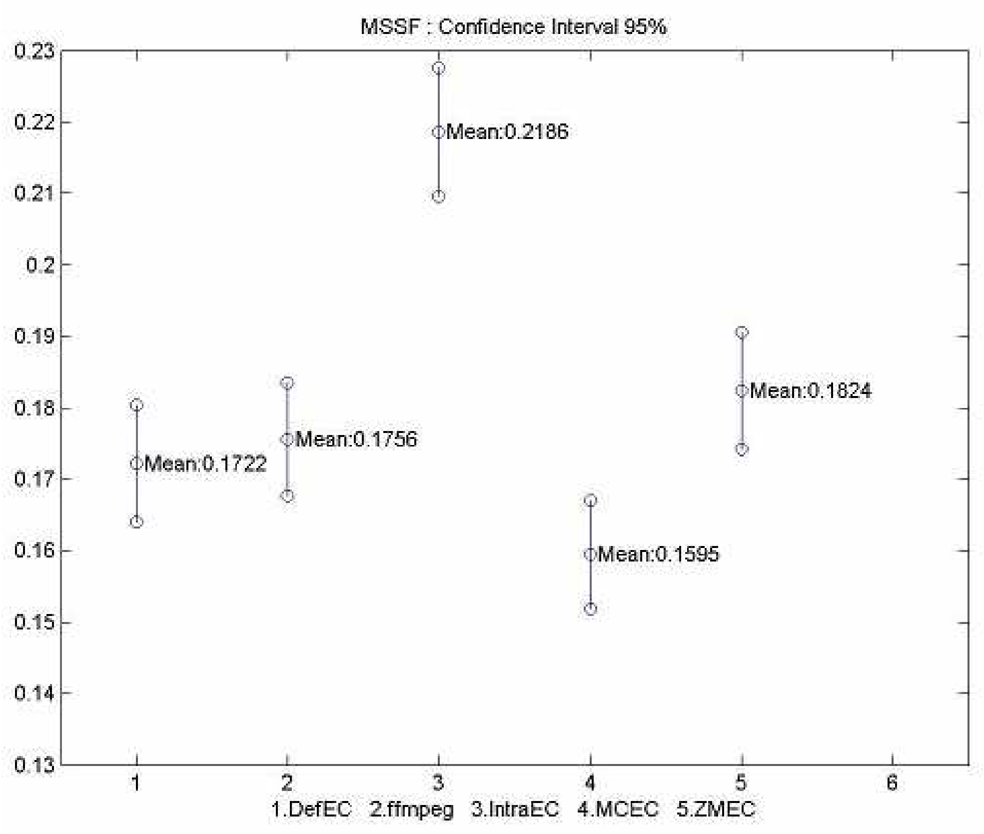

Figure 18: VQM scores showing mean and 95% confidence intervals for the case of Multiple Slices Single Frame loss.

Finally, multiple slices multiple frames loss is shown in Figure 19. Observing the graph, we can see that the FFmpeg method got the worst score, which can be explained by the fact that (for B frames) FFmpeg uses more than one reference frame to conceal the loss, and in this particular type of loss, several frames per GOP are lossy, thus FFmpeg had to use already concealed (imperfect) frames to conceal the loss in other frames resulting in lower quality concealment than when using perfect frames.

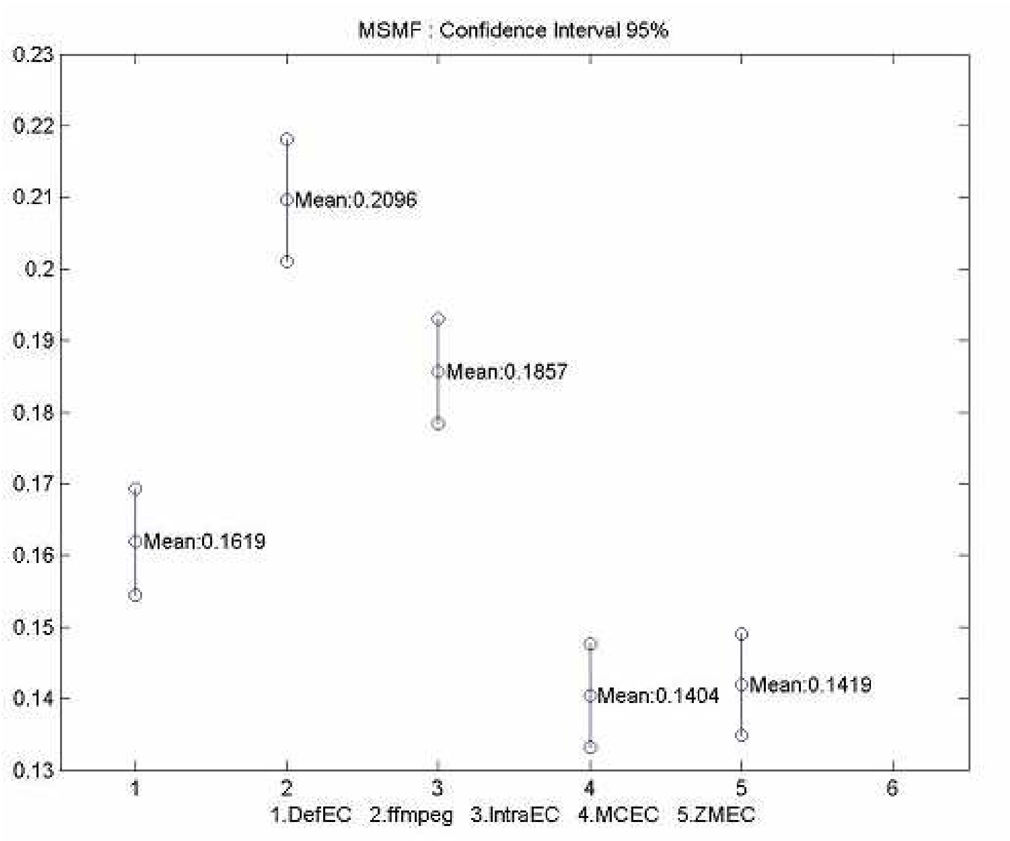

Figure 19: VQM scores showing mean and 95% confidence intervals for the case of Multiple Slices Multiple Frames loss.

It is important to note that in all cases the total difference in VQM scores is very small. There is no difference greater than 0.1 points between error concealment algorithms in each drop type, indicating that the difference in the final videos decoded with the five error concealment algorithms we used in our experiment is hard to perceive by most people.

4. CONCLUSION

From our results, we can conclude that, for the error patterns chosen, concealing the loss using any of the five error concealment methods would give perceptually very similar results. This conclusion suggests that, if either an encoder-based model or a network-based model of packet importance is being used to drop packets, and the model was developed using one of these error concealment algorithms, the visual results would be similar if in fact the decoder were using one of the other methods for error concealment. The results appear to be rather insensitive to the choice of error concealment algorithm suggesting that using the error concealment methods presented in this thesis, we could create a packet-prioritization algorithm at the encoder that is independent of the decoder.

There are several limitations to this study. The most obvious one is that the VQM evaluation assigns a single score to the entire GOP (as opposed to each frame or each slice). So it does not tell us whether, for example, certain single slice losses, with certain characteristics, are noticeably better concealed by one algorithm than by another. What the VQM evaluation tells us is merely that the GOP, when subjected to losses of the single slice variety, and concealed by a certain algorithm, is substantially the same in visual performance on the average as some other algorithm.

A second minor limitation with the study is that, for ease of implementing the VQM comparison, we did not handle the case where the first slice in the frame is dropped. If the first slice were dropped, the decoder would not be aware of the fact that a frame is missing, and the displayed output video would be shorter in length than the original input video. We considered a whole frame drop to be the case where the first slice is received and the rest of the frame is lost.

A third limitation with the study is that the error patterns covered from single slice losses up to whole frame losses (that is, one whole frame per GOP), but not beyond. A different subgroup within our research group is putting emphasis on high loss rates, where many frames might need to be dropped per GOP. It would be of interest to study this case.

In future work, we could try to improve the whole frame drop case so that in practice the entire frame loss includes the first slice. We could also try more types of losses, for example several whole frames per GOP, as indicated above. Finally, we can try to implement other different whole frame loss error concealment algorithms, since the ones used in this experiment resulted in similar behavior for the whole frame loss type.

REFERENCES

[1] S. Cen and P. Cosman, “Decision Trees for Error Concealment in Video Decoding,” *IEEE Transactions on Multimedia*, Vol. 5, No. 1, pp. 1-7, March 2003.

[2] G. M. Schuster, and A. K. Katsaggelos, “Motion Compensated Shape Error Concealment,” *IEEE Transactions Image Processing*, Vol. 15, no. 2, pp. 501–510, February 2006.

[3] W. Kwok and H. Sun, “Multi-directional interpolation for spatial error concealment,” *IEEE Transactions on Consumer Electronics*, vol. 39, No. 3, pp. 455-460, August 1993.

[4] X. Li and M. Orchard: “Novel Sequential Error-Concealment Techniques Using Orientation Adaptive Interpolation”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 10, pp. 857- 864, Oct. 2002.

[5] M.-H. Jo, H.-N. Kim and W.-J. Song, “Hybrid error concealments based on block content,” *IET Image Processing*, Vol. 1, No. 2, pp. 141–148, June 2007.

[6] J. Seiler and A. Kaup, “Adaptive Joint Spatio-Temporal Error Concealment for Video Communication,” *IEEE 10th Workshop on Multimedia and Signal Processing*, pp. 229-234, November 2008

[7] T. Pei-chun Chen, and T. Chen, “Second-generation error concealment for video transport over error-prone channels,” *IEEE Proceedings on Image Processing*, Vol. 1, pp. I-25- I-28, December 2002.

[8] S. Kanumuri, S.G. Subramanian, P.C. Cosman, A.R. Reibman, and V. Vaishampayan, *Predicting H.264 Packet Loss Visibility using a Generalized Linear Model, IEEE International Conference in Image Processing, October 8-11, 2006, Atlanta, GA.* pp. 2245 – 2248, February 2007.

[9] S. Kanumuri, P.C. Cosman, A.R. Reibman, and V. Vaishampayan, “Modeling Packet- Loss Visibility in MPEG-2 Video,” *IEEE Transactions on Multimedia*, vol. 8, pp. 341–355, April 2006.

[10] The website for VQM software. [Online]. Available: http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm.

[11] The website for H.264 HHI JM software. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[12] The website for FFmpeg software. [Online]. Available: http://www.ffmpeg.org.

[13] S. Moni and J. A. Tardif, “Intra-Frame Video Error Concealment,” U.S. Patent 6,697,126, Feb. 24, 2004.

[14] T-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman, A, Reibman, “A Versatile Model for Packet Loss Visibility and its Application to Packet Prioritization,” *IEEE Transactions on Image Processing*, Vol. 19, No. 3, pp. 722–735, February 2010.