Fighting Packet Loss on IP Video Streams

Jose Miguel Piquer
DCC, University of Chile
Miraflores 222, Piso 14
Phone: +56-2-940-7704
Santiago, Chile
jpiquer@dcc.uchile.cl

Felipe Lalanne
DCC, University of Chile
Miraflores 222, Piso 14
Phone: +56-2-940-7704
Santiago, Chile
flalanne@dcc.uchile.cl

Cristian Junge
DCC, University of Chile
Miraflores 222, Piso 14
Phone: +56-2-940-7704
Santiago, Chile
cjunge@dcc.uchile.cl

ABSTRACT

Packet loss is a major consideration factor, especially in real-time applications. Since it cannot be avoided, its effect has to be minimized. We propose an end-to-end approach, analyzing different transmission strategies and error concealment methods to minimize image degradation at the receiver. The proposed system uses pixel shuffling techniques for packet transmission and median filtering concealment techniques for diminishing the impact produced by losses.

Categories and Subject Descriptors

I.3 [Computing Methodologies]: Computer Graphics;
H.4.3 [Information Systems Applications]: Communications Applications—*Computer conferencing, teleconferencing, and videoconferencing*

General Terms

Experimentation, Measurement, Algorithms, Performance

Keywords

Perceptual metrics, error concealment, video coding

1. INTRODUCTION

When a video stream is transmitted over an IP network, packet loss must be taken into account and dealt with. Existing encoders distribute the image information sent on many IP packets which are reassembled at the receiver. To avoid huge bandwidth consumption, the encoders always perform some kind of compression on the stream sacrificing quality to adapt to the available bandwidth. This compression implies that information losses during the transmission are even more visible on the receiver.

Temporal compression, very commonly used by existing encoders works by detecting similarities between consecutive frames and transmitting only the differences. Those differences are usually calculated on macro-blocks (typically

of 8x8 pixels) and the degree of difference determines if a whole macro-block is transmitted or not. Once all the macro-blocks have been checked, those marked for transmission are put on as many IP packets are necessary and then sent to the receiver. As some packets will be lost, some macro-blocks will be missing, generating old blocks to remain longer than needed on the receiver's image. Each macro-blocks is a square clearly noticeable to the human eye, so losses are perceived and annoying even at very low percentages (between 1 and 10 percent). To fight this phenomenon, the encoders re-send a macro-block from time to time, even if it has not been marked for transmission, but this increases the bandwidth consumption and it is impossible to perform this operation too frequently, so the macro-block is clearly perceived by the receiver as an error.

The effort to improve multimedia services over IP is divided between network improvement implementing some level of Quality of Service, and application improvement trying to provide end-to-end intelligence to adapt to the changing network conditions. Our work is concerned with the end-to-end approach, testing different encoding algorithms to improve their resilience to packet loss.

Some work exists on the literature using the end-to-end approach to try to repair packet losses using schemes as Forward Error Correction [2, 4, 5]. The effort in this research is to be able to conceal the losses without adding more information to the stream, but just distributing the information randomized to avoid generating obvious patterns on the receiver. We have concentrated on modifying the differential encoder just on the network sender and receiver, keeping the motion detection and macro-block management of the traditional algorithms. The main idea is to avoid losing complete macro-blocks by shuffling the pixels of a macro-block over a group of IP packets, thus lowering the probability of losing complete blocks. To conceal the pixels that have been lost, some existing work can be used getting the information of its neighbors [6, 7].

We also address the issue of automatically measuring the received video quality [8, 10, 3]. This will allow us to test and compare the results obtained by different configurations of the environment in an objective way. Some existing work in the area [9, 3] is being used as our reference.

2. TRANSMISSION STRATEGIES

In common motion detection, the entire set of 8x8 macro-blocks of the image is read and checked for changes. When a change is detected, then the block is added to an IP packet that will be sent when a certain byte amount is reached or

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

LANC '07, 10-11 October 2007, San José, Costa Rica

Copyright 2007 ACM 978-1-59593-907-4/07/0010 ...\$5.00.

Figure 1: Loss patterns generated by different strategies.

there are no more blocks to read. Since a packet contains more than one block, then the order in which the blocks are read gains relevance. When a packet is lost, if the information is not distributed appropriately, then an entire section of the image could be missed.

To avoid this behavior, different strategies were used to read the blocks: sequential, grid, spiral, and random. The purpose was to determine the strategy that generated the least noticeable patterns even under bursty loss conditions in the received stream.

With the sequential strategy (figure 1(a)), under massive loss conditions, only one side of the image is updated.

Grid strategy (figure 1(b)) distributes consecutive blocks in different packets to avoid the previous effect. This shows some improvement, but still produces too obvious loss patterns.

Spiral strategy (figure 1(c)), tries to favor certain image areas (like the face in video conference), but massive losses generate compact loss patterns in the received image.

The best results are obtained by the random strategy (figure 1(d)), since no recognizable patterns are generated.

Despite global patterns are concealed when the random strategy is used, the quality of the final image is still not satisfactory, because lost macro-blocks are still noticeable by the receptor. This is a consequence of transmitting whole blocks in each packet.

3. PIXEL SHUFFLING

To avoid losing complete blocks of the image, each macro-block was divided into 4 random partitions. To accomplish this task, the concept *virtual-block* is introduced. A virtual-block is a set of N macro-blocks. For each possible virtual-block size, a random pixel distribution is generated in advance, using the same pseudo-random seed and algorithm, in both the sender and the receiver. When a virtual block is full, the image information contained in it, is distributed in N different packets¹ using the previously mentioned distribution. Each packet is given a virtual-block sequence number, which is then used by the receiver to reassemble the image.

¹ $N = 4$ in the experiments.

Figure 2: Received image using pixel shuffling

The results obtained by this method were satisfactory enough. Even though a bursty packet loss can still generate macro-block loss, the probability is much lower than before. Under 30% loss, the scene is still identifiable (figure 2), and no loss patterns are detected.

Another advantage of this method is that, it is possible to determine the effectively lost pixels, just as well as the received ones. This will prove to be useful for error concealment.

4. AUTOMATIC QUALITY MEASUREMENT

To effectively test the results, a simple quality measurement system was implemented. The used approach was to compare each pixel in the received image with its neighbors. If the difference exceeds a certain experimentally obtained threshold, then it is considered an error, and the degradation level is increased.

Different methods of comparison were tested for this purpose. To evaluate the effectiveness of the different methods, the errors detected by each one were plotted in a different window. In order for a method to be useful, the plotted error pattern should be as similar as possible to the pattern in the received image. In particular three approaches were used. Neighbor average, neighbor median and nearest neighbor distance.

The concepts *color distance* and *color modulus* had to be introduced for the implementation. The concepts itself are intuitive enough. Color distance between two pixels is calculated as the distance between two points on an *Euclidean* space, then the following formula is used.

$$(Y_1 - Y_2)^2 + (U_1 - U_2)^2 + (V_1 - V_2)^2 \quad (1)$$

Where Y , U , V are the color components in the YUV color space. Similarly, color modulus is defined as.

$$R^2 + G^2 + B^2 \quad (2)$$

In the neighbor average method, the average color distance to the pixel nearest neighbors is calculated. If the value is bigger than a certain limit, then it is considered an error. The problem with this method is that, when the difference in intensity between the background and the lost pixels is not large enough, then false negatives are produced, as shown in figure 3. In the background of the image there is a clear loss pattern, that is not being detected by the algorithm.

Figure 3: Neighbor average

In the neighbor median method, the median modulus of the pixel nearest neighbors is calculated. If the difference between the current pixel modulus and the median is bigger than a certain threshold, then it is considered an error. Even though every tested algorithm produces a certain amount of false positive errors, this approach produces too many to be ignored. In figure 4, this is clearly shown, as large color differences are plotted.

The nearest neighbor approach is the simplest algorithm. It works by calculating independently the color distance to the nearest north and east pixels: if both values are bigger than an established limit, then the pixel is considered lost. This method proved to have the most effective results on the tests. The figure 5 displays a very small amount of false positives and false negatives.

Using this algorithm, then the degradation of the image is calculated as.

$$\text{Errors}/\text{Width} * \text{Height} \quad (3)$$

5. ERROR CONCEALMENT

Using the pixel shuffling algorithm, even under massive losses (30% - 40%) the image content is still recognizable, but lost information can still be perceived by the receiving user.

To minimize the impact produced by losses, an error concealment strategy was developed. Since effectively lost pixels are known, then it is straightforward to know which pieces of information are needed to be repaired. Four different methods, commonly used in noise reduction were tested: average filter, modulus median filter, probabilistic (histogram) filter and component median filter. In order to appreciate better the results obtained with each different algorithm, a testing application was implemented that simulates pixel loss in a picture and then corrects the errors in the image using each method.

Average filter works by replacing the pixel value with the average color of the nearest neighbors. The images 6, 7, 8 and 9 were simulated using 50% changed blocks and 25% pixel loss per block, which is equivalent to 25% packet loss. Under this condition, the filter (figure 6) shows acceptable results, but errors are never completely concealed, the produced effect is that errors are blurred by the averaging process.

Modulus median filter works by replacing the pixel value with the median modulus value of the nearest neighbors. Under the previously cited conditions, the filter (figure 7)

Red			
Blue	0 - 127	0 - 127	128 - 255
	128 - 255	5	9
		8	2

Table 1: Color discretisation

results are not as good as with the average method. Not only the errors are never completely concealed, but the effect is that the errors are just grouped in denser areas. According to experimental tests, this effect tends to be more dramatic when the pixel loss percentage is increased.

Probabilistic filter works by using the concept of histogram [11]. A 5x5 pixel matrix is considered, centered in the lost pixel, then the color information of the 24 surrounding pixels is discretised in 4 bins according to table 1. The bin with more occurrences is the one that has the biggest probability of containing the lost pixel. The pixel value is replaced by the average of the pixels in the winning bin.

As shown in figure 8, this method presented superior results to the previous techniques, concealing the majority of the errors under the cited conditions.

In component median filter, each RGB colorspace component of the lost pixel is calculated as the median of the respective component of the nearest neighbors. The results are shown in figure 9. Most of the errors are corrected, showing only some residual artifacts that can be safely ignored.

The best results, in terms of perceptual video quality, were obtained by both probabilistic and component median filters, with almost no noticeable difference on the receiving end of the transmission.

6. EXPERIMENTS

Visual comparison between the probabilistic and component median concealment filters showed no significant advantage to any of the methods, then a more profound analysis had to be performed. A stored video sequence was transmitted using different simulated loss conditions (0 - 35%). The automatic quality measure technique, explained previously, was used to obtain a numerical value for the video quality, for both concealment methods.

The comparison results are shown in figure 10. The component median filter, shows a 60% quality improvement with respect to using no concealment method, and approximately 50% improvement in comparison to the probabilistic technique.

Figure 4: Neighbor median

Figure 5: Nearest neighbors distance

7. RESULTS AND DISCUSSION

We incorporated our algorithms into a video streaming application. The results obtained using the error concealment approach, under 25% packet loss conditions, were an improvement of the overall quality of the received video over 60%, according to the measurements.

Packet loss is still a very important factor to consider, especially if considering non-broadband networks, like GPRS, EDGE or UTMS. Our approach, shows that the effects of packet loss can be reduced using different techniques and without the need for information redundancy.

One of the mayor drawbacks of the pixel shuffling technique is the difficulty to incorporate compression algorithms to reduce the bandwidth used by the application, since pixel color correlation is lost. Some work is currently being done in this area, using feedback-based adaptability to the network conditions [1], increasing the delay between packets to adapt to the available bandwidth. In addition to this, we believe that JPEG compression could be used at packet level by separating the luminescence and color components without adding too much processing overhead.

8. ACKNOWLEDGMENTS

The authors would like to thank NIC Chile and Entel PCS for providing the necessary funding to make this research possible.

9. REFERENCES

- [1] N. Farber and B. Girod. Feedback-based error control for mobile video transmission. N. Farber and B. Girod, Feedback-based error control for mobile video transmission, in Proc. IEEE, Special issue on Video for Mobile Multimedia, to be published.
- [2] B. Lamparter, O. Boehrer, W. Effelsberg, and V. Turau. Adaptable forward error correction for multimedia data streams. Technical Report TR-93-009, 1, 1993.
- [3] Nalini Venkatasubramanian and Klara Nahrstedt. An integrated metric for video QoS. In *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*, pages 371–380, New York, NY, USA, 1997. ACM Press.
- [4] K. Park. Afec: an adaptive forward error-correction protocol and its analysis, 1997. K. Park. AFEC: an adaptive forward error-correction protocol and its analysis. Technical Report CSD-TR97 -038, Department of Computer Sciences, Purdue University, 1997.
- [5] K. Park and W. Wang. QoS-Sensitive Transport of Real-Time MPEG Video Using Adaptive Forward Error Correction. In *ICMCS, Vol. 2*, pages 426–432, 1999.
- [6] P. Salama, N. Shroff, E. Coyle, and E. Delp. Error concealment techniques for encoded video streams, 1995. P. Salama, N. B. Shroff, E. J. Coyle, and E. J. Delp, Error Concealment Techniques for Encoded Video Streams, in IEEE Int'l Conf. on Image Processing, vol. 1, pp. 9–12, 1995.
- [7] Y. Wang and Q. Zhu. Error control and concealment for video communication: A review, 1998. Y. Wang and Q.-F. Zhu, Error control and concealment for video communication: A review, Proceedings of the IEEE, vol. 86, no. 5, pp. 974–997, May 1998.

- [8] Z. Wang, A. Bovik, and B. Evans. Blind measurement of blocking artifacts in images, 2000. Z. Wang, A. C. Bovik, B. L. Evans: Blind measurement of blocking artifacts in images. in Proc. ICIP, vol. 3, pp. 981–984, Vancouver, Canada, 2000.
- [9] S. Winkler. Issues in vision modeling for perceptual video quality assessment, 1999. S. Winkler: Issues in vision modeling for perceptual video quality assessment. Signal Processing 78(2), 1999.
- [10] S. Winkler, A. Sharma, and D. McNally. Perceptual video quality and blockiness metrics for multimedia streaming applications, 2001. Stefan Winkler, Animesh Sharma, David McNally: Perceptual video quality and blockiness metrics for multimedia streaming applications. in Proceedings of the International Symposium on Wireless Personal Multimedia Communications, pp. 547–552, Aalborg, Denmark, September 9–12, 2001.
- [11] A. Wrangsjö and H. Knutsson. Histogram filters for noise reduction. In *Proceedings of the SSAB Symposium on Image Analysis*, March 2003.

Figure 6: Average filter results

Figure 7: Median filter results

Figure 8: Probabilistic filter results

Figure 9: Component median filter results

Figure 10: Quality comparison between concealment methods