REPAIR OPTIONS FOR 3-D WIREFRAME MODEL ANIMATION SEQUENCES

S. Varakliotis, S. Hailes

Computer Science Dept.
University College London
Gower Street, London WC1E 6BT, UK
{S.Varakliotis, S.Hailes}@cs.ucl.ac.uk

J. Ostermann

AT&T Labs - Research
200 Laurel Ave South
Middletown, NJ-07748, USA
osterman@research.att.com

ABSTRACT

To date, almost all work within the field of 3-D animation has focussed on raising perceptual appeal to an adequate level. However, it is a non-trivial task to match the requirements of real-time streams to the reality of the Internet. One of the key problems that must be addressed is that of how best to conceal errors in the event of packet loss. Thus we present the results of experiments designed to evaluate the effect of different possible schemes under different conditions of loss. We conclude that frame insertion methods show good performance at low loss rates, but that frame interpolation methods exhibit better performance at both low and high burst loss rates, though at the cost of added delay and complexity. Visualised traces subjectively validate these results, by exhibiting smoother animation and reduced artifacts on the animated wireframe models.

1. INTRODUCTION

Over recent years, the increasing power of end systems has driven the growth in the use of multimedia; many young people cannot remember a time before computers became multimedia capable. In the early 1990s, it became apparent that the likely deployment of high bandwidth networking would give rise to the ability to support voice over IP (VoIP) and video conferencing systems. This expectation has driven research and standardisation efforts in these fields to a respectable level of maturity.

This means that the non-networked media that users have come to expect must now be made to work in a networked form; aside from video and audio, 3-D graphics and animation have the clearest commercial benefits. It is for this reason that, despite the existence of a plethora of file formats and encodings for 3-D data (e.g. VRML), a significantly more efficient compression and animation framework is being sought in the context of MPEG-4 [1].

MPEG-4 is a very general standard aimed at addressing precisely the market outlined above. Within the context of animation, it addresses the 3-D scene compression and delivery through a tool set called Binary Format for Scenes (BIFS) [2]. BIFS is the compressed binary format, in which 3-D scenes are defined, modified (BIFS-Command) and animated (BIFS-Anim) [3]. BIFS is an extension of VRML, which has the added advantage of ensuring backwards compatibility. In addition to BIFS, the Synthetic/Natural Hybrid Coding (SNHC) tools yield reasonably high compression for still textures, face and body animation, and 3-D mesh coding.

In MPEG-4, the two main animation tools, BIFS-Anim and 'face-anim' are based on an adaptive arithmetic encoder that outputs a series of differential changes to scene components. This allows for low delay coding. We will be referring in the rest of this document to our zero-order prediction based coder that arithmetically encodes vertex displacements of animated 3-D meshes that do not suffer topological changes through time, as described in detail in [4]. However, there is considerably more to realising a codec suitable for use over the Internet than this.

In order to design a codec suitable for widespread Internet use, it is just as important to take into account the likely channel characteristics as to consider what happens in the signal processing domain. In fact, it is most unwise to design the latter without having the former in mind. The best-effort Internet is a somewhat hostile environment; one must cope with packet loss, packet reordering and duplication, delay, delay variation (jitter) and even fragmentation. Consequently, network-awareness is very highly desirable. To take a simple example, packet loss on the fixed network is most often associated with congestion at routers and is a relatively common occurrence in the wide area. Unless coding schemes are robust and unless the transmission rate can be adapted to take account of the congestion, perceptually significant problems will arise.

The network problems that arise from the best-effort nature of the Internet have been investigated and addressed over some period of time within the now established audio/video tools, such as vic and rat [5]. These tools use RTP [6] with adaptive playout buffering algorithms to cope with variable delay. However, for the reasons outlined above, congestive loss is a reality, ranging from individual packet loss through burst loss to network outage. As a consequence, online loss parameter estimation methods (notably Bernoulli and 2-state Markov chain models) have been studied with the aim of enabling adaptive applications to monitor the network congestion levels and adapt encoder transmission rates. Given this possibility to estimate loss and monitor congestion, loss concealment or redundancy techniques have been devised [7], along with suitable RTP payload formats, and network-friendly streaming protocols.

In the remainder of this work, we bring to bear the depth of experience at UCL in deploying streaming media repair techniques for the Internet [7], on the problems faced by novel media streams such as 3-D animation data in the same environment. We also study their performance for emerging applications and standards, such as MPEG-4. The suggested repair schemes may operate in tandem with a simple, network-friendly, rate transmission mechanism.

The rest of this paper is organised as follows. In section 2 we discuss options for repair of 3-D animation streams and we de-

0-7803-7304-9/02/$17.00 C2002 IEEE

scribe robust coding methods suitable for network-friendly streaming. In section 3 we outline the simulation experiments with analytic discussion on the results. Section 4 concludes this paper.

2. OPTIONS FOR REPAIR OF ANIMATION STREAMS AND RELATED LITERATURE

Even given the obvious similarity in presentation, it is not obvious that perceptual effect of loss in a stream containing animation of 3-D models should be anything like the effect of loss in a stream containing natural video. This derives from three factors:

- video represents a 2D projection of a 3D scene, whereas the third dimension remains within animation data
- there is a fundamentally different underlying data representation, so a lost packet has a fundamentally different visual effect in each case
- synthetic animation data does not necessarily follow a naturalistic model in terms of chrominance versus luminance. The HVS, however, is tuned to the naturalistic model.

It is probable, therefore, for these reasons that the error sensitivity of animation data is significantly greater than that for video and may well lie closer to that for audio. For obvious reasons, much effort has been expended in developing realistic model rendering, colouring, texturing and animation techniques within the animation community. Likewise, recent 3-D animation coding attempts [4, 8, 9, 10, 11] focus on compression. In all, there is little literature that addresses the problem of error concealment for animation data in the presence of packet loss. Based on the above remarks and past research experience with traditional multimedia coding, we explore some of the possible options for robust animation coding.

2.1. Sender-based repair

A number of audio repair techniques require the participation of the sender of an audio stream to achieve recovery from packet loss. There are two major classes of sender-based techniques: active retransmission and passive channel coding. Channel coding, in turn, is further subdivided into interleaving and forward error correction (FEC). FEC data may be either *media independent*, typically based upon parity operations and Reed-Solomon codes, or *media specific*, based on the properties of a signal. For detailed information on this taxonomy the reader is referred to [7].

Clearly, the above taxonomy could equally well be applied to 3-D animation data. Since possible application areas for animation include those with strong real time requirements such as multiuser virtual worlds and distributed games, active retransmissions are likely to be too costly for animated data in the same way that they are for traditional multimedia streams. Please note that interleaving will also not be investigated in this work, as its potential benefits on generic animation are unclear and will undoubtedly require extensive subjective evaluation tests before useful conclusions may be drawn. Consequently, we only discuss media specific and media independent FEC.

Media independent FEC schemes do not depend on the contents of the packet and consequently can only attempt to ensure that the repair is an exact replacement for a lost packet. Such techniques are easy to implement and cheap to execute but, because they cannot take advantage of media-specific knowledge, they are relatively expensive in terms of bandwidth consumed.

Media specific FEC is an attractive proposition for 3D animation streams, but suffers two drawbacks: it requires increased computational effort relative to media independent FEC, which increases latency; and it requires the modification of animation codecs, which may be hard for 3-D animation codecs that rely on lower level technologies and specialised hardware. In its simplest form, however, media specific FEC can use the same engine to produce both the primary and secondary encodings, at high and low quality respectively.

2.2. Receiver-initiated concealment techniques

Receiver-initiated error concealment methods are desirable when either the sender of a stream is unable to participate in the recovery, or when sender-based recovery schemes fail to correct all loss. Receiver-based methods ameliorate the effects of lost data by attempting to produce something that is perceptually as close to the original as possible, often by taking advantage of the short-term self similarity of the signal. For audio signals, these methods perform well for low to average loss rates (<15%) and for small packets (4-40 ms), since these figures represent audio signals shorter than the typical duration of a phoneme (5-100 ms). As the loss length approaches the length of a phoneme, these techniques are believed to break down, since the listener may miss whole phonemes, a significant perceptual loss. The lack of a direct equivalent to phonemes in visual animation signals means that the error bounds for audio are unlikely to apply in the same way in this case. Thus the true perceptual effect of errors and its relationship to stream content, packet sizes and loss lengths for any given concealment technique can only be determined by experimentation. The first step in this is a re-examination of the generic techniques that are available for receiver-initiated error concealment.

In the case of 3-D animation with free face deformations, which pertains to our codec [4], we identify the following categories for receiver-based concealment. Please note that we assume that each packet contains one or more whole codec frames.

Insertion based schemes repair losses by inserting a fill-in packet. Usually, the fill-in packet can be very simple, such as a repetition of the last successfully received packet. In zero-order prediction-based codecs, similar to the one we developed in [4], the decoded frame usually contains vertex differences between the model's current position and its previous position in 3-D space. The decoder, then, has several options:

1. To *repeat* a previously decoded frame 'as is' (Frame Repetition), in order to avoid erroneous motion on the model's faces. Good candidate frames are those from the short motion history. This is easy to implement, but requires additional memory at the receiver, especially in the case of loss bursts, where concealment could be based on repetition of multiple past frames to reach the animation sequence frame rate.
2. To *predict* the missing frame from the previously received one, thus *extrapolating* the motion of the missing frame (Motion Vectors). Prediction can be linear, in the case of a single frame concealment, or non-linear, for consecutive frame drops. With this scheme the model preserves its motion vectors in an attempt to minimise the perceptual distortion of the animation. This technique is also relatively easy to implement, but on average it requires more buffering and has greater complexity than scheme 1.

Interpolation based schemes use the packets surrounding the loss to produce a replacement for a missing sequence of packets.

Table 1: Summary of repair options of 3-D animation sequences.

Sender-based			Receiver-based			
Retransmission	Channel coding		Insertion		Interpolation	
	Media independent	Media specific	Repetition	Prediction	Tracking curve	Parameter based

These techniques account for the changing characteristics of the animation signal and, therefore, are well suited for longer loss bursts when the animation pattern is mainly directional instead of alternating. The options for repair at the receiver, using interpolation methods, can be categorized as following:

1. The receiver constantly updates a *tracking curve* of the model's animation path. In the event of loss, this curve is used to interpolate the position and predict the velocity of the animated vertices for missing frames, approximating a smoother animation path (Interpolation). For better results, a receiver could trade delay for quality, in order to collect a reasonable set of sampling points along the animation path, and employ non-linear interpolation and optimization techniques. In its proposal for distributed animation in multi-user worlds, the SNHC group of MPEG-4 advocate a similar method in [12] in conjunction with a dead reckoning principle algorithm. Although the proposed animation streams in [12] are not frame-based, the ideas can be applied to our approach for error repair. Such techniques require increased processing power at the receiver and imply playout delays. Thus, they can be well applied to near real-time or stored animation avatars.
2. The receiver can, apply more specific *parameter-based* recovery techniques. In special cases, where the animation exhibits some form of regularity or short-term stationary behaviour in the samples of consecutive frames, it is possible to describe the animation path piecewise using regressive analysis methods on the last received set of samples. A synthesis step during concealment, based on an estimate of motion parameters (e.g. velocity), will generate an approximation of the missing frames. These schemes may be hard to implement for animation and we have no experience yet with regards to performance.

Table 1 sumarises the repair options on 3-D mesh sequences. In the rest of this paper we discuss experiments that show the performances of receiver-based concealment methods.

3. CONCEALMENT FOR VERTEX-BASED ANIMATION

Figure 1 shows three possible scenarios of the animation trajectory. When applying piecewise linear interpolation, as shown in our experiments in §3.1, we can achieve good approximation of the trajectory for relatively short packet loss bursts (commonly occuring in the Internet.) For this scheme to be successful we assume that the principle of *locality of vertex motion* holds true. According to this principle, the motion trajectory is unlikely to change much for a single component (x, y, or z) within a short time interval. To express this in a formal notation, the trajectory functions F_x, F_y, F_z are monotonic within time interval $[T_a..T_b]$. The visual result of non-monotonicity depends on the packetisation scheme and would be seen as artifacts in the form of: (i) *blips* on the rendered surface, if vertices of one sub-mesh are split between different packets, or (ii) *irregular*, *enlarged*, or *shortened* sub-meshes, if all vertices of one sub-mesh are contained in the same lost packet(s). Such artifacts are almost invisible[1] in high frame-rate animation streams.

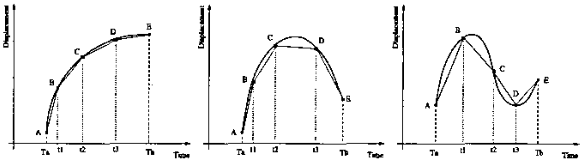

Figure 1: 3 scenarios using linear interpolation in a 3-D animation sequence: (a) monotonic, (b) asymptotic, (c) free-form trajectory.

3.1. Experiment outline and results

In this section, we present the results of experiments we have conducted to study the performance of the repair methods presented in the previous paragraphs. The experiments covered here focus on receiver-initiated techniques referenced in §2.2, whereas sender-based mechanisms are presented in a follow-up document, with a Rate-Distortion framework for Internet-friendly streaming [13].

We are using a number of animated models called *Telly*, *Fred*, *Robot*, *Bounceball*, and *Chicken* (courtesy of AT&T, MPEG-4 SNHC, Seoul National Univ. [9], and Microsoft Research [10].) For brevity we will refer only to Telly, Fred, and Robot in this paper. Telly consists of 800 frames, 9 nodes and 918 moving vertices at 24 Hz, whereas Fred is animated at 15 Hz with 100 frames, 24 nodes, and 651 vertices. Robot is similar to Fred, but consists of 32 nodes and 6680 vertices. All sequences have been encoded with I-frames at various regular positions, namely at multiples of 8, 15, or 30 frames. Fred and Robot contain *dense* animation data (i.e. all of a model's vertices are displaced between consecutive frames by at least one component x, y, or z), as opposed to Telly's *sparse* animation (i.e. only a few vertices change from one frame to the next). We simulate burst packet loss with a 2-state Markov model as described in [14]. Various non-cumulative loss patterns have been generated and tested on the sequences within the packet loss percentage range [0..30]. The graphs presented herein have been generated by averaging experiment results over 50 loss patterns for Fred, and 20 patterns for Telly and Robot. In the remaining text, the PSNR is calculated as the average Euclidean Distance between the original and coded sequences, as in [4]. Finally, we assume that one encoded animation frame fits into one RTP packet with respect to the path MTU.

Figure 2 shows the PSNR achieved after repairing with one of the three proposed methods, namely Frame Repetition, Motion Vectors, and Interpolation, referenced hereafter as FR, MV, and IN, respectively. The graphs show a decreasing trend of PSNR as loss rate increases, highlighting the relative performances of each repair method. The effect of losing I-frame packets is expressed by sudden drops in the average PSNR, in the order of 1-2 dB. This anomaly is due to the non-cumulative loss patterns that result in random I-frame losses as has been explained in detail in [4].

From plots (a)-(c) it is evident that MV performs better than FR in Fred for I-frame frequencies up to 15Hz. At 8Hz the relative performance of MV and FR is already reversed for node EarRight (c), suggesting that the frequency of I-frames bears some significance to the relative performance of the repair techniques. This behaviour is observed in all nodes of our Fred data with a swap in performance at 10Hz for the majority of nodes. Observation of Telly data (b) shows inverted results, with performance swap threshold frequency of 5Hz. In Telly, sparse vertex animation takes

---

[1] Visualised samples of our experiments are available at: `http://www.cs.ucl.ac.uk/staff/S.Varakliotis/icme2002`

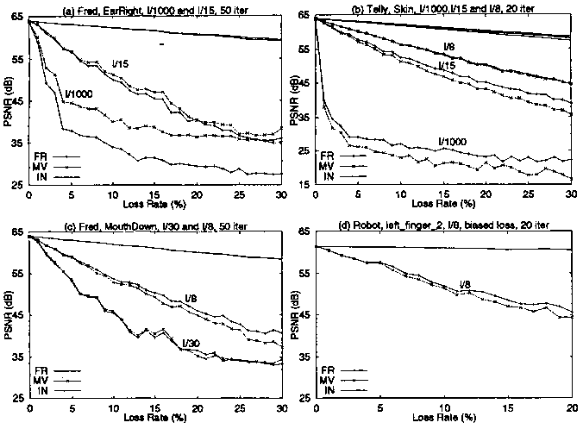

Figure 2: PSNR versus Loss Rate for different sequences, nodes, and I-frame combinations.

place, and different vertices are animated from one frame to the next. Repeating the previous frame results in animating the vertices present in the previous frame only, and FR performs better than MV even in the absence of I-frames. This evidence suggests that FR is more suitable for sparse animations; however, this claim requires further validation. We also observe knees at 4% loss in the PSNR curves of I/1000 in (a) and (b), which validate the intuitive result that lack of I-frames in the encoded sequences sharply deteriorates their animation quality in the event of frame losses.

Plot (d) shows the same experiment performed on Robot for a shorter loss rate range, where the loss patterns have been manually edited (here called *biased* loss) in order to minimise the effects of loss on I-frames. In the range shown, the relative performance of FR and MV is as in Telly, but without observing any threshold.

All plots show that IN outperforms both others. It is, therefore, the preferred method irrespective of the loss pattern or I-frame frequency, but achieves this status at the expense of additional computation and increased latency. It can also be seen in all four graphs that IN hardly exhibits any PSNR anomalies (sudden drops) discussed previously. IN will fail altogether for loss bursts that span a longer packet duration than the maximum frame latency allowed at the receiver, but such bursts are unlikely to occur unless they are the consequence of a disconnection, about which one can do nothing.

4. CONCLUSION

We have surveyed the available options for concealment of 3-D wireframe model animations. We have distinguished sender-based from receiver-based approaches and focused our experiments on the latter. In our experiments with low loss rates and bursty patterns, the suggested receiver-based method based on frame repetition perform better than methods based on motion vectors for sparse vertex animations. The result is reversed in the case of dense animation sequences. Under higher loss rates and regardless of the loss pattern, both methods are outperformed by interpolation, at the expense of complexity and memory buffers.

The concealment methods presented here are by no means an exhaustive list, but they serve as a starting point for further refinement and study. We are further extending our work to develop a Rate-Distortion optimised framework for streaming of 3-D animation data. In our future work we also employ sender-based techniques in a network friendly rate adaptation scheme, so that the encoder respects congested Internet conditions. The combined network and media approach to streaming animation data enables these 3-D wireframe models to be used in novel real-time applications of streamed multimedia, such as video-conferencing, e-commerce avatars, multiuser virtual worlds, and other state-of-the-art applications.

5. REFERENCES

[1] Euee S. Jang, "3-D Animation Coding: its History and Framework," *IEEE International Conference on Multimedia and Expo*, August 2000.

[2] Julien Signès, Yuval Fischer, and Alexandros Eleftheriadis, "MPEG-4's Binary Format for Scene Description," in *Signal Processing: Image Communication*, January 2000, vol. 15, pp. 321–345.

[3] Julien Signès, "BIFS: Combining MPEG-4 media to build rich multimedia services," in *Signal Processing: Image Communication*, January 2000, vol. 15.

[4] Socrates Varakliotis, Jörn Ostermann, and Vicky Hardman, "Coding of animated 3-D wireframe models for Internet streaming applications," *IEEE International Conference on Multimedia and Expo*, August 2001.

[5] http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/ and http://www-nrg.ee.lbl.gov/vic/, "VIC and RAT," 1995-2000.

[6] Henning Schulzrinne, Steve Casner, Ron Frederick, and Van Jacobson, *RTP: A Transport Protocol for Real Time Applications*, RFC 1889, IETF, January 1996.

[7] Colin Perkins, Orion Hodson, and Vicky Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, September 1998.

[8] Jeong-Hwan Ahn, Chang-Su Kim, C.-C. Jay Kuo, and Yo-Sung Ho, "Motion-compensated compression of 3-D animation models," *IEE Electronics Letters*, vol. 37, no. 24, pp. 1445–1446, November 2001.

[9] Jeong-Hyu Yang, Chang-Su Kim, and Sang-Uk Lee, "Compression of 3D triangle mesh sequences," in *Proc. IEEE Workshop on Multimedia Signal Processing (MMSP-01)*, Cannes, France, IEEE 2001, pp. 181–186.

[10] Jerome Lengyel, "Compression of time-dependent geometry," in *ACM Symposium on Interactive 3-D graphics*, Atlanta, August 1999.

[11] Marc Alexa and Wolfgang Müller, "Representing animations by principal components," *Computer Graphics Forum*, vol. 19, no. 3, pp. ??–??, 2000.

[12] ISO/IEC JTC1/SC29/WG11, "N4415: Systems AMD 4 PDAM (AFX+MUW)," February 2002.

[13] Socrates Varakliotis, Stephen Hailes, and Jörn Ostermann, "Animated 3-D wireframe model streaming in a Rate-Distortion framework," Tech. Rep., UCL-CS, February 2002, (In preparation).

[14] Andrea Basso, Socrates Varakliotis, Roberto Castagno, and Florin Lohan, "Transport of MPEG-4 over IP/RTP," in *European Trans. on Telecommunications: special issue on Packet Video 2000 workshop*, May - June 2001, vol. 12:(3).