Real-Time Motion-Based Frame Estimation in Video Lossy Transmission

Sherif G. Aly and Abdou Youssef
The George Washington University
Department of Computer Science
801 22nd St. NW
Washington, DC 20052
(202) 994-6569
{sherif,youssef}@seas.gwu.edu

Abstract

In movie transmission, video frames are subject to loss due to noise and/or congestion. The loss of video frames could cause a loss of synchronization between the audio and video streams. If not corrected, this cumulative loss can seriously degrade the motion picture's quality beyond viewers' tolerance. In this paper, we initially study and classify the effect of audio-video de-synchronization. Afterwards, we develop and study motion-based techniques for estimating the lost frames using the existing received frames, without the need for retransmissions or error control information. The estimated frames are injected at their appropriate locations in the movie stream. The objective is to bring back the synchronization within the tolerance level of viewers, while attempting to find a very close estimation to the original frames at a suitable computation cost.

Keywords

Synchronization, frame-estimation, motion-estimation, tolerance, frame loss.

1. Introduction

Transmitting multimedia contents over networks is becoming practical and prevalent owing to increasing bandwidth and better compression. Multimedia contents include video, audio, both audio-video, and presentations. Under varying network conditions, losses do occur in the movies being transmitted, especially in video rather than audio. The loss of video frames might cause a loss of synchronization between the audio signals and the video frames.

The effect of such video loss is inconsistency between the audio and the video. In other words, the display of the video frames might precede the playing of the audio stream. If synchronization is not restored again, the cumulative effect of this synchronization degrades the movie's quality beyond viewers' tolerance.

Some encoding schemes such as MPEG bundle the audio and the video streams into one stream, namely a systems stream with its embedded time stamping synchronization mechanisms [1]. Nevertheless, it is not always the case that one video stream has to be associated with only one audio stream. A movie could be transmitted for example with one video stream, and many audio streams. The video could be transmitted as one stream, while several audio streams, each in a different language, could be transmitted alongside the underlying video stream.

Still, in other situations when transmitting over low bandwidth channels, the movies' high bandwidth requirements necessitate the separation of the video and audio streams, so each stream can be routed independently to offload certain channels. Although time stamping might be useful to restore synchronization between the audio and the video streams, the mechanism does not contribute to the restoration of any video loss that happened during transmission.

Other synchronization correction techniques include protocol based approaches [2], temporal models for synchronization [3], protocol architecture [4], relative time stamping [5], and transformation based error concealment [6].

Motion estimation, on the other hand, has been a vital area of research, especially in image recognition and compression. Much research has been developed to estimate motion between frame sequences [9-17]. Nevertheless, when using motion estimation to reconstruct lost frames, and to restore synchronization between audio

0-7695-0942-8/01 $10.00 © 2001 IEEE

and video streams, one encounters many time and resource limitations.

Many Internet applications providing video and audio streams to PC users have to operate in a limited resource environment. The luxury of time and abundant resources is not present. We cannot therefore use many of the previously developed time consuming motion tracking techniques. We develop a simple yet efficient motion tracking technique to estimate lost frames and to ultimately restore synchronization between transmitted audio and video streams.

For our purpose, we will concentrate on the loss in the video stream. In this paper we investigate human tolerance towards audio-video synchronization loss, and establish a corresponding classification of such tolerance. Then we examine techniques, including motion-tracking frame-estimation in particular, that estimate to a high degree of resemblance the lost video frames, and preserve both temporal and spatial synchronization.

2. Classification of the Effect Of Audio-Video Synchronization Loss on Viewers

In this section, we study and classify the effect of the loss of synchronization on viewers' tolerance due to either contiguous or non-contiguous loss of video frames.

We found by experimenting on human viewers that both the contiguous loss and the non-contiguous loss of the same number of frames have ultimately the same de-synchronization effect on viewers. Nevertheless, contiguous loss caused the synchronization problem to appear earlier, as expected.

We also found that the synchronization loss starts to become apparent as soon as the cumulative frame loss reaches five. This is when viewers start noticing incompatibility between sound and picture. Clearly, the more the loss, the less the tolerance. We classify the effect of synchronization loss on viewers as *highly acceptable*, *acceptable*, *fair*, *annoying*, and *completely unacceptable*. Figure 1 shows our findings.

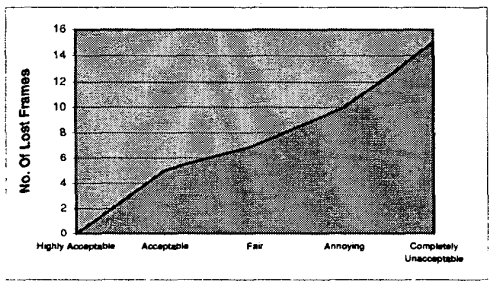

Figure 1 Tolerance of Non-Synchronization

Several approaches could be applied to remedy the synchronization problem. One approach is to compensate for all the lost frames at the receiver, by estimating them. In situations where this is too costly in time and/or memory hardware, it would be best to attempt to compensate for as many lost video frames as possible, and thus bring up the synchronization to a higher tolerance level such as from acceptable to highly acceptable, or from fair to acceptable.

Another approach, which is preventative, can be applied at the sender as opposed to the receiver. It adds extra frames to the movie video stream before it is transmitted. Since it was previously found that the loss of five frames does not affect much the quality of the viewers' perception, as shown in Figure 2, we can add five frames to the video stream before it is transmitted. The number of video frames added to the stream will still maintain the video and audio streams within a highly acceptable tolerance level. In this case, we can afford to lose up to ten more frames during transmission while still maintaining a highly acceptable synchronization level.

A third approach works by processing the audio stream itself. If moments of silence exist in the audio stream, it is possible to remove those moments of silence from the audio stream in order to bring back the synchronization between the video and the audio to a more tolerable level.

In this paper, we investigate the first approach, which is the full estimation of all the lost video frames, thus bringing the synchronization level back to what it was before the transmission of the streaming movie. The other approaches are the subject of ongoing research.

3. Prior Research for the Estimation of Missing Frames

In similar research, we had previously developed and evaluated four different frame estimation techniques: one-way frame duplication, two-way frame duplication, linear interpolation, and quadratic interpolation. For the purpose of illustration, we will assume that frames x, and x+k have been received and decoded, and are resident in the buffers waiting to be displayed, and that frames x+1, x+2... x+k-1 have all been lost. Our objective is to attempt to quickly estimate all those lost frames, and to add them in the appropriate locations.

3.1 One-way and Two-way Duplication

With one-way frame duplication the estimated frames x+1, x+2, x+k-1 are identical to frame x. This will cause a freezing effect to the video stream. When k is not very large, the freezing is more like jerkiness in motion, but is often unnoticed. With larger k, however, the freezing and jerkiness become noticeable and possibly unacceptable.

With two-way frame duplication, frames are duplicated from both frames x and x+k equally. Thus, frames x, x+1... x+$\lfloor k/2 \rfloor$ are made identical to frame x,

while frames x+⌊k/2⌋+1, x+⌊k/2⌋+2... x+k-1 are made identical to frame x+k.

Two-way frame duplication causes a two-part freezing effect in the video stream interrupted in the middle by a sudden jerkiness. This sudden jerkiness seems to be more noticeable and annoying than the homogeneous freezing in one way duplication.

3.2 Linear Interpolation

Given frames x and x+k at the receiver, and all the frames in between lost, this method estimates the pixels of the missing frames as linear interpolations of the corresponding pixels in frames x and x+k. Specifically, if we refer to the value of pixel (i,j), in frame t as $V_{ij}(t)$, the estimated pixel value $V_{ij}(t)$ for t=x+1, x+2, ..., x+k-1 is

$$V_{ij}(t) = \left(\frac{V_{ij}(x+k) - V_{ij}(x)}{k} \right)(t-x) + V_{ij}(x)$$

3.3 Quadratic Interpolation

With quadratic interpolation, pixel values for a specific pixel location in the existing received frames are used to create a parabolic curve that best fits the existing pixel values. Least-squares data fitting is used for such curve generation, using three or more existing frames surrounding the missing frames. For each pixel location in the frames, one such curve is generated. The corresponding pixel values in the missing frames are thus fit onto the generated curve as a means of estimation.

These three techniques work well in very slow motion movies, but not in fast motion or when there are long runs of lost frames. Motion-based approaches would be preferable. This is addressed next.

4. Motion Tracking

Motion tracking between two images is the process by which portions of the first image are mapped to existing portions of the second image. It would thus be known to a certain degree of certainty based on quantitative measures that a given portion of the first image has moved to another location in the second image.

We use the concept of motion tracking to estimate motion between existing frames in a movie stream, and hence to estimate lost frames in between.

Given a sequence of frames with several frames lost or corrupted in the middle, we use the two surrounding frames of the lost sequence to estimate the motion of blocks between frames. The locations of the objects in lost frames are linear interpolations of the block motion as shown in Figure 2.

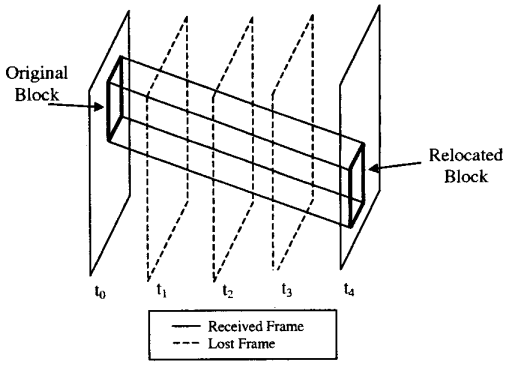

Figure 2 Motion Tracking for Frame Estimation

We divide the frames into blocks of certain dimensions. For our purpose, we started experimenting with blocks of sizes similar to the block sizes used in MPEG. We divided the frames into 16x16 blocks, and experimented on those. Furthermore, we generalized and experimented with other block sizes: 64x32, 32x32, 32x16, 16x16, 16x8, and 8x8 pixel blocks.

The calculation of motion between two frames x and x+k could be very tricky. The accuracy of the calculation of the motion vectors between frames x and x+k is very sensitive to issues like brightness changes. If a portion of an image x is brighter than the same portion in image x+k, the accuracy of motion tracking could be seriously jeopardized.

Thus, given our frames in the usual RGB model, we convert them to the corresponding luminance/chrominance model. The luminance of an image represents the brightness, or the intensity of light. The chrominance, on the other hand, represents the color part. Human vision is much more sensitive to luminance than to chrominance.

What we are especially interested in is not the colors, but rather the major features of the image itself. We are more interested in the luminance as opposed to the chrominance portion. Even if a scene were to have an abundance of light, and another scene having dimmed light, we would still want to capture the motion of the blocks in the scene based on the features of the image, and not based on the colors.

4.1 The Luminance/Chrominance Model

Instead of having RGB components, we would thus have a Y, C_b, and C_r components of the image. The Y forms the luminance component, and the C_b, and C_r form the chrominance component. The luminance/chrominance model could be derived from the existing RGB model by using the following formulas, which the TV community has derived experimentally:

$$Y = 0.299R + 0.587G + 0.114B$$

$$C_b = -0.1687R - 0.3313G + 0.5B + 128$$

$$C_r = 0.5R - 0.4187G - 0.0813B + 128$$

It is worthy to note that the luminance itself does not capture all the brightness of any given image. The model itself as criticized in [9] is derived from non-linear pre-distorted RGB signals. Because of such non-linear pre-distortion, also known as the gamma correction, an amount of luminance information is carried along with the two chrominance signal components. The authors in [9] present a technique by which such defect could be remedied.

4.2 Normalization

After converting the existing RGB model to the luminance/chrominance model, we will use only the luminance portion for motion tracking.

In order to get rid of any additive luminance variations between frames, we mean-normalize the luminance model. That is, we subtract from each frame the pixel mean, so that the frame becomes of mean zero. Similarly, to get rid of multiplicative luminance variations, we variance-normalize the model. Based on experimental calculations, the normalization process helped to significantly enhance motion tracking.

4.3 Block Motion Scenarios

To calculate the motion vectors, we have to consider three scenarios for block motion amongst frames:

1. *Persistent Blocks*: such types of blocks exist in both the source frame and the destination frame, but have merely been displaced from the source frame to the destination frame. Examples include the displacement of objects within a frame.
2. *Disappearing Blocks*: such types of blocks exist in the source frame, but do not exist in the destination frame. In other words, they disappeared out of the boundaries of the frame somewhere along the way from the source frame to the destination frame. Examples include the motion of objects out of a frame's view area.
3. *Emerging Blocks*: Such types of blocks do not exist in the source frame, but instead exist in the destination frame. In other words, they have emerged in the frame somewhere along the way from source to destination. Examples include the motion of objects inside a frame's view area.

Our current research primarily focuses on motion tracking for persistent blocks. Further research will enhance the tracking of disappearing and emerging blocks.

4.4 Motion Vector Calculation

After performing luminance calculations, and mean and variance normalization, we then calculate the motion vectors. Frame x is subdivided into equal size blocks. The motion of each block is tracked in frame x+k by performing mean square error minimization between the source block in frame x, and candidate blocks in frame x+k.

Then comes the issue of which blocks in the destination frame are candidates for being displaced blocks from the source frame. A simple, but highly inefficient and inaccurate way, considers all blocks in the destination frame as candidates. Nevertheless, this is extremely time consuming and inaccurate.

The search domain has to be limited in a way that accommodates the presence of different types of movies that could have different inertia for blocks. Blocks in action movies for example can move at a higher speed than blocks in news broadcasts or video conferencing.

Obviously, it is very unlikely that a block in frame x would be highly displaced in frame x+k. We limit the potential candidates in frame x+k to the slightly displaced blocks around the source block in frame x. Based on this mode of operation, we only search corresponding neighboring blocks in frame x+k to create the motion vectors.

The degree of displacement is a parameter in our system that is modified to accommodate for sensitivities in motion in different types of movies, or even different types of scenes.

After creating the block motion vectors between frames x, and x+k, the vectors are then used to estimate the missing frames in the middle. The position of each block in the missing frames between frames x, and x+k is created as a linear displacement from the corresponding position originating from frame x to frame x+k.

4.5 "Hole" Artifacts

The linear displacement of blocks would create certain empty pixel locations in the estimated frames. The overlapping of blocks in the estimated frames causes "hole" artifacts to appear.

Filters are applied to the estimated frame to remedy the presence of the artifacts. We thus apply basic averaging filters to the generated image to remedy the artifacts appearing due to block displacement. Median filters are also candidates for application. Furthermore, varying the block sizes has a great impact on the amount of artifacts as will be demonstrated in the performance evaluation section.

Although the process of motion tracking itself is more time-consuming than previous techniques, the process is highly and naturally parallelizable if it were to be used in commercial applications. The performance results are shown next.

5. Performance Evaluation

We conducted experiments that involved videos chosen to have clear conversations with clear lip motion, and transitions of scenes between one person and another.

The following data was gathered from a video that had twenty induced frame losses. All twenty frames were estimated using the motion tracking technique presented in the previous section. We experimented on a multitude of block sizes in motion tracking, the results for six of those will be presented in this paper, namely 64x32, 32x32, 32x16, 16x16, 16x8, and 8x8 pixel blocks.

Both subjective and objective evaluations were performed on the achieved results. Subjective evaluations were performed by displaying the videos and sequences of the estimated frames for each technique to viewers. Nevertheless, for the purpose of presentation in the paper, a sequence of estimated frames would be presented for each of the techniques mentioned above.

Objective evaluations include two different ways for the signal-to-noise-ratio (SNR) computations. The first computes a global SNR where all the estimated frames are collectively treated as a single set of data. The second computes the signal to noise ratio for each frame individually.

Figure 3 shows the SNR computations for the estimated frames using motion tracking. Different charts for block sizes of 64x32, 32x32, 32x16, 16x16, 16x8, and 8x8 pixels have been generated. The SNR computations frame wise were very similar, and thus only one chart will be displayed below to illustrate the general shape of the curve.

It is worthy to note that although the frame-wise SNR computations for the different block sizes gave very similar results, the visual quality of the different estimated frames was highly varying. The images will thus be displayed later on to demonstrate that objective SNR computations alone are not sufficient for the judging process. The subjective visual evaluations are very important alongside with the objective evaluations.

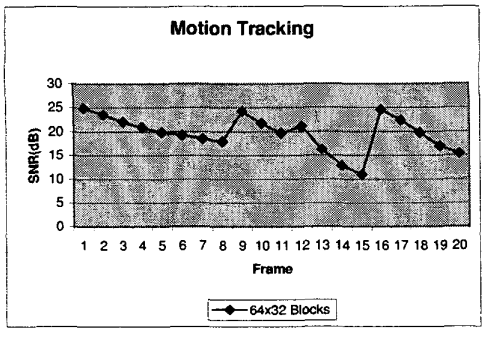

Figure 3 Frame-wise SNR for Motion Tracking

Figure 4 demonstrates the global SNR comparing all six different block sizes used in the experiments. Global SNR computations treat all estimated frames in a video stream as a single data set. A global mean square error is calculated, and hence a global SNR.

Surprisingly, the initial choice of a 16x16-block size in conformance to what MPEG uses for its operation didn't yield the best results with motion tracking.

Overall, the usage of 64x32 blocks yielded the best objective and subjective results. As shown in Figure 4, the global SNR computations for motion tracking show that 64x32 pixel block sizes yield the best results. The subjective visual results will be demonstrated later.

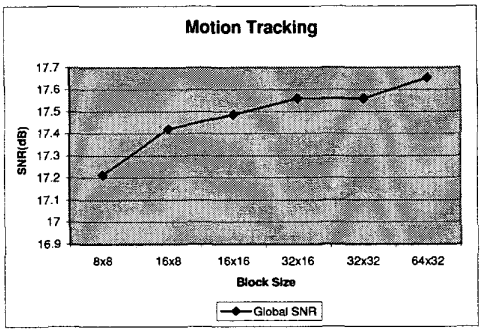

Figure 4 Global SNR for Motion Tracking

For subjective evaluation, we first display the original videos without any synchronization problems, we indicate which frames are "to be lost" and estimated and then we display the estimated videos.

Figure 5 shows the original sequence of frames that have been decoded at the receiver; the three middle frames will be lost and estimated.

Figure 6 to Figure 8 demonstrate the same sequence of frames, but this time, motion tracking is utilized to estimate the three middle frames.

Note that only some of the estimated frame sequences for the selected block sizes are displayed. The difference between them may not be apparent to the casual observer. Some of them yield better subjective results than others. Nevertheless, all results will then be zoomed and displayed to demonstrate both bad block sizes and good block sizes to used. The block size of choice will thus be identified.

Note that the artifacts are of the form of dark spots of pixels in the reconstructed images. They indicate displacement of blocks. Observe that the lower the block sizes, the more the artifacts appeared in the estimated images. The images will be zoomed upon later on to magnify the artifacts to the readers.

Figure 5 Original Frames, the Middle Three to be Lost and Estimated

Figure 6 Estimated Frame Sequence using Motion Tracking with 8x8 Blocks

Figure 7 Estimated Frame Sequence using Motion Tracking with 32x16 Blocks

Figure 8 Estimated Frame Sequence using Motion Tracking with 64x32 Blocks

Although the reconstructed images might look the same for a casual observer, it is worth to note that the block size selection has a great impact on the amount of artifacts in the reconstructed images. The following images will magnify one of the reconstructed images from all six block sizes experimented upon, and will demonstrate both bad and good selections of block sizes. Please note the decrease in artifacts as the block size increases.

Figure 9 Original Image

Figure 10 Reconstructed Image using Motion Tracking with 8x8 Blocks

Figure 11 Reconstructed Image using Motion Tracking with 16x8 Blocks

Figure 12 Reconstructed Image using Motion Tracking with 16x16 Blocks

Figure 13 Reconstructed Image using Motion Tracking with 32x16 Blocks

Figure 14 Reconstructed Image using Motion Tracking with 32x32 Blocks

Figure 15 Reconstructed Image using Motion Tracking with 64x32 Blocks

It was our decision to select motion tracking with block sizes of 64x32 due to their higher visual quality, and their faster estimation. Nevertheless, for more fine grain motion tracking, 32x32 blocks could be used. It was found unadvisable to decrease the block sizes below 32x16 because of the corresponding significant increase in artifacts due to block motion estimation using motion tracking.

6. Conclusions and Future Work

It is possible to remedy

video loss and to restore synchronization between video and audio streams via the quick estimated reconstruction of lost video frames, and their injection in the appropriate locations.

Initially, a study of human tolerance to the loss of synchronization caused by the loss of video frames was performed. A classification of such tolerance was then established. We previously considered four estimation techniques, namely, one-way duplication, two-way duplication, linear interpolation, and quadratic interpolation. We then investigated the utilization of motion tracking based frame estimation as presented in this paper. The studies done within the scope of this research focused on restoring full synchronization between the video and audio streams. The lost frames were estimated using existing received frames only, and without the existence of any further data.

Both objective and subjective evaluations were performed on the estimated frames. It was found that although the usage of different block sizes in motion tracking yielded very similar SNR results, the subjective evaluation of the different estimated images was very different. The presence of "hole" artifacts as presented in this paper seriously impaired the usage of smaller block sizes. A 64x32-block size was found to give the best visual quality.

We are currently investigating the enhancement of motion tracking for disappearing and emerging blocks. Furthermore, we are looking into the usage of filters to enhance the generated image. It can be argued that as the restoration techniques become more elaborate, they incur such long delays and buffering as to make retransmission a preferable solution. While this is true in many applications, the restoration techniques are preferable in low-bandwidth and/or high transmission cost situations. Furthermore, motion tracking is a highly parallelizable problem by its nature, and could be effectively utilized in commercial applications.

7. References

[1] Haskell, Barry G., Atul Puri, & Arun N. Netravali. Digital Video: An Introduction to MPEG-2. New York: Chapman & Hall, 1997.

[2] Ehley, Lynnae, Borko Furht, & Mohammad Ilyas. *Evaluation of Multimedia Synchronization Techniques*, Proceedings of the International Conference on Multimedia Computing and Systems, 1994.

[3] Wahl, Thomas, & Kurt Rothermel. *Representing Time in Multimedia Systems*, Proceedings of the International Conference on Multimedia Computing and Systems, 1994.

[4] Naik, Khsirasagar. Specification and Synthesis of a Multimedia Synchronizer, Proceedings of the International Conference on Multimedia Computing and Systems, 1994.

[5] Son, Sang H., & Nipun Agarwal. Synchronization of Temporal Constructs in Distributed Multimedia Systems with Controlled Accuracy, Proceedings of the International Conference on Multimedia Computing and Systems, 1994.

[6] Wah, Benjamin W., and Xiao Su. Streaming Video with Transformation-Based Error Concealment and Reconstruction, Proceedings of the IEEE conference on Mutlimedia Computing and Systems, 1999.

[7] Rhee, Injong. Retransmission-Based Error Control for Interactive Video Applications over the Internet, Proceedings of the IEEE conference on Mutlimedia Computing and Systems, 1999.

[8] http://www.uni-duisburg.de/FB9/NGA/mitarbei/bruck/komp en/kompen.htm

[9] Tomasi, Carlo. *Pictures and Trails: a New Framework for the Computation of Shape and Motion from Perspective Image Sequences.* IEEE Conference on Computer Vision and Pattern Recognition, June 1994.

[10] Conklin, G., and Hemami, S. *Multi-Resolution Motion Estimation.* Proceedings of ICASSP '97, April 1997.

[11] Aaron, D., and Hemami, S. *Dense Motion Field Reduction for Motion Estimation.* Proceedings of Asilomar Conference on Signals, Systems, and Computers, November, 1998.

[12] Yang, Y., and Hemami, S. *Rate-Constrained Motion Estimation and Perceptual Coding.* Proceedings of the IEEE Conference on Image Processing, October, 1997.

[13] Ruiz, V. et Al. *An 8x8-Block Based Motion Estimation Using Kalman Filter.* Proceedings of the ICIP97, 1997.

[14] Bennamoun, M. *Application of Time-Frequency Signal Analysis to Motion Estimation.* Proceedings of the ICIP97, 1997.

[15] Yoshida, T. et. Al. *Block Matching Motion Estimation Using Block Integration Based on Reliability Metric.* Proceedings of the ICIP97, 1997.

[16] Fan, C., and Namazi, N. *Simultaneous Motion Estimation and Filtering of Image Sequences.* Proceedings of the ICIP97, 1997.

[17] Magarey, J. et. Al. *Optimal Schemes for Motion Estimation Using Color Image Sequences.* Proceedings of the ICIP97, 1997.