



Synchronization-Sensitive Frame Estimation: Video Quality Enhancement

SHERIF G. ALY
ABDOU YOUSSEF

sherif@seas.gwu.edu
youssef@seas.gwu.edu

Department of Computer Science, The George Washington University, Washington, DC, 20052, USA

Abstract. Network transmission is liable to errors and data loss. In movie transmission, packets of video frames are subject to loss or even explicit elimination for many reasons including congestion handling and the achievement of higher compression. Not only does the loss of video frames cause significant reduction in video quality, but it could also 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 study and classify the effect of audio-video de-synchronization. Afterwards, we develop and examine the performance and appropriateness of the application of many client-based techniques in the estimation of 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 to restore the loss. The objective is to enhance video quality by finding a very close estimate to the original frames at a suitable computation cost, and to contribute to the restoration of synchronization within the tolerance level of viewers.

Keywords: frame estimation, synchronization, motion-estimation, interpolation, hybrid

1. Introduction

Transmitting multimedia contents over networks is becoming practical and prevalent owing to increasing bandwidth and better compression. The Motion Picture Expert Group (MPEG) encoding schemes, amongst others, have notably been used as standards for multimedia applications. Video, audio, both video and audio, and presentations are but part of many such applications. When transmitting both video and audio at the same time, as in movies, certain measures have to be taken to ensure that the video components are synchronized with the audio components, and such standards adopt many schemes by which synchronization is preserved.

Under varying degrees of network reliability, losses do occur in the movies being transmitted, especially in video rather than audio because of its significant size compared to the corresponding audio portion. The loss of video frame packets can seriously degrade video quality beyond viewer tolerance, and might contribute to the loss of synchronization between the audio signals and the video frames.

The effect of such video loss is loss of consistency 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.

It is not always the case that one video stream has to be associated with only one audio stream. A movie could be transmitted with one video stream, and many audio streams. The

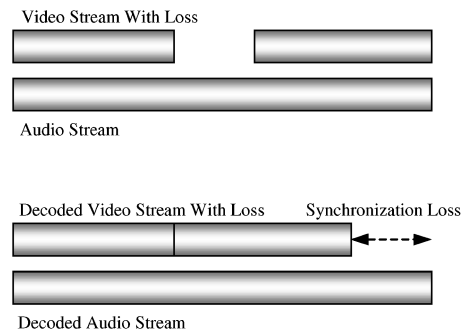


Figure 1. Synchronization problems due to video loss.

video could be transmitted as one stream, while several audio streams, each in a different language, could be transmitted alongside the underlying video stream. 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 [10].

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 [8], temporal models for synchronization [18], protocol architecture [12], relative time stamping [15], and transformation based error concealment [17].

For our purpose, we will concentrate on the more prevalent video stream loss. When the packets making up the video stream are lost, the corresponding frames made up by such packets are either partially or fully absent at the decoding end. At the receiving end, if synchronization mechanisms are not adopted, the decoded video stream is shorter than the originally transmitted video stream. Given decoded video and audio streams of disparate lengths, synchronization problems start showing up between the video and the audio as shown in figure 1.

In this paper we investigate human tolerance towards audio-video synchronization loss, and establish a corresponding classification of such tolerance. Then we examine the appropriateness of the application of several techniques and their combinations that estimate to a high degree of resemblance the lost video frames, and preserve both temporal and spatial synchronization. More importantly, we develop a hybrid system to best utilize the available techniques. The objective is to enhance video quality and to contribute to bringing back the synchronization to a more tolerable level, based on the classification previously mentioned.

2. Classification of the effect of audio-video synchronization loss on viewers

We studied and classified the effect of the loss of synchronization on viewers' tolerance due to either contiguous or non-contiguous loss of video frames.

Although such experiments were not our primary research concern, they were essential, along with the results of similar research, to give us an indication of human tolerance to the degradation of movie quality due to frame loss and to the loss of synchronization. Selected movies involved in this experiment contained clear conversations, distinct backgrounds, and clear transitions of scenes. The movies selected had their audio well synchronized with the existing video frames.

We used professional movie editing software to induce frame loss on 30 frame/second movies. Contiguous frame losses were applied, as well as non-contiguous frame losses. A contiguous frame loss in our context is a loss that has multiple consecutive video frames lost one after the other. The purpose behind this was to determine whether contiguity of frame losses contributed differently to the perception of synchronization loss.

For the non-contiguous frame loss experiments, twenty movie versions were created for each movie taking part in the experiments. Each created version had different frame losses. The first movie having one frame loss, the second having two frame losses, the third having three frame losses, and so on up to twenty frame losses. The induced losses were selected so that they were not concentrated at any one single portion. They were distributed across the original movie.

Randomly chosen human subjects were then asked to rate the given movies. The subjects were experimented upon in the same environment and under the same conditions. Each original movie was displayed to human subjects, then the newly created versions with induced loss were displayed one after the other, in increasing order of number of lost frames.

The subjects were then asked to indicate when they initially started noticing degradation in quality and a lack of synchronization between the audio and the video. They were also asked to rate the synchronization quality of each movie as either *highly acceptable*, *acceptable*, *fair*, *annoying*, or *completely unacceptable*. The results were then used to come up with a classification of human tolerance to non-synchronization.

2.1. The classification

There was unanimity among all the human subjects on the ranking of all the versions. We found that the contiguous loss has ultimately the same effect as the non-contiguous loss, for the same number of lost frames. As expected, contiguous loss caused the synchronization problem to appear earlier.

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 in the form of a slight skew between what is being said, and what the displayed video presents. Clearly, the more loss, the less the tolerance. The results agree with the findings of the IBM European Networking Center as relates to when desynchronization is observed, after approximately 160 ms of audio-video skew. We further extend that to create a classification of tolerance.

We thus classify the effect of synchronization loss on viewers as highly acceptable, acceptable, fair, annoying, and completely unacceptable. Based on this classification or scale, we measured the viewers' tolerance to synchronization as a function of the cumulative number of lost frames. Figure 2 shows our findings.

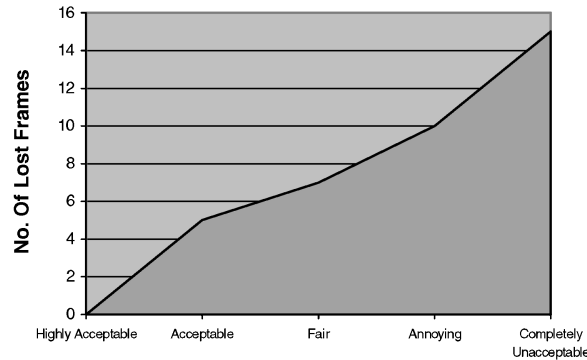


Figure 2. Tolerance of non-synchronization.

3. Proposed quality enhancement approaches

Several approaches can be adopted to enhance quality and contribute to the remedy of the synchronization problem. One approach is to compensate for all the lost frames at the receiver, by estimating them. Nevertheless, there arise 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.

Preventative approaches can be applied at the sender as opposed to the receiver. Applying such approaches at the sender provide the luxury of abundance of resources, and the operation on full data sets, as opposed to partial data sets caused by loss. Such approaches would add extra frames to the movie video stream before transmission, and within available bandwidth budgets.

Because it takes a certain amount of frame loss before a significant degradation in the category of video quality happens, adding redundant frames to the movie stream prior to transmission can still maintain a highly acceptable quality, except that the skew between audio and video is in the opposite direction. This way, at least double the amount of packets could be lost before the tolerance category drops. To expand more on the concept, as many video frames as required could be added to the stream, as long as bandwidth budget is taken into consideration. The elimination of unwanted frames at the client side would then be easier than the actual estimation of lost frames.

Nevertheless, several serious considerations have to be made when preventative approaches are deployed. Many problems include the appropriate location of frame insertion, the techniques used in the estimation of frames with different scene types, and the insertion of frames in a way such that synchronization could be preserved.

Another 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 enhancing quality, and contributing to bringing the synchronization level back to what it was before the transmission of the streaming movie.

4. Estimation of missing frames

We develop and evaluate the appropriateness of the application of five different frame estimation techniques to estimate lost or corrupted frames at the client side, taking into consideration resource constraints. *Motion tracking, quadratic interpolation, linear interpolation, two-way frame duplication, and one-way frame duplication* were developed and utilized. We investigate the suitability of each such developed technique for the estimation of missing frames under varying motion and loss conditions. Furthermore, hybrids of such techniques will be deployed either on the frame level or on the block level to best utilize each technique under varying loss patterns.

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 $x + 1, x + 2, \dots, x + k - 1$ have all been lost. Our objective is to attempt to quickly estimate all those lost frames as faithfully as feasible, and to add them in the appropriate locations without retransmission or any error correction information.

4.1. 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.

Such research has been vital, especially in image recognition and compression. Much research has been developed to estimate motion between frame sequences [1, 5, 7, 9, 11, 14, 16, 19, 20]. Nevertheless, when using motion estimation to reconstruct lost frames, and to restore synchronization between audio and video streams, one is encountered with many time and resource limitations.

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. Motion estimation in general has been used in standards like MPEG. Nevertheless, the assumptions and constraints by which we apply motion tracking in this context are fundamentally different.

When MPEG does its motion estimation, it does it offline with abundance of resources, but our purpose is to perform it online to estimate lost frames under limited resources. Furthermore, when MPEG applies motion estimation, it operates on a full data set, while in our purpose, we operate on a partial data set due to the presence of loss. When MPEG does motion estimation, it does it primarily for the purpose of compression, but we perform it for the purpose of loss estimation. Finally, although motion vectors transmitted with streams like MPEG could be utilized as estimates of motion between frames, the generality of our techniques allows us to track motion between frames on the fly regardless of the encoding scheme.

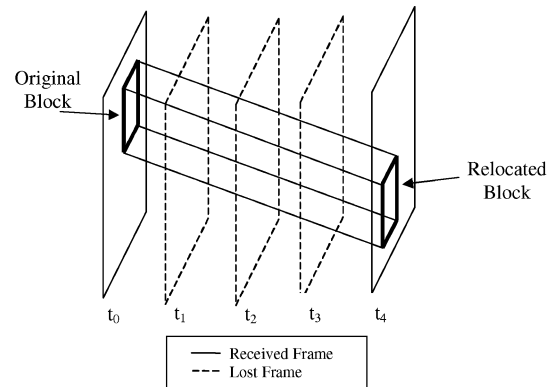


Figure 3. Motion tracking for frame estimation.

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 3.

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 16×16 blocks, and experimented on those. Furthermore, we generalized and experimented with other block sizes: 64×32 , 32×32 , 32×16 , 16×16 , 16×8 , and 8×8 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.

4.1.1. 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:

$$\begin{aligned}Y &= 0.299R + 0.587G + 0.114B \\C_b &= -0.1687R - 0.3313G + 0.5B + 128 \\C_r &= 0.5R - 0.4187G - 0.0813B + 128\end{aligned}$$

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 [6] 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 author in [6] present a technique by which such defect could be remedied.

4.1.2. 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 experimentation, we found that the normalization process helped to significantly enhance motion tracking.

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

- *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.
- *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.
- *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.

4.1.4. 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, and limited in a way to accommodate for 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 $x + k$. Furthermore, the values of the pixels within the blocks themselves could be linear interpolations.

4.1.5. "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.

Applying filters to the estimated frame is used 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.

4.2. 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 as shown in figure 4. Least-squares data fitting is used for such curve generation, using three or more

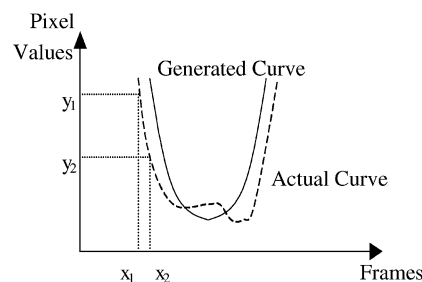


Figure 4. Quadratic interpolation using best fit curve generation.

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. Such process is naturally parallelizable as demonstrated in [4].

4.3. 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 $t = x + 1, x + 2, \dots, x + k - 1$ is:

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

4.4. One-way and two-way duplication

It is well known that in video streams, adjacent frames tend to have a high degree of similarity between them. Thus, if adjacent frames are lost, one way of estimating the lost frames is to simply duplicate adjacent frames.

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 $x, x + 1 \dots x + \lfloor k/2 \rfloor$ are made identical to frame x , while frames $x + \lfloor k/2 \rfloor + 1, x + \lfloor k/2 \rfloor + 2 \dots x + k - 1$ 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.

4.5. Hybrid frame estimation

It is always desirable to be able to utilize the best of what is available. Each frame estimation technique previously described has its own advantages and disadvantages. Some techniques behave better than others under certain circumstances, and behave worse than others otherwise. It is thus essential to develop an appropriate taxonomy of the different available techniques.

4.5.1. Requirements of a taxonomy. In order to deploy hybrid techniques to solve our problem, it is essential to develop an appropriate taxonomy that classifies the problem into one of several classes. Each class would then be paired with one of the existing techniques that best solves the corresponding class.

There are certain requirements for developing such a taxonomy. Such requirements are stated below:

1. *Perfectness*: For every class of problems, there has to be a technique that best solves the class. In other words, from amongst all the available techniques, there has to be a clear winner.

When perfectness is not satisfied, this is an indicator that either an alternative taxonomy has to be found that creates a clear technique winner for every available class, or that a refinement to the existing taxonomy has to take place. The perfectness property is testable.

2. *Low classification complexity*: After defining the taxonomy, it is essential that the actual classification process be of low complexity. In other words, the determination of where in the taxonomy a problem lies in should not incur much overhead. This property is measurable.
3. *Optimality*: This refers to the minimality of the classification time complexity.
4. *Scalability*: Scalability in this context means the ability to incorporate new frame estimation techniques into the hybrid. Scalability is a desirable property in many respects, but it is hard to test for priori. It is not achievable to test whether a system will be able to accommodate new techniques before they are actually created. In this case, scalability is non-testable and non-measurable.
5. *Minimality*: If two or more classes have the same winning technique, then it might be best to unite the classes into one class. Nevertheless, this should only happen if the classes are related and their unity reduces the classification complexity of the system. Such property is testable and measurable.
6. *Completeness*: It is very important that all problems could be classified, and could be solved using the existing techniques. It is highly undesirable that the taxonomy includes classes that are not handled by any of the existing techniques. This property is testable.
7. *Mutual exclusion*: No class has multiple winning techniques. This property is testable.

4.5.2. Technique analysis. When examining each technique individually, we observe the advantages and disadvantages of each. Both objective and subjective performance evaluation results found in [3] were used to analyze each technique individually. Frame sequences in [3] present SNR evaluations as well as visual results for the utilization of each of the techniques.

Duplication techniques in general are most suitable for small losses. When one or two frames are missing, simply substituting missing frames with surrounding frames is the most appealing alternative. For one thing, the loss is so small such that the jerkiness created by duplication would pass unnoticed by the vast majority of viewers.

Furthermore, within the span of one or two frames, i.e. the span of a tiny fraction of a second, not much change is taking place, and it would thus be most appealing to apply duplication. Two-way duplication was preferred over one-way duplication because it decreases users' perceived freezing time. Missing frames are duplicated from both ends of the loss as opposed to one end only.

In addition, one could argue that for very high frame losses, duplication might also be a desired technique. When large frame losses occur, it is more likely that much action took

place where the loss occurred, and thus attempting to estimate the loss might not be feasible. Simply duplicating the lost frames using surrounding frames would be a graceful solution to the loss as opposed to the application of any other technique that might or might not be effective in remedying the large amount of loss that occurred.

But again, simply having a large frame loss does not necessarily imply that estimation techniques would fail. In some situations, large frame losses occur in portions of the video that do not have much motion. Scenes could be slow in pace, and thus not much motion is taking place, which would imply the presence of high temporal redundancy that would promote the usage of other frame estimation techniques that estimate losses much better than duplication does.

Nevertheless, there are certain applications, such as video surveillance, where the application of estimation techniques in high loss situations, regardless of how much motion is taking place, is highly undesirable. When such high frame loss occurs, it would be best to clearly indicate that some loss occurred. One way of doing so is to simply freeze the video. It would be unethical to mislead viewers as to what happened exactly in place of the loss. Simply masking the large loss in this case by means of estimation would clearly violate the objective behind the application, which is the surveillance of every action taking place within a specific area.

Quadratic interpolation on the other hand is very suitable for low motion and with moderate to low frame loss. Lip motion is a clear example of low motion. It occurs in video conferencing and news broadcasts. Nevertheless, if the type of transmission is not known, there has to be a mechanism that would allow the decision engine to be able to decide that low motion could be taking place where the loss occurred, and thus quadratic duplication needs to be applied.

Although linear interpolation might also be suitable for estimating low motion, though not as well as quadratic interpolation, it is also very suitable for estimating moderate pace motion. Linear interpolation also has its disadvantages. It creates a ghost effect when much motion is involved.

Finally, when a lot of motion is involved between frames, motion tracking would be best suitable to track such motion and estimate the loss occurring between frames. The major disadvantage of motion tracking over the rest of the other techniques is that it requires more processing time.

Given the advantages and disadvantages of the estimation techniques, their suitability of application given the number of lost frames, and the type of motion taking place in scenes, there has to be a decision system that will decide upon which technique to apply given the circumstances surrounding the loss.

Such engine should take into consideration the amount of loss that happened; furthermore, it should be able to determine the kind of motion taking place so as to apply the appropriate technique. Low motion would require quadratic interpolation, moderate motion would require linear interpolation, and faster motion would require motion tracking. On the other hand, the decision engine should not consume much processing time, otherwise it would defeat its purpose of real time application.

Based on the foregoing considerations, we introduce a taxonomy based on the (1) degree of motion, and (2) extent of frame loss. Motion can be low, moderate, or high. Similarly,

4.5.3. The decision process. Given some of the criteria previously mentioned, deciding which technique to use requires a decision engine that is capable of performing the following:

1. Determining the amount of frame loss.
2. Recognizing the presence of low motion.
3. Recognizing the presence of moderate motion.
4. Recognizing the presence of fast motion.
5. Doing 1–4 in a short time.

There are several sophisticated techniques that would be able to determine with high accuracy the kind of motion taking place within a specific segment of a movie. Some techniques would perform complicated motion analysis to determine to a high degree of accuracy that low motion, moderate motion, or even fast motion is taking place. Nevertheless, using such sophisticated techniques would defeat our purpose.

Deciding which technique to use to estimate lost frames is already an overhead imposed on our system. We do not want to utilize sophisticated classification techniques that would further increase the overhead. Instead, we would like to utilize some classification techniques that would allow us to better determine which estimation technique is more suitable for deployment without much processing overhead.

One very appealing way of deciding on the amount of motion between frames surrounding the loss is to use SNR computations. Much as SNR is specifically effective for the comparison of compressed images and originals, SNR calculations could also be utilized to determine the degree of change happening between two images as shown in figure 7.

Frames surrounding the loss could then be utilized in the SNR computations. Since the surrounding frames could be substantially different in content depending on where the loss

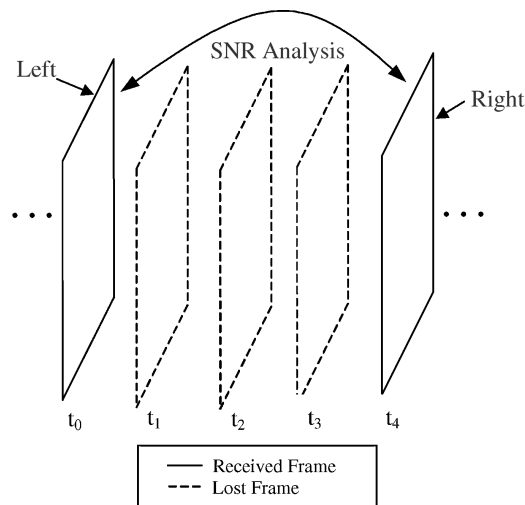


Figure 7. Determining the degree of motion.

occurred in the movie stream, it is not an issue then to determine how good the SNR is, but rather how bad it is.

High motion between frames would then imply lower SNR values compared to low motion that would give higher SNR values. SNR computations would then be a quick means of determining the degree of change between surrounding frames, and hence the rate by which change happened over time amongst such frames.

Motion vector analysis could further be used to give a more accurate estimate of the amount of motion taking place surrounding the loss. Nevertheless, motion vector analysis is more time consuming, and should then be used as a last resort in the decision making process.

The decision engine as shown in the flowchart of figure 8 would start by determining if any frame loss happened. If no loss occurred, then the engine does not go through any motion analysis. The decision engine would then traverse the different criteria in order to select the most appropriate technique for application.

We would start off first by determining the number of lost frames. If low frame loss occurred, then two-way duplication is the preferred technique because simply duplicating frames in such low loss situations will usually pass unnoticed by viewers. Furthermore, duplication in general is the fastest technique to apply.

After checking for low frame loss, we would then check for the existence of low motion, in which case quadratic interpolation would be applied if there is moderate frame loss, and then for moderate motion, in which case linear interpolation would be applied. If both those criteria fail, then before we start using motion tracking, which consumes more processing time than the other estimation techniques, we would check for the existence of high frame loss.

If high frame loss occurs, and no low or moderate motion exists, this would mean that the surrounding frames are substantially different. When the surrounding frames are substantially different, and the number of lost frames is large, then it would be best to simply duplicate the lost frames. This is a means of both saving time, and performing a graceful estimate of the lost frames in a way the other techniques would not be able to handle under such high number of lost frames. If the high frame loss criterion fails, this would then mean that the surrounding frames are substantially different, but the number of lost frames is not large. In this case, motion vector analysis, and hence the more time consuming motion tracking would take place.

4.5.4. The decision system. The decision system will be responsible for quickly deciding what type of motion is taking place amongst frames surrounding the loss, and hence applying the appropriate frame estimation technique. In order to be able to perform such decision, the decision system must be able to identify the amount of loss that happened, and furthermore, to identify the type of motion taking place surrounding the frame loss. It needs to identify whether low, moderate, or high motion is taking place.

The amount of motion taking place surrounding frame loss will be determined by signal to noise ratio analysis. Computed signal to noise ratio values for frames surrounding the loss will determine whether low, moderate, or high motion is taking place surrounding the lost frames. Based on such signal to noise ratio values, and based on the amount of existing frame loss, the appropriate technique will be applied to remedy the error.

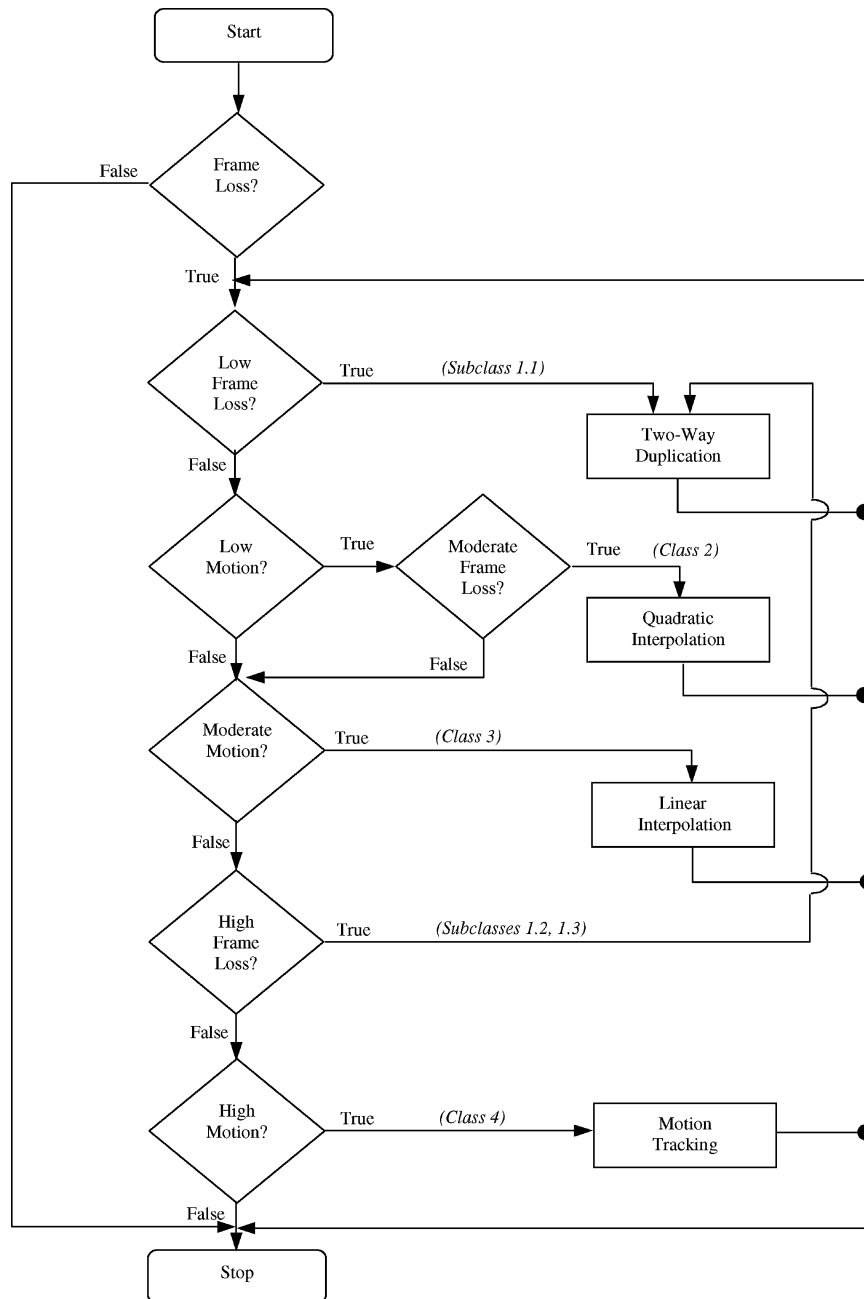


Figure 8. The decision process.

The issue is then to classify SNR ranges that correspond to low, medium, and high motion. The advantage of having such SNR ranges available before hand is that it reduces the majority of the complexity of the decision problem to simply computing SNR values surrounding frame losses, and then using the predetermined classification of ranges to determine the type of motion taking place. The amount of real time processing is thus significantly reduced.

4.5.4.1. Defining the degree of motion. Previously, we had mentioned that our decision system would primarily be based upon the degree of motion, and the amount of frame loss. In other words, we base our decision on the existence of low, moderate, or high motion and frame loss.

As it relates to motion, it is therefore necessary to define what is meant by low, moderate, and high motion. Based on such definition, both subjective and objective experiments would be performed to determine before hand the SNR thresholds between frames that correspond to each one of the previously defined types of motion.

We subjectively define the degree of motion as follows:

- *Low motion:* Intra object motion. Such type of motion includes facial expressions, lip motion, blinking, finger motion, but does not include terminal motion such as the motion of arms or legs.
- *Moderate motion:* Rotational and terminal motion. Such type of motion includes the movement of arms and legs, neck motion, and the rotation of bodies.
- *High motion:* Inter object motion. Such type of motion includes the spatial movement of objects relative to one another. Whether it is an object moving in front of a stationary background, or a background moving behind a stationary object, or objects moving relative to one another.

4.5.4.2. Defining the degree of frame loss. We define the degree of frame loss as being low, moderate, and high. Such degree along with the degree of motion primarily determines the type of technique to be applied in frame estimation. Table 1 shows the classification of the extent of frame loss. This classification is based on our experimentation.

4.5.5. Suitability of the decision system. One could come up with a different process that applies the different techniques based on different criteria; nevertheless, our decision process proves to be adequate for the following reasons:

Table 1. Degree of frame loss.

No. of lost frames	Degree of frame loss
[0, 2]	Low
[3, 4]	Moderate
[5, ∞]	High

1. The ability to exploit the strength and avoid the weakness of each technique.
2. Operation using simple inputs, namely SNR values and the amount of loss.
3. Minimum classification overhead, based on SNR and frame loss classification. SNR cutoff values for determining low, moderate, and high motion are predetermined offline. There is no need to perform online analysis.
4. Insensitivity to scene types. The decision process does not care about the type of scenes in the movie, whether it is a person talking, or a plane flying, the process is insensitive to such issues.

4.5.6. Conformance to taxonomy requirements. We need to verify that the proposed hybrid decision system conforms to the taxonomy requirements stated earlier.

4.5.6.1. Perfectness. The proposed system summarized in figures 6 and 8 demonstrate perfectness by having a clear winner for every established class of problems. The superiority of such techniques in the corresponding classes was previously justified in the paper.

4.5.6.2. Low classification complexity (optimality). The determination of which class the loss belongs to is linear with the frame size. It takes $O(N \times M)$ complexity to determine the class the problem belongs to. N and M are the frame height and width consecutively. Furthermore, the classification is independent of the amount of frame loss.

Since classification of any sort can not be possibly be done without inspecting at least one frame fully, $\Omega(N \times M)$ is clearly a lower bound on classification time. Since we achieved it, our classification is time-optimal.

4.5.6.3. Scalability. As relates to scalability of the hybrid decision system with the video length, the decision system is independent of video size. On the other hand, the hybrid decision system is also independent of the amount of loss a video can encounter.

What remains is the scalability of the hybrid decision system with respect to the incorporation of new loss estimation techniques as they are developed. Nevertheless, determination of how scalable a system is to the addition of techniques that have not yet been developed to the existing taxonomy is difficult to determine a priori.

4.5.6.4. Minimality. The problem classes involved in the hybrid decision system are unique. Each class of problems is best solvable using one of the given techniques. Thus, there is a one to one mapping between the techniques and the problem classes. Each one of the techniques applied to the problem classes is a clear winner, and if any of the classes were to be further united, we would lose such property.

4.5.6.5. Completeness. The SNR thresholds determining the degree of motion, as stated in the previous section, are comprehensive. They cover all possible SNR ranges, and thus the determination of the degree of motion is complete. Furthermore, the determination of the degree of frame loss is also complete. All frame loss possibilities are accounted for in those thresholds. The SNR and the amount of frame loss are the only two input variables necessary for the system to classify the problem and match them with the corresponding

appropriate technique. Therefore, there is no situation where the classification system is unable to account for. The hybrid decision system is complete.

4.5.6.6. Mutual exclusion. No two problems are classified to more than one class. The classification is thus mutually exclusive.

4.5.7. Determining SNR thresholds. In order to determine the SNR ranges corresponding to the different types of motion, we conducted experiments on different videos with different types of motion. The objective was to determine SNR thresholds that distinguish low, moderate, and high motion between frames.

We conducted both objective and subjective experiments on the frames of the movies involved in the experiment in order to determine the SNR thresholds associated with low, moderate, and high motion. Both techniques will be described below.

4.5.7.1. Subjective threshold determination. For each video involved in the experiment, frames were extracted from the video stream. In order to analyze SNR values associated with different types of motion previously mentioned, SNR values were then computed for every existing frame in the movie, and the sixth consecutive frame corresponding to each such frame as shown in figure 9. In other words, frame zero was compared to frame six, frame one with frame seven, frame two with frame eight, and so on until the end of the video. The reason why the sixth consecutive frame was an issue of comparison was because of its relation to the number of frames that have to be lost before users start noticing synchronization anomalies.

Based on the observed type of motion between every five consecutive frames, calculated SNR values were associated with every pair of frames taking part in the computation. Subjective evaluations were performed on the frames and the corresponding computed

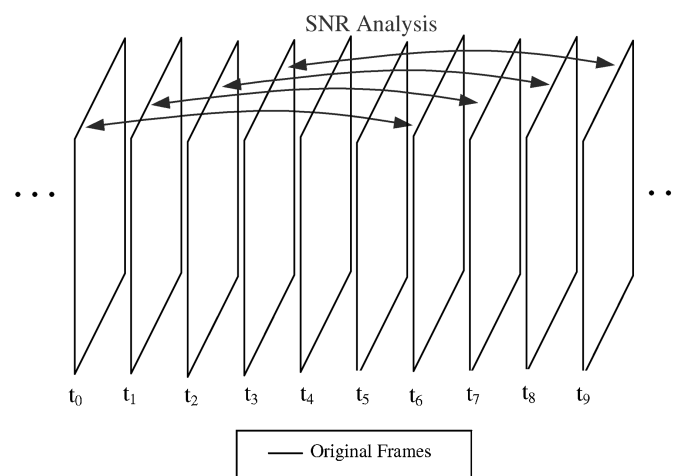


Figure 9. Subjective SNR threshold determination.

Table 2. Motion SNR thresholds.

SNR (dB)	Degree of motion
>15	Low
[10, 15]	Moderate
<10	High

SNR values to come up with the classification of the SNR threshold values corresponding to different types of motion.

Using the notion of low, moderate, and high motion, the following SNR thresholds were found as shown in Table 2.

4.5.7.2. Objective threshold determination. A video segmentation system developed in [2] was used to objectively determine the presence of segments in our experimented upon videos, and to be able to determine SNR relations to types of motion. This system was initially developed to segment videos given certain sensitivity parameters that determine how fine-grained the segmentation should be.

The automated segmentation of the system in [2] is strongly coupled with human perception of video segmentation, and was thus the system of choice for our experiments. It is calibrated and sensitive to both quantitative and qualitative aspects of video segmentation including both human notions of segments in videos and mathematical models for video segmentation. Users of such system can control the granularity of segmentation depending upon the desired application.

Different segmentation runs, five to be specific, were applied to each of the videos. Each run had different segmentation granularity ranging from coarse to fine grain. The finer the segmentation granularity, the more the video is segmented, thus catching minor motion.

In order to catch the three types of motion previously defined, namely low, medium, and high motion, fine-grain segmentation was applied to the videos. Pre-computed SNR values were then compared to the video segmentation results. SNR thresholds were established for the different types of motion. The following segment boundaries shown in figure 10 demonstrate low, medium, and high motion.

The boundaries of the segments were examined along with their corresponding predetermined SNR values to conclude the SNR values associated with such segments. Furthermore SNR ranges corresponding to low, moderate, and high motion were found, and were similar to SNR threshold values found in the subjective evaluation section.

For low motion, there are more frames per segment than when there is higher motion between frames. What the figure demonstrates to the casual observer is the fact that although fine-grain segmentation is applied, there are more frames in a segment when there is lip motion than when there are higher degrees of motion.

The most extreme situation, where there is a high degree of motion, each two consecutive frames, and sometimes individual frames form a segment on their own. Such segmentation allowed us not only to determine the SNR thresholds associated with the types of motion, but also the number of frames per segment allowed us to determine the degree of motion as perceived by the segmentation system.

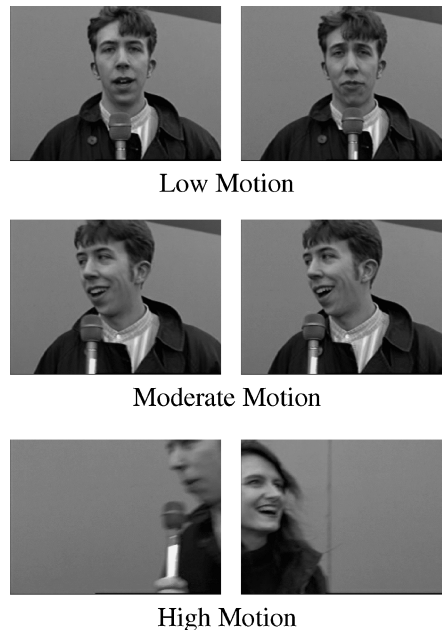


Figure 10. The degrees of motion.

4.5.8. The affordability of the application of hybrid techniques. A remaining issue now is how affordable the application of hybrid techniques is. Although the application of hybrid techniques produces better results, the classification of the problem itself is an overhead, regardless of how minimal it might be. There arise situations though where it might not be affordable to incur the extra overhead for classifying a problem and applying hybrid techniques on it.

To determine how affordable such application is, we can apply hybrid techniques only if the overhead of the classification in addition to the maximum application time of any existing error correction technique is within budget. If that is not the case, then the application of any default technique could be preferred.

4.5.9. Block-based hybrid estimation. In order to achieve better precision in the estimation of motion in frames, one has to consider the fact that different portions of frames could be experiencing different degrees of motion. For example, a portion of a frame could have low motion represented by a person simply talking, but another portion could have high motion represented by birds flying over the person's head.

A variation of hybrid frame estimation would then operate on individual blocks within a frame as opposed to the entire frame. Operating on individual blocks would then attempt to capture different degrees of motion in different portions of the frame. The choice of block sizes would thus be crucial. Smaller blocks do not necessarily imply better performance.

Each block would thus be treated like a frame would be treated. Nevertheless, the SNR thresholds previously computed for determining low, moderate, and high motion in frames would have to be calibrated for the operation on blocks as opposed to frames.

5. Conclusions and future work

It is possible to remedy video frame 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 in the video stream.

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. Five estimation techniques were developed and applied to solving the problem namely motion tracking, quadratic interpolation, linear interpolation, two-way duplication, and one-way duplication. Furthermore, a classification of video loss was developed and hybrids of estimation techniques were built in order to best utilize the techniques based on the given loss class identified by the amount of loss and the degree of motion in the place of loss. The studies done within the scope of this research focused on enhancing video quality, and 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 one-way and two-way duplication created fast and good estimates of the lost frames, but also created a freezing effect in the video stream. Linear interpolation and quadratic interpolation eliminated this freezing effect at the cost of some time overhead. Linear interpolation gave the best results overall. However, subjective evaluations of quadratic interpolation and motion tracking showed that those techniques produce very good estimates of lip motion and fast motion consecutively, which is important in news broadcasts and teleconferencing, among other applications.

We are currently investigating additional, more sophisticated techniques for frame estimation and synchronization restoration. They include motion tracking enhancement, processing of the audio stream, differentiated error protection, and 3D transforms. 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.

References

1. D. Aaron and S. Hemami, "Dense motion field reduction for motion estimation," in Proc. Asilomar Conference on Signals, Systems, and Computers, Nov. 1998.
2. S. Almogbel and A. Youssef, "Flat and hierarchical segmentation," Dsc. Dissertation, the George Washington University, 2000.
3. S. Aly and A. Youssef, "Synchronization-sensitive frame estimation techniques for audio-video synchronization restoration," in Proc. Internet Computing 2000 Conference, Las Vegas, Nevada, 2000.
4. S. Aly and A. Youssef, "Frame estimation for restoring audio-video synchronization using parallelized quadratic frame interpolation," in Proc. PDPTA'2000 Conference, Las Vegas, Nevada, 2000.

5. M. Bennamoun, "Application of time-frequency signal analysis to motion estimation," in Proc. ICIP97, 1997.
6. G. Bruck, "A comparison between the luminance compensation method and other color picture transmission systems," IEEE Transactions on Consumer Electronics Vol. 36, No. 4, pp. 922-932, 1990.
7. G. Conklin and S. Hemami, "Multi-resolution motion estimation," in Proc. ICASSP '97, April 1997.
8. L. Ehley, B. Furht, and M. Ilyas, "Evaluation of multimedia synchronization techniques," in Proc. International Conference on Multimedia Computing and Systems, 1994.
9. C. Fan and N. Namazi, "Simultaneous motion estimation and filtering of image sequences," in Proc. ICIP97, 1997.
10. G. Haskell, A. Puri, and A. Netravali, Digital Video: An Introduction to MPEG-2, Chapman & Hall: New York, 1997.
11. J. Magarey et al., "Optimal schemes for motion estimation using color image sequences," in Proc. ICIP97, 1997.
12. K. Naik, "Specification and synthesis of a multimedia synchronizer," in Proc. International Conference on Multimedia Computing and Systems, 1994.
13. I. Rhee, "Retransmission-based error control for interactive video applications over the internet," in Proc. IEEE Conference on Multimedia Computing and Systems, 1999.
14. V. Ruiz et al., "An 8×8 -block based motion estimation using Kalman filter," in Proc. ICIP97, 1997.
15. S. Son and Nipun Agarwal, "Synchronization of temporal constructs in distributed multimedia systems with controlled accuracy," in Proc. International Conference on Multimedia Computing and Systems, 1994.
16. C. Tomasi, "Pictures and trails: A new framework for the computation of shape and motion from perspective image sequences," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, June 1994.
17. B. Wah and X. Su, "Streaming video with transformation-based error concealment and reconstruction," in Proc. IEEE Conference on Multimedia Computing and Systems, 1999.
18. T. Wahl and K. Rothermel, "Representing time in multimedia systems," in Proc. International Conference on Multimedia Computing and Systems, 1994.
19. Y. Yang and S. Hemami, "Rate-constrained motion estimation and perceptual coding," in Proc. IEEE Conference on Image Processing, Oct. 1997.
20. T. Yoshida et al., "Block matching motion estimation using block integration based on reliability metric," in Proc. ICIP97, 1997.



Disk
followed

Sherif G. Aly received his B.S. degree in Computer Science from the American University in Cairo, Egypt, in 1996. He then received his M.S. and Doctor of Science degrees in Computer Science from the George Washington University in 1998 and 2000, respectively. He worked for IBM during 1996, and later taught at the George Washington University from 1997 to 2000 where he was nominated for the Trachtenberg prize-teaching award for his current scholarship and scholarly debate. He then spent two years as a guest researcher for the National Institute of Standards and Technology from 1998 to 2000. Dr. Aly published numerous papers in the area of distributed systems and multimedia. He is currently working as a Research Scientist at Telcordia Technologies in the field of Internet Service Management Research. His current research interests include multimedia, directory enabled networks, and image processing. Dr. Aly is a member of IEEE.



Abdou Youssef received the B.S. degree in Mathematics from The Lebanese University, Lebanon, in 1981, the M.A. and Ph.D degrees in Computer Science from Princeton University, Princeton, NJ, in 1985 and 1988, respectively. He taught for a year in 1982 at the Institute of Applied Sciences, The Lebanese University. Dr. Youssef joined the Department of Electrical Engineering and Computer Science at The George Washington University, Washington DC, in Fall of 1987, serving as Assistant Professor from 1987 to 1993, then as Associate Professor from 1993 to Spring 1999, and as Full Professor since September 1999, now in the newly formed Department of Computer Science. He has spent his sabbaticals at the National Institute of Science and Technology and, partly, at the Johns Hopkins University. He has published numerous papers in the areas of parallel processing and computer architecture, interconnection networks, data compression, image & video processing, information retrieval, and multimedia. He co-edited a book titled "Interconnection Networks for High-Performance Parallel Computers," published by the IEEE Computer Society Press. He is a three-time recipient of the Teacher of the Year Award from his Department and School, in 1995, 1997, and 1998. He is also listed in the Who is Who Among America's Teachers. Dr. Youssef has lectured throughout the world, including Germany, China, Brazil, and Canada. His current research interests include image & video processing, data compression, distributed multimedia, advanced information retrieval, parallel processing and algorithms. Professor Youssef is a senior member of IEEE.