

Methods for Matching Compressed Video to ATM Networks

Rahul Garg¹

rahul@niyati.iitd.ernet.in

Tel: +91-11-686-7431, 685-7649

Fax: +91-11-686-8765

Department of Computer Science and Engineering

Indian Institute of Technology

Haus Khas, New Delhi

India

R. J. Safranek

C. R. Kalmanek Jr.

\$Date: 1998/03/07 12:14:38 \$

¹Corresponding Author

Abstract

Over the last few years three technologies have reached the stage of maturation where they can be synergistic. These are wideband, high speed networking, high quality video compression (MPEG-I and II) and high capacity affordable digital storage media.

This paper addresses the interaction of these three technologies. In particular, it examines the problem of taking a compressed video data stream that is stored on a server, and transmitting it over an ATM channel which has a capacity smaller than that required by the data stream. The conventional approach to this problem would be to transcode by decoding the video data, and then to re-encoding so as to meet the channel constraints. Currently this is not a cost effective solution since, while MPEG decoders are relatively inexpensive, encoders are not.

Our approach to this problem is to partially decompress the video bitstream. Then, perform the transcoding in the quantized data domain. Finally, a valid bitstream is reassembled and transmitted. This approach has the advantage of providing nearly identical quality as the traditional transcoding approach, at a fraction of hardware cost.

Keywords: MPEG, ATM, Transcoding, Video on Demand, Video, Rate Control.

1 Matching Previously Compressed Video to a Transmission Channel

An interesting application that is starting to be examined is the delivery of stored compressed video over varying transmission media. In examining this application, a potential problem becomes apparent quite quickly; namely: what if the transmission media has a lower rate capacity than the compressed video requires? The traditional approach to this problem would be to decode the video, and then recompress to the lower bit-rate as shown in Figure 1. Unfortunately, MPEG is a asymmetric codec. The encoder requires dramatically more processing power than the decoder. So much so, that this approach is not an economically viable solution. Another approach would be to use a scalable encoder where the bitstream consists of multiple layers. The base layer will provide the minimum quality version of the program, while the enhancement layer(s) provide additional quality. This approach has the disadvantage of providing only a small number of possible transmission rates.

An application where it is desirable to have finer control over transmission rate than is achievable by layered codec is variable bit-rate video transport over ATM (Asynchronous Transfer Mode) networks. Since these type of networks do not have infinite capacity, a congestion control policy is agreed to when a connection is setup. An example of one such policy is leaky bucket control. In this control policy, the transmitter agrees to a transmission rate and a burst size. The burst size specifies the maximum data that the transmitter can send at rate higher than transmission rate.

Reibman and Haskell [6] have shown the constraints an encoder must meet to satisfy such a control policy and have given examples of suboptimal rate control algorithms that satisfy those constraints. This requires the encoder to agree to meet conditions on the compressed bitstream that would not be known when the video was compressed for storage on the server. Therefore, the stored video may not meet them.

2 Overview of Approach

Our approach to this problem is to perform the rate change process in the *compressed domain*. This requires only partial decompression of the bitstream and avoids the need to perform the most computationally expensive operations of the encoder. Simultaneous with the development of this approach Morrison et. al. [4], and Yong et. al. [8] have proposed similar techniques for bit rate adjustment. This work differs from their proposals by eliminating the need to compute any DCT's in the transcoder, it presents an analysis of requantization error, and an approach to optimal rate control.

All of the current image and video compression standards use bitstream that consists of four types of data, a) bitstream synchronization information; such as

start and marker codes, b) control information; such as image size, frame rate, coding modes and quantizer scales, c) entropy coded unquantized data; such as motion vectors and d) entropy coded quantized data; such as DCT coefficients.

Our approach is to partially decompress the bitstream, including the entropy coding stage, and perform the rate conversion on this representation. The technologies required to perform this modification process have recently been discovered. Smith and Rowe [7] have shown how to perform arithmetic operations in the compressed transform domain. Their focus was on performing special effects, i.e. graphic overlays, inserts, dissolves, etc., on intraframe encoded images.

If the compressed video stream consists only of intracoded data, i.e. there is no temporal prediction, the rate change process is quite simple. The standard video codecs (H.261, MPEG-I,II, H.263) all use the same approach to quantization. Each DCT coefficient is quantized using a uniform quantizer (possibly with a dead zone). The quantizer step size for each coefficient is comprised of two components, a quantization matrix, $QM(i, j)$, where (i, j) are used to specify the particular coefficient, which give a minimum step size for each coefficient, and a scaling parameter called $Mquant$. $Mquant$ is used to adjust output rate of the bitstream. As $Mquant$ increases, the quantizer step-size increases and the output rate decreases. Conversely, as $Mquant$ decreases, the output bit-rate increases. In particular

$$S(i, j) = Int[\frac{X(i, j)}{Mquant * QM(i, j)}] \quad (1)$$

and

$$\overline{X}(i, j) = S(i, j) * Mquant * QM(i, j) \quad (2)$$

where $X(i, j)$ is the input DCT coefficient and $\overline{X}(i, j)$ is its reconstructed value, $S(i, j)$ is its quantized value and $Int[]$ denotes integer division with rounding.

The requantization process can be carried out on the compressed domain representation by means of the following equation:

$$S_{new}(i, j) = Int[S_{old}(i, j) * \frac{Mquant_{old}}{Mquant_{new}}] \quad (3)$$

where S_{old} is the representation value of input DCT coefficient, $S_{new}(i, j)$ is the representation value of the requantized version, $Mquant_{old}$ is the input value of $Mquant$ and $Mquant_{new}$ is the value required to meet the output rate constraints.

Straight forward application of these methods will not work for motion compensated DCT based coders since they use the previously encoded frame as a prediction for the next frame. If the quantization of the reference frame is changed, the resulting quantization error must be propagated to the succeeding

frames. It turns out that this can be performed in the compressed domain as well. Chang and Messerschmidt [1] have developed a system for compositing multiple video streams into a single video stream. As a part of this work, they have developed a technique that can be applied to error propagation problem.

In motion compensation, a spatial domain block from a previous frame is selected to be used as the prediction for a block in the current frame. It should be noted that this block can come from anywhere in the previous frame and does not need to be aligned to the DCT grid used for encoding the current frame. Chang and Messerschmidt noted that the prediction information for a DCT block in current frame can have contributions from at most four blocks in the previous frame. Using this observation, they proposed decomposing the motion compensation problem into upto four identical subproblems. Each DCT block in the previous frame, that contributed to the prediction is multiplied in the spatial domain by a binary mask that selects the subset of pixels which contribute to the prediction. However, these pixels may not be located in the correct portion of the DCT block. To remedy this, the selected pixels are shifted to the appropriate location. Lastly, the contributions from each of the four contributing blocks are summed up to form the prediction.

Their main contribution was to describe a method of efficiently implementing the process directly on the transformed domain data. In the transformed domain, the extraction and shifting process can be performed using matrix multiplications.

$$DCT(B') = \sum_{i=1}^4 DCT(H_{i1}) * DCT(B_i) * DCT(H_{i2}) \quad (4)$$

where B' is the predicted block in the current frame, the B_i , are blocks in the previous frame which contribute to the prediction, H_{i1} is a constant matrix which depends on the number of columns that need need to be extracted from block B_i , and H_{i2} is constant matrix which depends on the number of rows that need to be extracted from block B_i . Since the H matrices are constant, they can be precomputed. This reduces the worst case complexity of transform domain motion compensation operation to two matrix multiplication.

In Figure 2 an architecture that can perform the rate conversion process and take into account requantization error is presented. The input bitstream is first converted into the compressed domain representation. Then the quantized DCT coefficients are multiplied by the input $Mquant$ value to generate their reconstructed values. Since the same quantization matrix is used throughout this process, it can be ignored during the requantization process. The requantization error from previous frame is now added to this value. This quantity is now requantized using value of $Mquant$ that is based on the output rate constraints. The requantized DCT coefficients and the new value of $Mquant$ are transmitted. Next, the requantization error is computed by differencing the error corrected quantized DCT coefficients and its requantized value. This error

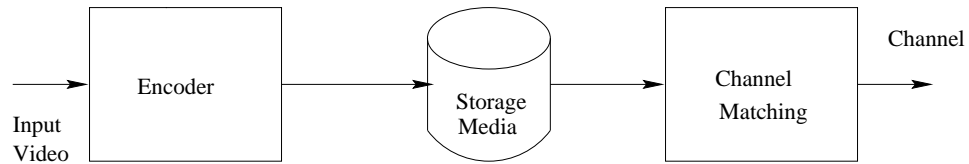


Figure 1: Overview of Channel Matching Process

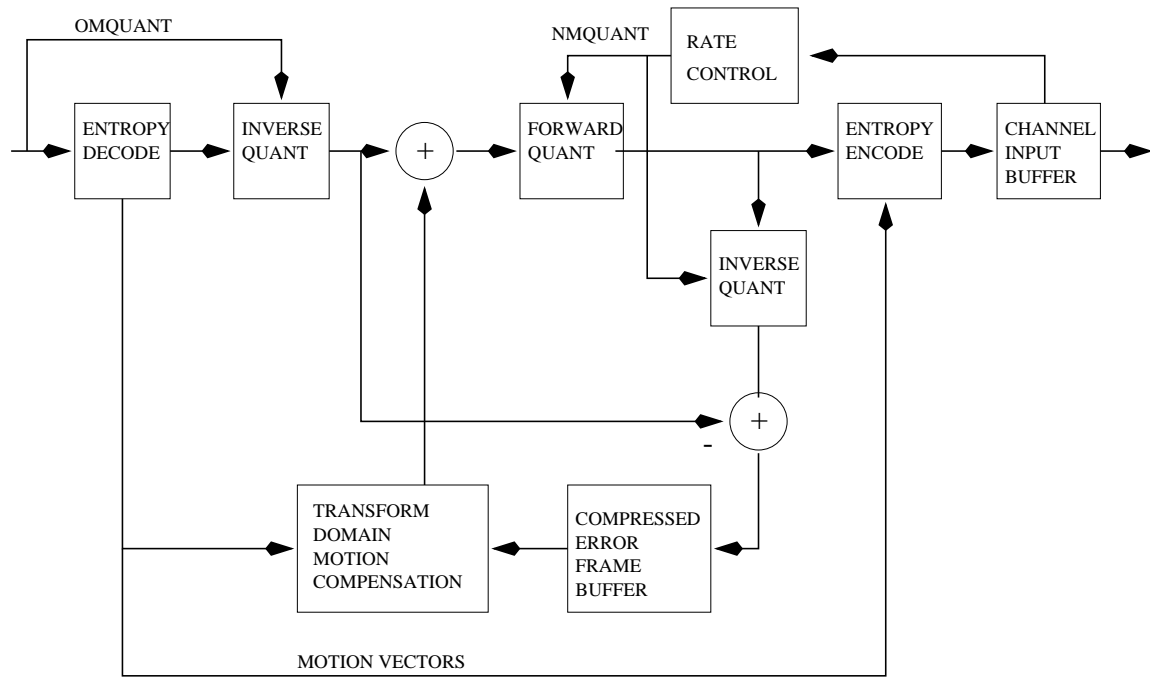


Figure 2: Rate Conversion Architecture for Motion Compensated Coders

is now stored in a compressed domain buffer for future use. When it is required, the method of Chen and Messerschmidt is used to motion compensate it in the transform domain. It should be noted that no DCTs are required to perform the rate conversion operation.

Once the data has been requantized and error propagation accounted for, the bitstream must be reassembled. As a part of this operation, some of the control parameters may need to be recomputed. For example, MPEG sends a buffer fullness value for each frame.

The computational complexity of this approach is quite moderate. An implementation of this approach for intra coded data can run in real-time on a 50 MIPS class processor for data rates of upto 3 megabits per second. The full implementation, which includes the transform domain motion compensation runs at about 20 frames per second for normal Group of Picture configurations for 1 megabit per second MPEG-I video streams on the same processor.

3 Optimality of Requantization Process

In the previous section it was shown that the requantization process can be performed efficiently in the transform domain. It does not address the problem of optimality of this process. The optimal requantization of a DCT coefficient would result in the requantization coefficients having the same value as if it had been quantized directly from the original input data. This is shown in Figure 3. The upper path illustrates the direct encoding of video material to desired channel constraints. The lower path shows the type of system being considered here. The input video is initially quantized with finer step sizes than those used for the final output. This version is stored and at some later time, the quality is reduced to that represented by the coarse quantization. This process can be viewed as a successive coarsening, since information can only be lost via the requantization process.

In order to establish the conditions under which optimal requantization can be performed, examine Figure 4. Optimality occurs whenever the unquantized data and its representation level when quantized with the fine quantizer, both get mapped to the same representation level in the coarse quantizer. From this figure, it is clear that no requantization error occurs whenever the entire quantization cell of the fine quantizer is completely contained in the quantization cell of the coarse quantizer. For this situation to occur in general, all of the decision boundaries of the coarse quantizer must match up with decision boundaries from the fine quantizer. Note that some decision boundaries from the fine quantizer will not be matched.

If these conditions are not satisfied, the requantization process is not optimal and additional error is introduced. This is illustrated in Figure 5. In this case, the unquantized data and its representation level using the fine quantizer fall on different sides of decision boundary of the coarse quantizer. This will occur

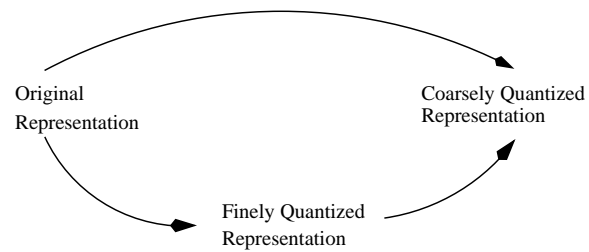


Figure 3: Possible Approaches to Quantizing a Video Signal

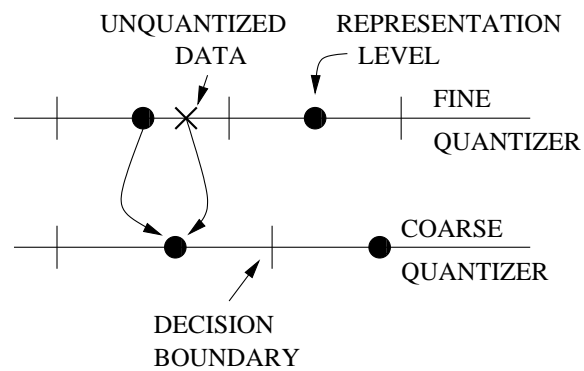


Figure 4: Situation Where Optimal Re-Quantization Can Occur

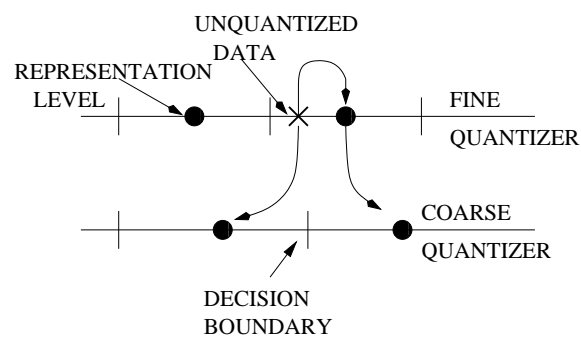


Figure 5: Situation Where Optimal Re-Quantization Can Not Occur

whenever a quantization cell from the fine quantizer straddles a decision boundary of the coarse quantizer. This leads the requantization process to generate a different representation level for the coarse quantizer than would be generated by direct quantization of the original data. Since in the fine quantization process the exact value of the original data is lost, it is only possible to estimate the magnitude of the error that is introduced. From Figure 5, it can easily be seen that the minimum error is equal to $\delta_{coarse}/2$, where δ_{coarse} is the step size of the coarse quantizer, and the maximum depends on the difference between the decision boundaries of the coarse and fine quantizers.

Now that the conditions for optimal requantization and bounds on the requantization errors are known, the last question that remains is how to control the requantization process so as to minimize requantization error while minimizing perceived quality variation in the output video stream. Ramchandran, Ortega, and Vetterli [5] have recently introduced an optimization algorithm that can be applied to this problem. They propose an operational rate-distortion approach to formulating the rate control problem. This allows the use of discrete optimization under constraints to be used to solve it.

A straight forward extension of this framework can also be used to simultaneously meet the output rate constraint, and minimize both requantization error and perceived quality variations. The constraint on minimizing quantization error in their formulation is replaced by a constraint on minimizing requantization error. Using only this constraint has the potential to introduce drastic quality variations though. Since conditions exist under which the requantization error is zero, the optimization could produce no requantization error by switching back and forth between several quantizers with vastly different step sizes. This can lead to perceptible quality variation within a video sequence that would not occur if direct quantization were used. This can be remedied by introducing another constraint which encourages quantizer step size to vary smoothly across an image. For conditions where small rate adjustments are required, this constraint will allow for smooth changes in output image quality at the cost of introducing some requantization error.

4 Conclusions

This paper has presented an approach to adjusting the output bit-rate of a previously compressed video bitstream in an efficient manner. In addition the expressions for requantization error were developed and an algorithm for performing optimal assignment of *Mquant* has been presented.

These techniques are not limited to just the ATM leaky bucket channel, but can be applied to any type of rate conversion situation. Other applications include variable to constant bit-rate conversion and rate reduction of constant bit-rate streams. In addition, since the rate conversion can be done inexpensively, it opens up the possibility of using feedback information of channel to

control the transmission bit-rate. Currently if congestion occurs in packet networks, bits are dropped in transmission. This can result in suboptimal video quality at the receiver. Recently several approaches on how such networks can generate a feedback signal to the transmitter have been proposed [3] [2]. Such feedback in conjunction with the rate conversion technology can help reduce network congestion and minimize the probability of packet loss.

Finally, the same techniques can be used to solve the commercial insertion problem. This is a typical problem in broadcast television applications where a local broadcaster wants to add a locally generated commercial into a national program feed. In this problem, a second video stream must replace a portion of current one. For this to occur seamlessly, the buffer occupancy constraint of the current stream must be met. This can be done by switching between the national feed and the local commercial at the appropriate time (at a Group Of Pictures boundary), and passing the resulting stream through the rate conversion apparatus. The rate converter will ensure the output bitstream is valid and of maximal quality. When the commercial has completed, another switch back to the national feed is executed.

5 References

References

- [1] S. F. Chang and D. E. Messerschmidt. Manipulation and compositing of MC-DCT compressed video. *IEEE Journal on Selected Areas of Communications*, 13(1):1–11, January 1995.
- [2] H. Kanakia, P. P. Mishra, and A. R. Reibman. An adaptive congestion control scheme for real-time packet video transport. *Computer Communication Review*, 23:20–31, October 1993.
- [3] S. Keshav. A control-theoretic approach to flow control. In *Proceedings of SIGCOMM*, pages 3–15, Zurich Switzerland, September 1991. ACM.
- [4] D. G. Morrison, M. E. Nilsson, and Ghanbari M. Reduction of the bit-rate of compressed video while in its coded form. In *Proceedings of the Sixth International Workshop on Packet Video*, September 1994.
- [5] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG coders. *IEEE Transactions on Image Processing*, 3(5):533–545, September 1994.
- [6] A. R. Reibman and B. G. Haskell. Constraints on variable bit-rate video for ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2(4):361–372, December 1992.

- [7] B. C. Smith and L. Rowe. Algorithms for manipulating compressed images. *IEEE Computer Graphics and Applications*, pages 32–42, September 1993.
- [8] M. Yong, Q. Zhu, and V. Eyuboglu. VBR transport of CBR-encoded video over ATM networks. In *Proceedings of the Sixth International Workshop on Packet Video*, September 1994.