Rate-Constrained Two-Layer Coding of H.261 Video

Vince Rhee

Department of Electrical Engineering
Texas A&M University
College Station, TX 77843-3128

Jerry D. Gibson

Department of Electrical Engineering
Texas A&M University
College Station, TX 77843-3128

Abstract

This paper describes a proposed method for two-layer coding of low-rate H.261-encoded video subject to a rate constraint. A refinement scheme is utilized where the base layer consists of H.261 video encoded at some low rate, and the refinement layer consists of the difference between the original video and the H.261-encoded video, rate constrained so that the total bit rate of the two layers equals some desired higher rate. We show that the proposed method provides significant improvement over the lower rate baseline H.261-encoded scheme, as expected, both quantitatively (in terms of an error measure) as well as qualitatively (in terms of perceived picture quality). We also show that the additional computational costs incurred are typically held to around 10-20% at the encoder, with varying costs at the decoder depending on the specific method implemented. Finally, we show that the proposed scheme offers a more graceful degradation in performance in the presence of packet loss when compared to single-layer H.261.

Keywords: *video coding; H.261; layered coding; rate-constrained coding*

1 Introduction

The combination of more powerful personal computers and increasing availability of switched digital services (such as Integrated Services Digital Network, or ISDN) has made desktop videoconferencing an attractive option. Additionally, an international video coding standard already exists in the form of ITU-T Recommendation H.261, which specifies a coding standard for rates as low as 64 kbits/s, with additional rates offered in integral multiples of 64 kbits/s [1].

The motivation for developing a layered coding scheme is as follows. Suppose we wish to carry out a desktop videoconference between end users operating at different rates. Currently, in order for all users to be able to participate, the transmission rate must be set to the lowest common rate among the users, thus wasting the additional capacity of some of the users. A layered video coding scheme would allow higher capacity users to utilize their additional available rate. All users would be sent a baseline H.261 video stream at the lowest common rate; in addition, however, using the refinement capability inherent in this scheme, those users who are capable of handling higher rates would be sent additional bits, utilizing their higher capacity to improve the received video quality.

Alternatively, such an approach would be well-suited for packet video networks such as those based on the rapidly-developing asynchronous transfer mode (ATM). The baseline H.261 packets would be marked as "high priority", while the additional refinement packets would be marked as "low priority"; in the event of network congestion, these lower priority packets would be dropped, allowing the baseline H.261 packets to be delivered and thus guaranteeing a quality of service no worse than the lower rate H.261 video.

The approach chosen to solve this problem utilizes a layered refinement of the baseline H.261 video. Specifically, the difference between the input (uncoded) video stream and the resulting H.261-encoded stream is computed. This difference is then separately quantized and is sent as a refinement bitstream, similar to schemes proposed by Ghanbari [2] and Minami [3]. However, the methods of [2, 3] are variable bit rate schemes, placing no constraints on the resulting bit rate of the refinement layer. We propose a scheme to constrain the resulting bit rate of the refinement layer to some desired value using dynamic block selection combined with rate allocation theory.

Our target is narrowband ISDN, consisting of 2×64

1058-6393/95 $4.00 © 1995 IEEE

kbits/s data channels (128 kbits/s total) [4], where the H.261 video is encoded at 64 kbits/s and is refined to a higher rate of 128 kbits/s. Additionally, we would like the resulting video quality to be comparable to that obtained by encoding the video in a single layer at the higher rate, thus presenting our layered approach as an attractive alternative to the single-layer, higher rate coding approach.

In Section 2, we outline the specific coding algorithm used. Section 3 describes the setup of our simulations. Simulation results are presented in Section 4, and our conclusions are given in Section 5.

2 Layered coding method

The general steps involved in our coding algorithm are:

1. determination of coding error
2. selection of blocks based on coding error
3. allocation of bits to selected blocks

Each of these steps is described below in more detail.

2.1 Coding error

The H.261 video coding scheme utilizes temporal prediction in order to greatly reduce the required bit rate that needs to be sent. As part of this procedure, the encoder, upon coding the input video stream, generates a locally-decoded copy of the output that would be generated by the decoder from this coded bitstream; this "local output" is then used by the encoder in its predictions for the next frame.

The proposed coding scheme also utilizes this locally-generated output. For each frame, the coding error is determined on a block-by-block basis, where the error is defined as the sum of the squared errors over the entire block, i.e. $\epsilon_{block} = \sum_{j=1}^{N} \left[f(j) - \hat{f}(j)\right]^2$, where $f(j)$ is a pixel in the original image, $\hat{f}(j)$ is the corresponding pixel in the local output image, and N is the number of pixels per block.

2.2 Dynamic block selection

After the coding errors are determined for each block in a frame, the block errors are ordered from largest to smallest. From these, the B blocks with highest error are selected for refinement; the value of B is chosen in conjunction with the number of bits b referred to in Section 2.3. Note that although the number of blocks B chosen in each frame is constant, the location of these blocks from frame to frame changes depending on the block errors within each frame. Thus, this method targets those blocks with the worst coding error.

In order to implement the coding scheme efficiently, a fast sorting algorithm is needed; the one used is patterned after the `qsort()` routine presented in [5] (The number of operations needed for the `qsort` and related algorithms is $O(n \log n)$, where n is the number of elements, making it one of the faster algorithms available.).

2.3 Rate allocation/quantization

For each block selected in Section 2.2, b bits are assigned, yielding a total number of refinement bits, Bb, equal to a desired refinement layer bit rate.

The assignment of the b bits within a block is determined via any one of a number of bit allocation rules present in the literature; the one used here is patterned after the simple rule presented in [6, pp524–528]. In order to utilize such a bit allocation rule, the variances for each position in a block must first be known. That is, for a block of size N, we compute the variances associated with each of the N positions in the block. The b bits are then assigned to these N positions, and the block is then quantized according to the aforementioned bit allocation procedure.

The necessary variances are determined by coding a number of test video sequences. The coding errors are found for each position within a block over the entire set of test sequences, and the variances are then calculated from these error values. These in turn are used to generate the desired bit allocation. Upon determining the block bit allocation from these test sequences, this allocation is then used on the actual sequence(s) to be encoded; note that the bit allocation is held fixed throughout the duration of the actual coding process.

3 Simulation

The two sequences used in our simulation are the well-known "Miss America" and "Salesman" sequences. In order to determine the refinement layer bit allocation, the necessary variances were obtained using parts of four video sequences (different from the sequences used in the simulation) as well as parts of both the "Miss America" and "Salesman" sequences.

It was decided beforehand that the frames be of CIF (352 x 288 pels) resolution in order to provide a viewing area of "reasonable" size. With this in mind, two different refinement methods were used for each of the simulation sequences, each based on the algorithm outlined in Section 2. The two methods are described below in more detail.

3.1 Method I (CIF only)

The first method performs refinement on CIF-sized frames throughout the entire coding process. The baseline H.261 video was encoded at 64 kbps, 10 frames/second. It was determined that refinement would be performed on 10% of the blocks within a frame (~160 blocks), with a target refinement layer bit rate of ~6400 bits per frame. Coding and refinement were performed in a straightforward manner as described above.

3.2 Method II (QCIF → CIF)

The second method uses QCIF-sized frames (176 × 144 pels) during the refinement process. The same number of blocks and refinement layer bits are used as in Method I; note, however, that this results in ~40% of the blocks being refined within this smaller frame size. After refinement, the frames are then upsampled to CIF resolution via a pyramidal-type expansion modelled after the method in [7].

Also note that the point spread function utilized in this expansion will result in some blurring in the resulting image. This blurring could be avoided by staying with the smaller QCIF frames; however, as pointed out above, we were targetting CIF-sized frames in the end result.

Finally, we simulated the effect of packet loss on the baseline and refined schemes. For each of the two bitstreams, corresponding to the single-layer baseline 128 kbps H.261 and the two-layer refined schemes, the resulting bit rates *after encoding* were dropped to 64 kbps (perhaps simulating the effect of a mandated drop in bit rate due to a congestion control scheme on a network), and the NMSE was calculated for the new sequences relative to the corresponding sequences without packet loss. Note that for the purposes of this simulation, packet loss was obtained by randomly dropping half the blocks in each frame of the sequences, thus bringing the resulting rate down to 64 kbps.

4 Simulation results

The simulation performance of the methods were evaluated on the basis of two numerical criteria, mean squared error and computational overhead, as well as a subjective qualitative assessment.

The [normalized] mean squared error (NMSE) is given by

$$\mathrm{NMSE} = \frac{\sum_j \left[f(j) - \hat{f}(j)\right]^2}{\sum_j f^2(j)}$$

where $f(j)$ is a pixel in the original frame of the sequence, $\hat{f}(j)$ is the corresponding pixel in the coded frame of the sequence, with the summation being carried out over all pixels in a frame. The frame-by-frame NMSE values for the two simulation sequences are plotted in Figure 1.

As can be seen from the plots, our coding process offers considerable reduction in NMSE for the "Miss America" sequence, especially for the higher motion segments of this sequence (Frames 125–150) where the reduction in NMSE can approach 75% over baseline 64 kbps H.261. The numerical results for the "Salesman" sequence do not demonstrate as marked an improvement owing to the more active and detailed frames present in this sequence.

Qualitatively, however, much more improvement is observed in both sequences. Using Method I, a large portion of the blocking artifacts present in the low-rate baseline encoded sequences are eliminated, save for those present in the more violent movements and detailed objects. In fact, for the "Miss America" sequence, there is only a small perceptual difference between the refined sequence and the H.261 sequence encoded at the higher rate (128 kbps). The "Salesman" sequence still suffers some, however; this is again due to the fact that the more detailed background cannot be efficiently coded with the low baseline bit rate, and there are an insufficient number of refinement bits present to flesh out these details.

Method II yields no detectable blocking artifacts in either of the simulation sequences. This is due, in part, to the fact that because we performed the actual refinement on QCIF-sized frames, we were able to effectively refine four times as many blocks in the resulting CIF-sized frames (after expansion) than in Method I. The effect of these "additional" bits is more apparent in the "Salesman" sequence, where much more of the background detail is present when compared with Method I. The point spreading caused by the pyramidal upsampling also helps by smoothing some of the harsher edges that may have been present; however, as mentioned above, this results in some blurring of the resulting images. The blurring is somewhat noticeable; however, the resulting sequences are still perceptually more appealing (smoother motion with no

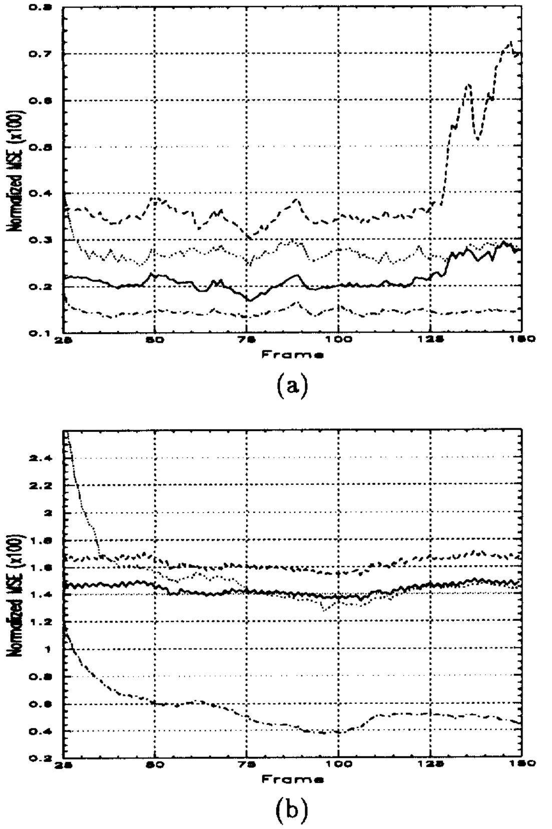

Figure 1. Normalized MSE for (a) "Miss America" and (b) "Salesman" sequences (64 kbps H.261: dashed; Method I: solid; Method II: dotted; 128 kbps H.261: dash-dot)

blocking artifacts, much more detailed background) than in Method I, even though the *numerical* performance is hampered somewhat by this blurring.

The [encoder] overhead, consisting of the time needed to select and quantize the desired refinement blocks and measured as a fraction of the CPU time needed to baseline-encode each frame, is plotted in Figure 2. As can be seen from the graphs, the encoder overhead for the "Miss America" sequence using Method I is held to around 10–11%, while the overhead in Method II is closer to 13–14% (with a slightly larger range of variation); for the "Salesman" sequence, the values are 10–11% and 18–20%, respectively.

Decoder overhead (not shown), consisting of the addition of the baseline bitstream with the refinement layer (as well as the pyramidal expansion in the case of Method II), was also calculated. For Method I, this overhead is again kept fairly low at ~15% for both sequences. However, for Method II, the expansion to CIF-sized frames is quite computationally intensive and leads to a rather large decoder overhead (~50%); this issue may have to be addressed in the future.

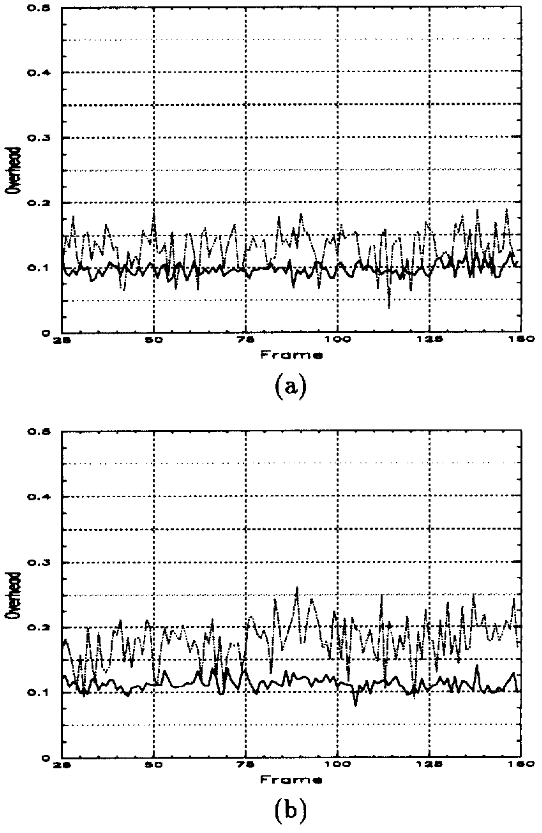

Figure 2. Encoder overhead as a fraction of baseline encoding time for (a) "Miss America" and (b) "Salesman" sequences (Method I: solid line; Method II: dotted line)

Finally, Figure 3 shows the performance of the schemes in the presence of packet loss, and it is here that some of the robustness of the refinement scheme becomes apparent. The loss of a block (packet) in the baseline 128 kbps H.261 stream is equivalent to the loss of a complete block of pixels in a frame; multiple packet losses within a single frame will lead to noticeable degradation in the resulting sequence. However, especially in the context of a congestion control scheme, block (packet) losses in the refinement scheme correspond to the loss of refinement blocks only, leaving the lower-rate [64 kbps] baseline blocks intact; this provides a more graceful degradation in performance.

These points are illustrated quite clearly in Figure 3. Notice that packet losses in the baseline 128 kbps H.261 sequence cause large, irregular variations in NMSE when compared to the same stream without packet loss. The refinement stream, even in the presence of packet losses, results in lower NMSE in general and also provides more uniform output quality (again in terms of the quantitative NMSE measure).

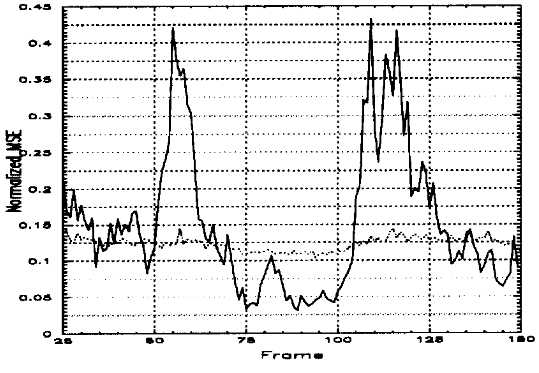

Figure 3. Normalized MSE after packet loss for "Salesman" sequences, relative to the corresponding sequences without packet loss (single-layer 128 kbps H.261: solid; two-layer 128 kbps refinement: dotted)

5 Conclusions

We have described a two-layer scheme for refining H.261-encoded video subject to a rate constraint. We have presented simulation results that illustrate the plausibility of such a scheme, as well as pointing out certain inherent weaknesses that need to be addressed. Results indicate that the proposed scheme can be effective in achieving higher overall quality, though the end results depend somewhat on the sequence in question.

Regarding the simulation results, it should be noted that although the technique in Method II tends to blur the resulting images – thus hampering the numerical performance – these sequences, when viewed, tend to be more pleasing to the eye which serves to point out that quantitative measures such as MSE are not necessarily indicative of performance when dealing with visual results. However, the better perceptual quality resulting from Method II is traded off against the decoder overhead.

The simulation also illustrates the robustness of the refinement scheme with regards to packet losses. The layered structure of our proposed scheme is well-suited to the dropping of packets (blocks); the baseline H.261 scheme, not being layered in nature, can be ill-suited in handling lost packets.

Acknowledgments

This research was supported, in part, by the Texas Advanced Technology Program under Project No. 999903–017 and by Compaq Computer Corporation through the Texas Engineering Experiment Station Program for Automation in Manufacturing.

References

[1] CCITT Recommendation H.261, CDM XV-R-37-E, *Video codec for audiovisual services at p×64 kbit/s*, December 1990.

[2] M. Ghanbari, "An adapted H.261 two-layer video codec for ATM networks," *IEEE Trans. Comm.*, Vol. 40–9, September 1992.

[3] S. Minami, "CCITT H.261 compatible mixed bit rate coding for video for ATM networks," in *IEEE ICC '92 Conference Record*, 1992.

[4] M. Schwartz, *Telecommunication Networks*. Addison-Wesley, 1987.

[5] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*. Prentice-Hall, 2nd ed., 1988.

[6] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[7] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, Vol. 31–4, April 1983.

[8] D. Minoli and R. Keinath, *Distributed Multimedia Through Broadband Communications Services*. Artech House, 1994.