ADAPTIVE VIDEO CODING FOR MOBILE WIRELESS NETWORKS

Benjamin Belzer, Judy Liao, John D. Villasenor

Electrical Engineering Department
University of California, Los Angeles
405 Hilgard Avenue
Los Angeles, CA 90024-1594

ABSTRACT

Wireless video transmission over a dynamic network requires adaptation to changes in bandwidth, network traffic, and channel characteristics. New computing hardware and algorithms are needed that enable low-power, flexible, adaptive, and robust video communication in hostile environments with no access to an installed communications infrastructure. The coding algorithms we are developing are based on subband decomposition using integer-coefficient filters, and adaptively deliver video at rates between 60K bits/sec and 600K bits/sec in accordance with the available bandwidth. Robustness in the variable length coder is obtained by using low-overhead Reed-Solomon block codes, by performing intra-frame coding only, and by using end-of-frame and end-of-subband symbols to maintain both inter- and intra-frame synchronization.

1. INTRODUCTION

Current wireless systems generally offer only voice and data, and are constrained by fixed bandwidth allocations, fixed network configurations such as cellular systems, and by a reliance on a tethered infrastructure of fixed base stations or servers that are linked by a wireline network. To overcome these constraints, hardware and algorithms for a dynamically reconfigurable network of wireless multimedia transceivers are being designed. Since video is the highest-bandwidth traffic in a multimedia environment, video compression algorithms that are efficient and rate-adaptive are crucial if efficient use of channel capacity and high connectivity are to be achieved. The video compression problem is especially acute in the wireless context because the communications channel is evolving and possibly very poor, necessitating that careful attention be made to error correction and tolerance to fades.

2. BASELINE ALGORITHM AND CONSIDERATIONS

The goals of an adaptive wireless video compression algorithm include 1) low hardware complexity, 2) adaptivity, 3) robustness, and 4) scalability. In order to address the robustness requirement and avoid inter-frame error propagation, we are implementing video compression on a frame by frame basis, without any motion compensation. Clearly, this trade-off extracts a significant (up to a factor of five) penalty in terms of compression ratio, but it limits the effects of many types of bit errors to a single frame, and also reduces the hardware complexity of the system. The basic coding scheme we are using is based on subband decomposition using wavelet transforms. The advantages of wavelet coding in terms of efficient compression are now well-established [1,2]; wavelet transforms also enable rapid rate-adaptation, and if the proper filters are used, low power consumption.

Figure 1 contains a block diagram of the baseline algorithm that is being employed. While the basic steps in this diagram are similar to those in standard compression approaches (e.g. transformation, quantization, entropy coding, error protection), each one of these steps has been modified as described below to increase its functionality in the wireless system.

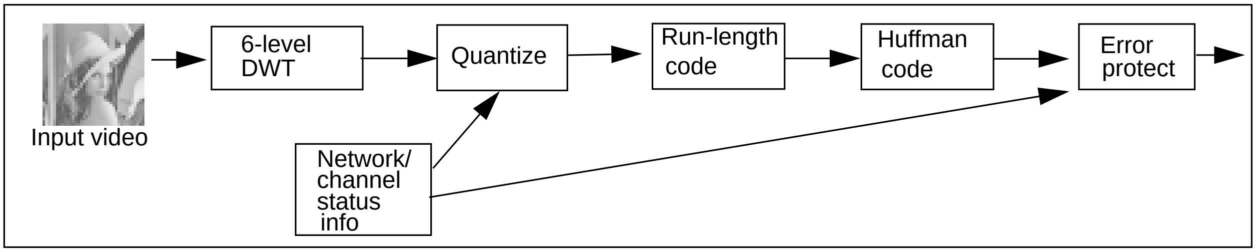
3. WAVELET FILTERS FOR LOW-POWER VIDEO CODING

Since the wavelet transform itself represents the most computationally intensive step in a subband coding algorithm, the length and nature (floating point versus integer) of the filter coefficients will have a significant effect on the overall power requirements. The filter bank experiencing the most widespread use in image compression [1] has floating point coefficients and has lengths of 7 and 9 for the low and high pass filters. While this filter bank allows very good compression, it is less well-suited to a wireless environment because of the filter lengths and because the coefficients are not integers.

In seeking a more suitable filter it is first necessary to identify appropriate evaluation criteria. Regularity [3] is commonly suggested as a criterion for filter evaluation, but there is only a partial correlation between filter regularity and reconstructed image quality. A more reliable evaluation can be based on the impulse and step responses obtained by

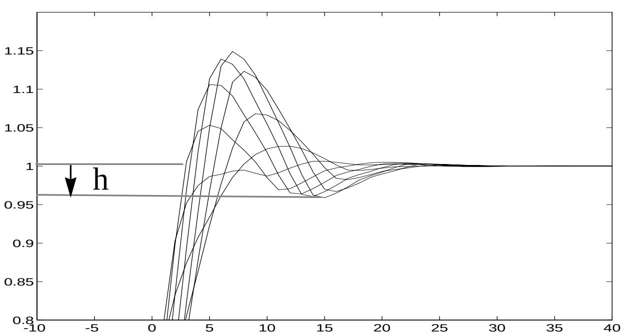
Proceedings of the 1994 IEEE International Conference on Image Processing (ICIP'94), vol. 2, pp. 972-976, Oct. 1994

Figure 1: Block diagram of basic coding system for portable, rate-adaptive wireless video transmission



considering an L-level wavelet synthesis/analysis system as a single-input, single-output linear shift-variant system with a response that varies according to the input location modulo (2^L, 2^L) (see Figure 2). Using these evaluation methods, we identified several filters that are short (a total of 8 low pass and high pass taps as opposed to 16) and have integer coefficients [4], resulting in a reduction of chip area and power consumption by a factor of three. The compression performance of these filters is only marginally less (within .5 dB PSNR) than that of the 16-tap filter bank.

Figure 2: Small values of the wavelet filter step response second sidelobe (h) lead to less edge ringing in the reconstructed image.

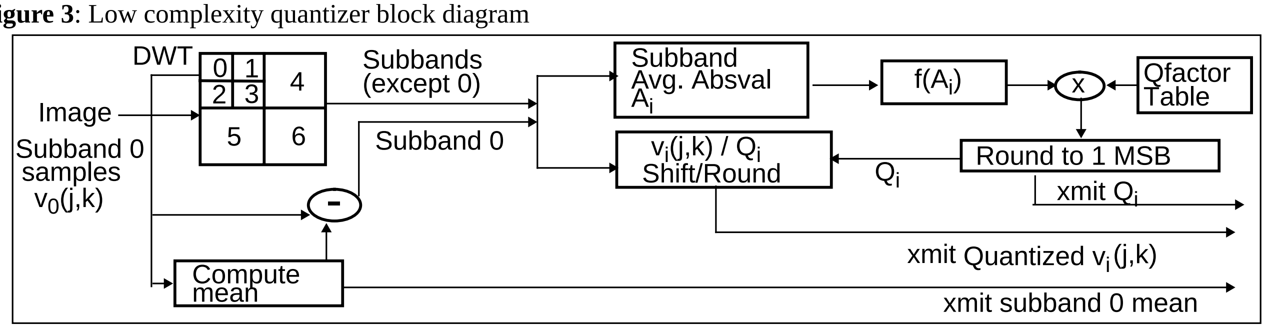


4. QUANTIZATION, ENTROPY CODING, AND ERROR PROTECTION

Quantization is one of the critical issues in a wireless video system. Some level of adaptivity is needed in order to attain reasonably low data rates; in addition it is in the quantizer that rate adaptation is most easily performed. In the initial implementation we have chosen scalar quantization, performed adaptively on a subband-specific basis based on a ‘‘Q’’-factor similar in function to that used in the JPEG standard. A block diagram of the quantizer is shown in Figure 3. The quantizer architecture is simplified for low-power implementation by choosing the quantization step sizes Q_i for each subband as a function of the subband average absolute values rather than the (theoretically optimal) subband variances. The bit-allocation function $f(A_i)$ is currently $1/\log(A_i)$ as in [5], but it can easily be changed since it is implemented via an 8 point quadratic approximation table lookup. After normalization by the user “Q-factor”, the Q_i are rounded to a power of two so that a shift rather than a multiply is used to perform the actual quantization. The resulting quantizer implementation requires only 1 add and 7 shifts per pixel, as well as 4 multiplies per subband. Additional simplifications of this architecture make it possible to remove the multipliers entirely [6]. The average reduction in image quality due to the simplifying approximations is about 0.3 dB PSNR.

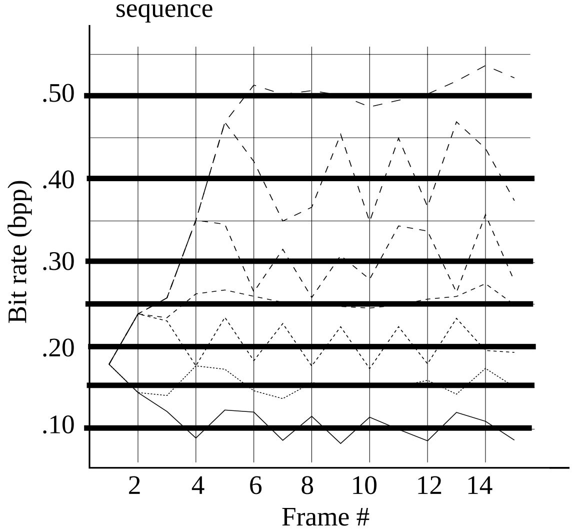
Coding rate adaptivity is achieved by varying the quantizer Q-factor, which normalizes all subband quantizer step sizes. We use a table of 32 Q-factors capable of achieving bit-rates of between 0.1 and 0.8 bpp over a wide range of 8 bit greyscale images. One problem with the Q-factor approach is that the Q-factor needed to achieve a given bit-rate is image dependent. We handle this problem by feeding back the number of bits output from the last compressed image in order to adjust the Q-factor for the image

Figure 3: Low complexity quantizer block diagram



about to be compressed. Figure 4 shows graphs of the desired vs. actual bit rates achieved by this feedback loop for the “football” video sequence.

Figure 4: Desired vs. achieved bit rate for the football sequence



The average absolute error achieved by the feedback method is about 0.02 bpp; thus, a reasonably small transmit buffer will suffice to contain fluctuations about the average. In this figure the feedback loop starts from the same point for all images and therefore takes as many as five frames to settle at its desired rate. This settling period can be greatly reduced by a simple scheme for estimating a Q-factor to within 0.1 bpp of the desired rate. The estimate is obtained by noting an approximate linear relationship between the coding rate in bpp and the inverse Q-factor, so that the Q-factor for a desired coding rate can be determined if the slope of the rate vs. inverse Q-factor line is known (Fig. 5a). This slope in turn is linearly related to the average absolute value of the highest level subbands (Fig. 5b), which is available from the adaptive quantizer. The desired Q-factor estimate is a ratio of linear expressions in the desired rate and in the average absolute value of the high level subbands. Since there are only 32 available Q-factors, the estimation can easily be implemented via a 2-dimensional table lookup.

In order to achieve the high compression ratios required for 60 kbs video, run-length coding followed by Huffman coding is used on the 18 upper level subbands of the 6 level DWT (the lowest level subband is quantized at 8 bpp). The run-length coder performs a horizontal raster scan of each high level subband and codes the subband into ordered pairs of runlength and level, similar to the JPEG standard. A single fixed Huffman code is then used to encode the run-level pairs. The two variable length codes together give lossless compression of about 6:1. However, the variable

Figure 5a: Coding rate vs. 1/Q-factor for various images

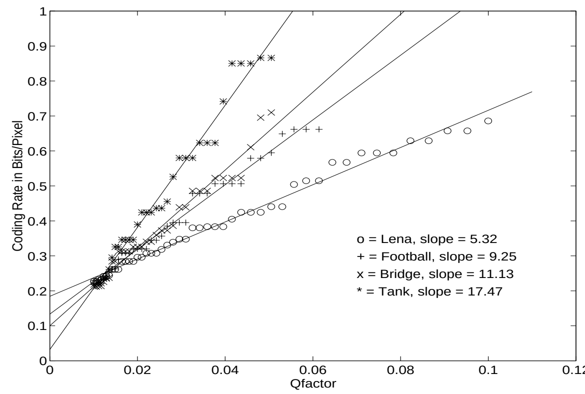
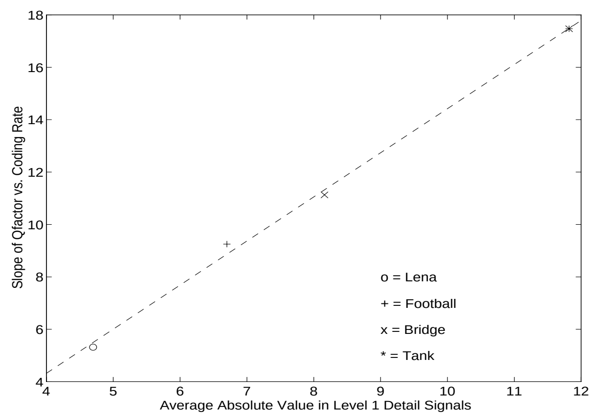
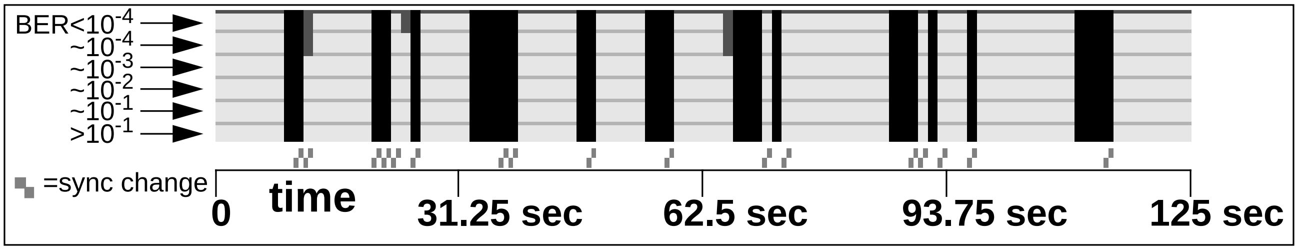


Figure 5b: Slope of the coding rate vs. 1/Q-factor curve is linearly rated to the avg. abs. value of the highest frequency subbands



length nature of these codes means that the decoder must find the next codeword boundary based on the last decoded word, which leads to codeword synchronization errors when the received bitstream is corrupted by noise. We use two methods to recover from such sync errors. First, frame sync is ensured by using an end-of-frame marker which is not variable length coded, and can be searched for by a simple digital matched filter. Second, to recover from synch errors within a frame, we use a unique End-of-Subband symbol, EOS. If the decoder detects that a subband has been filled with data and that the next codeword is not an EOS, then the decoder discards all received data until it detects the EOS symbol. Thus sync errors are limited to a given subband provided that no false EOSs are detected and that the true EOS is detected correctly. Detection of the correct EOS is aided by the self-synchronizing properties of Huffman codes [7]; if the bit error rate in the received bitstream is low enough (< 0.001) then with high probability the decoder will recover codeword sync within a small number of codewords. To ensure a low bit error rate in the received bitstream, we use a low overhead (8%) Reed-

Figure 6: Channel behavior time line for the wireless video transmission experiment



Solomon error correcting code capable of correcting 10 bytes out every 255 byte block. The compression system shown in Figure 1, including the shifted integer filters, the simplified quantizer and the above described run-length and Huffman coding, achieves 31.4 dB PSNR on the 512x512 greyscale “lena” image at 0.201 bits per pixel.

5. EXPERIMENTAL RESULTS

We have evaluated the above algorithms by transmitting the compressed bits over a direct sequence spread spectrum radio developed at UCLA [8]. The wireless prototype at UCLA is based on an all-digital direct sequence spread-spectrum (BPSK) transceiver operating at a carrier frequency of 915 MHz. The transmitted power is 8 mW and the access scheme is CDMA, with a user-selectable spreading factor of 2^n-1 where 3 <= n <= 7. Three transceivers were used in the experiments, thereby allowing exploration of the effects of interference from a third receiver. The experiments were performed indoors with a transmitter to receiver distance of 2 meters; adverse channel conditions were created by moving the receiver behind various obstacles and by using the third radio to provide a jamming signal.

For each experiment, both a 131Kbyte test sequence and a compressed video bitstream were transmitted. The test sequence was composed of a 256-byte string repeated 512 times, with a counter inserted after each occurrence of the string. Analysis of the received test sequence to determine the channel conditions was a nontrivial task because portions of the sequence were lost in transmission, and because the counter itself can be corrupted or missing. Additional complications arise because the wireless link can sometimes lose or insert a bit. Under a simple bitwise compare this would look like a massive bit error, so care must be taken in the analysis to differentiate between true bit flips and synchronization changes. The test sequence was analyzed in sections of 10K bits, representing a compromise between the desire for high time resolution on the one hand and high sensitivity in bit error rate measurements on the other hand. Since the link data rate is 8kbs, the time resolution is on the order of 1 second, consistent with the rate of channel behavior changes experienced in the experiment. Local bit error rates down to below 10^{-4} can not be measured with this approach.

Figure 6 shows a time line of the channel behavior when the receiver was periodically moved behind a metal shield, thereby obstructing line of sight transmission. The BER as a function of time is plotted on a logarithmic scale which increases from top to bottom. Also shown are markers indicating sync loss events in which bits were either lost from or inserted into the received bitstream.

The times when the receiver was behind the metal shield are clearly seen on the time line as periods of signal loss. During these periods bit loss as well as bit errors occurred; these sync loss events made it necessary to insert 32-bit uncoded start-of-codeblock markers before each 255 byte Reed-Solomon codeword. Despite these challenging channel conditions, the video codec was able to deliver intelligible video by simply freezing the last received frame during severe data loss events, and then continuing the sequence once codeword sync was regained. Examples of received video frames are shown in Figures 7a and 7b.

Figure 7a: Football sequence intraframe coded @64kb/s, 5 frames/s, received error-free

Figure 7b: Football sequence intraframe coded @64kb/s, 5 frames/s, channel BER greater than 10^{-3}

ACKNOWLEDGMENT

This work was supported by the U.S. Department of Justice/Federal Bureau of Investigation, ARPA/CSTO under Contract J-FBI-93-112, Computer Aided Design of High Performance Network Wireless Networked Systems.

REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Proc*., vol. 1, pp. 205-220, 1992.

[2] M. Vetterli and C. Herley, ''Wavelets and filter banks: Theory and design,'' *IEEE Trans. on Signal Proc.*, vol. 40, pp. 2207-2232, 1992.

[3] O. Rioul, ''Simple regularity criteria for subdivision schemes,'' *SIAM J. Math Anal.*, vol. 23, pp. 1544-76, 1992.

[4] J. Villasenor, B. Belzer, and J. Liao, ''Filter Evaluation and Selection in Wavelet Image Compression,'' presented at the 1994 IEEE Data Compression Conference, Snowbird, UT, March, 1994.

[5] T. Hopper, C. Brislawn, and J. Bradley, "WSQ Gray-Scale Fingerprint Image Compression Specification, Version 2.0", Criminal Justice Information Services, Federal Bureau of Investigation, Washington D.C., February, 1993.

[6] K. Nishibori, B. Belzer, and J. Villasenor, "A Low-Complexity Adaptive Scalar Quantizer Unit for a Wireless Video Codec", in preparation.

[7] Lelewer D.A., Hirshberg D.S., "Data Compression", *ACM Computing Surveys* 19, pp. 261-296, 1987.

[8] C. Chien et. al, "A Single-Chip 12.7 Mchips/sec All-Digital IF BPSK Direct Sequence Spread-Spectrum Transceiver in 1.2 micron CMOS", *ISSCC 1994 Digest of Technical Papers*, Feb 1994.