Low Complexity Video Coding for Teleconferencing

Stephen Neuendorffer
Nariman Farvardin

Department of Electrical Engineering
and
Institute for Systems Research
University of Maryland, College Park
College Park, MD 20742

ABSTRACT

We describe a video compression system suitable for teleconferencing applications based on Hierarchical Table-Lookup Vector Quantization and Variable Length Inter-Block Noiseless Coding (VL-IBNC). We apply IBNC selectively to utilize both the inter-frame and intra-frame correlation in the video signal. This structure has low encoding and decoding complexity and low bit-rate performance comparable to more complex algorithms.

INTRODUCTION

Video teleconferencing is an area which has sparked several commercial products in recent years. Unfortunately these systems suffer from two main drawbacks: either they require expensive hardware, or they suffer from inferior transmitted image quality. This research is an attempt to develop a low complexity encoding algorithm that can run in real-time on a readily available, general purpose, hardware platform, the desktop personal computer.

Vector Quantization (VQ) has been used for source coding of both speech and video signals [1]. VQ is a lossy compression scheme where each input block (or vector) is replaced by a reproduction vector from a codebook that most closely resembles it. Once this minimum distortion reproduction vector is found, we can approximate the original signal by only storing or transmitting the appropriate index into the codebook. If $\mathbf{B}$ is an N-dimensional input vector, and $\mathbf{C}_k$ is the k'th vector in the codebook, then the codeword

$$I = \underset{k}{\operatorname{argmin}} \left(d(\mathbf{B}, \mathbf{C}_k) \right)$$

where $d(\mathbf{X},\mathbf{Y})$ is a distortion measure. A commonly used distortion measure (and the one used in this paper) is the sum of squares distortion measure

$$d(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^{N} (X_k - Y_k)^2$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_N)$. VQ codebooks are usually designed by applying the Generalized Lloyd Algorithm (GLA), which iteratively attempts to find a local minimum in the total distortion caused by encoding a training sequence [1]. Some other algorithms, such as simulated annealing may come closer to a global minimum of the total distortion, but in general it is a computationally formidable problem to find a globally optimum VQ codebook.

Unfortunately, encoding for an unstructured vector quantizer involves a full search of the codebook which amounts to calculating the distortion between each input vector and all codevectors in the codebook. This is a very time consuming process. Ideally, it would be nice in many applications to precompute the optimum codevector for every possible input vector and access them as a table lookup. However, for even modest codebook sizes and vector dimensions, such tables become unmanageably large. Hierarchical Table-Lookup Vector Quantization (HTVQ) attempts to approximate such a system, by encoding the input vector using a series of table-lookups, thus greatly reducing complexity [2].

An M-stage HTVQ consists of a series of M tables, where table i ($1 \le i \le M$) takes k_i input samples and produces one output sample. The input samples for the first stage are taken directly from elements from the input vector B, which have previously been scalar quantized to a finite number of levels. The input for each succeeding stage i is then taken from the outputs of stage i-1. The output of stage M is the encoded codebook index. Although the minimum distortion codevector is determined

*Prepared through collaborative participation in the Advanced Telecommunications & Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

at each stage, the final codevector is not necessarily the minimum distortion reproduction vector, since the quantization is performed successively. For simplicity and to minimize the storage requirements, the number of input samples to each table, k_i, is usually two. This M-stage HTVQ then approximates a 2^M-dimensional full-search VQ.

An HTVQ can be designed with any algorithm that will design a VQ. However, it is necessary to design a separate VQ for each stage of the HTVQ. The HTVQ tables for each stage are then created by calculating the minimum distortion codevector for each combination of possible inputs for that stage. For example if we wish to design an 8 dimensional HTVQ, with k_i=2, we will first design three VQ's with 2, 4, and 8 dimensions respectively. Then we design the HTVQ tables, which correspond to three 2-dimensional VQ's. It is important to note that the characteristics of the VQ for the final stage determine the characteristics of the entire HTVQ. HTVQ has also been successfully applied to video coding [3].

Inter-Block Noiseless Coding (IBNC) can be used to further reduce the encoding rate, if the VQ is designed as a Tree-Structured VQ (TSVQ) [4] [5]. In a TSVQ, the VQ indexes can be expressed as paths in a binary search tree, with the reproduction vectors as leaves of the tree. Thus reproduction vectors that are close in a tree-traversal sense (Hamming distance between VQ indexes) are also close in a distortion sense (Euclidean distance between codevectors). We can then define a codeword I_2, relative to another codeword I_1, by considering their relationship within this binary search tree.

A TSVQ is most easily designed by using the splitting algorithm. In this algorithm, instead of designing the whole codebook at once, we begin by designing a two vector codebook C^1 using the Lloyd iteration. These two vectors will partition the input space into two subspaces: S_0^1 and S_1^1. For each subspace we again design a two vector codebook. The codebook C^2, which is the union of the codebooks created from the S_k^1 subspaces, contains four codevectors and is a codebook on the input space. We can repeat this process M times to create a codebook of size 2^M. Each codevector in the final codebook C^M is then contained in subspaces $S_{x_1}^1, S_{x_2}^2, \ldots, S_{x_M}^M$ and we assign it the index I, where the most significant bit of I is x_1 mod 2 and the least significant bit is x_M mod 2.

We define the prefix length two codewords $L_p(I_1, I_2)$ to be the length of the path in the tree shared by indexes I_1 and I_2 and the suffix path $S(I_1, I_2)$ to be the path that is unique to I_2. Thus I_2 can be completely described relative to I_1 in terms of $L_p(I_1, I_2)$ and $S(I_1, I_2)$. We also define a prefix length event, designated as $D_k(I_i, I_j)$, to be the event that $L_p(I_i, I_j)$=k. IBNC consists of sending the prefix lengths and suffix paths for each block, relative to a previously encoded block in the video signal. Each event D_k can be encoded either with a fixed length code (resulting in Fixed-Length Inter-Block Noiseless Coding (FL-IBNC)) or with a variable length code (termed Variable-Length IBNC (VL-IBNC)).

This framework allows us to utilize either the spatial redundancy within a single frame or the temporal redundancy between frames. Let $X_n(i, j)$ represent the VQ encoded index of the i'th block in the j'th column of the n'th frame of a video signal. Let 2^K be the total number of codevectors in the the codebook and K is the number of bits needed to represent $X_n(i, j)$. In cases where spatial redundancy predominates, we use intra-frame IBNC between $X_n(i, j)$ and $X_n(i, j-1)$. In this case we are only concerned with the set of nine prefix length events given by:

$$\bigcup_{k=0}^{K} D_k\left(X_n(i-1, j), X_n(i, j)\right).$$

However, in cases where a large temporal redundancy exists, it might be preferable to use inter-frame IBNC between $X_n(i, j)$ and $X_{n-1}(i, j)$. The prefix length event set for inter-frame IBNC can similarly be written as

$$\bigcup_{k=0}^{K} D_k\left(X_{n-1}(i, j), X_n(i, j)\right).$$

We also notice the effect of certain video signals on the bitrate of both inter-frame and intra-frame IBNC. Firstly, intra-frame Coding performs very well only when a frame contains large solid areas, and very poorly only when the frame contains no solid areas. Video signals in teleconferencing applications usually contain some large solid areas (such as a background) and some non-solid areas (such as a speaker's face). The inter-frame method on the other hand, performs well when the current frame is very similar to the previous frame, and poorly when the two are very different. The inter-frame method will perform well on image sequences with little motion (and large inter-frame correlation), regardless of the content of the individual images. However, the inter-frame method performs poorly on sequences with a large motion component (and small inter-frame correlation).

Video signals in teleconferencing applications generally

contain different areas of content. Some areas which contain motion, such as the speakers face and hands, would be better applied to the intra-frame scheme. Other areas, such as a background which contains almost no motion, would be better suited to the inter-frame approach.

This paper discusses an attempt to produce a low bitrate, motion invariant video coding scheme combining inter-frame and intra-frame IBNC. Our goal is to implement a video-teleconferencing algorithm in software on general purpose processor, while not sacrificing the image size or quality available in more expensive systems.

CODING ALGORITHM

We propose a mixed-frame extension of IBNC, which consists of inter-frame IBNC with an extra prefix code identifying to the case that $X_n(i-1, j)=X_n(i, j)$ but $X_{n-1}(i, j)\neq X_n(i, j)$. The 10 mixed-frame IBNC prefix length codes are thus

$$\bigcup_{k=0}^{K} D_k\left(X_{n-1}(i,j), X_n(i,j)\right) \cup D_K\left(X_n(i-1,j), X_n(i,j)\right)$$

The added intra-frame component does not significantly increase the bitrate in sequences with large inter-frame correlation. However, in sequences with little inter-frame correlation on which inter-frame IBNC performs poorly, the bitrate of mixed-frame IBNC approaches the bitrate of intra-frame IBNC plus overhead.

CODER DESIGN

Our current coder is based on grayscale QCIF video (176x144, 8 bits per pixel, 15 frames per second). The target of this coder is a real-time video-teleconferencing system running in software on a Pentium-166 MHz desktop machine. Figure 1 contains a block diagram of this encoder. The first encoding step consists of a 4x2 dimensional Tree-Structured Vector Quantizer. This TSVQ is implemented with a 3-stage hierarchical table-lookup structure. The codebook for each stage contains 256 codevectors. Stages one and two are designed using the GLA, while the last stage was designed as a TSVQ using the splitting algorithm.

The second encoding step consists of the Inter-Block Noiseless code. We compare inter-frame, intra-frame and mixed-frame IBNC, each combined with a 4 bit Fixed-Length IBNC, and a Variable-Length IBNC based on first-order Huffman coding. This Huffman code is statically designed from the "Miss America" sequence, and is computed sparately for each scheme.

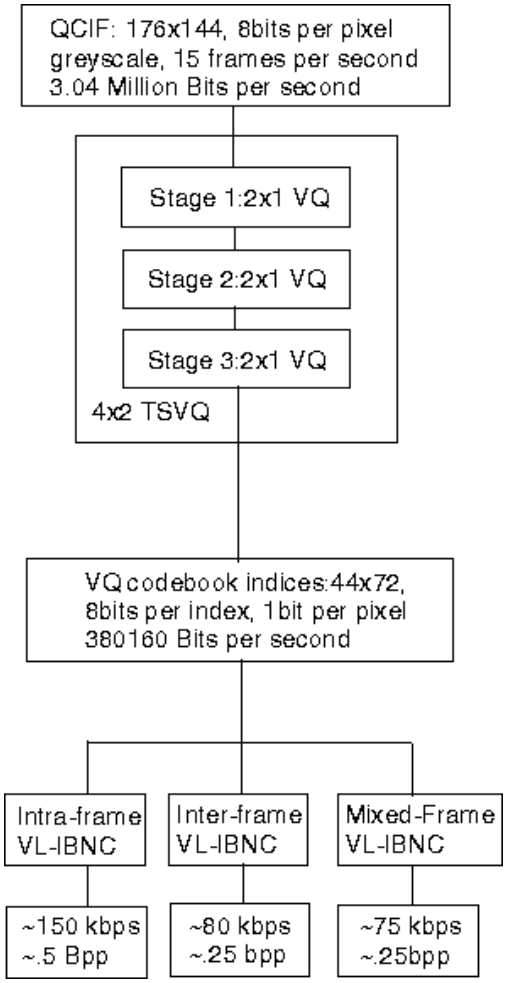

Figure 1:Encoder Block Diagram

RESULTS

These measurements are based on the first 100 frames of the "Claire" sequence. Table 1 summarizes the average encoding rate in kilobits per second for different schemes. In all cases, the average peak signal to noise ratio (PSNR) is 29.2 dB.

Table 1: Average Rate

	Intra-frame	Inter-frame	Mixed-frame
FL-IBNC	248 Kbps	207 Kbps	202 Kbps
VL-IBNC	142 Kbps	79 Kbps	75 Kbps

Since intra-frame coding does not attempt to utilize any temporal redundancy, it performs relatively poorly in sequences with little motion (such as "Claire"). On the other hand, inter-frame coding relies solely on the similarity between frames and performs well on "Claire".

The mixed-frame scheme performs comparably to inter-frame coding on Claire. The advantage of mixed-frame encoding can be seen in Figure 2. As the amount of motion increases towards the right, mixed-frame encoding actually performs much better than inter-frame coding alone. VL-IBNC also meshes well with mixed-frame coding. VL-IBNC tends to perform better with inter-frame coding than with intra-frame coding. In inter-frame coding, prefix length events $D_0(X_{n-1}(i, j),X_n(i, j))$ and $D_8(X_{n-1}(i, j),X_n(i, j))$ are very common, resulting in a low prefix length entropy. These events roughly correspond to cases where objects are moving, or objects are staying still. With intra-frame coding, color gradients tend to increase the frequency of "moderate" prefix lengths, resulting in a higher prefix length entropy. However, mixed-frame coding tends to preserve the low entropy of inter-frame coding by producing a high probability of $D_0(X_{n-1}(i, j),X_n(i, j))$, $D_8(X_{n-1}(i, j),X_n(i, j))$, and $D_8(X_n(i-1, j),X_n(i, j))$.

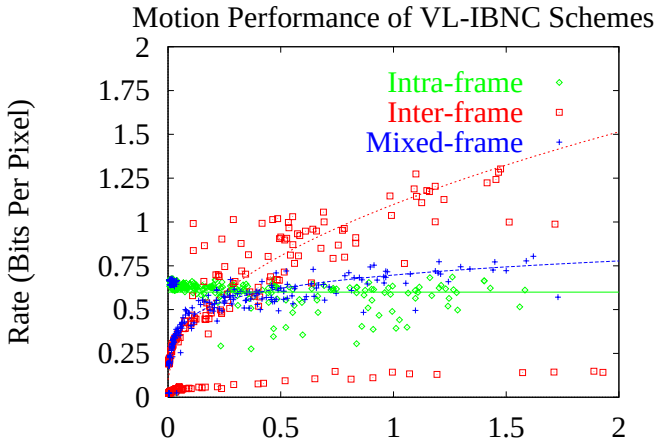

Figure 2.

MOTION PERFORMANCE

In Figure 2 we graph encoded rate versus motion for a large number of frames from a variety of video sequences. We measure the motion between two frames by finding the total squared difference E between the unencoded frames (which have already been scalar quantized to 8 bits per pixel grayscale images). If $\mathbf{B}_n(i, j)$ is the (i,j)'th block in the n'th frame, then $E = \sum_{(i,j)} d\left(\mathbf{B}_n(i, j),\mathbf{B}_{n-1}(i, j)\right)$.

Since we are taking the difference between two frames in a sequence, E roughly corresponds to average amount of motion over the time period between the frames. In addition, since the frame rate is fast relative to the motion that is occurring, we can call E an instantaneous measure of the motion within the sequence. The bit rate is calculated as the average number of encoded bits per pixel for frame $\mathbf{B}_n(i, j)$ and does not include the encoded size of the previous frame for inter-frame and mixed-frame codes.

The values in Figure 2 have a relatively wide variance from the characteristic rate versus motion curves. This spread is caused by the vector quantization error, since the amount of motion is calculated before the VQ stage, but the IBNC is calculated afterwords. Especially noticeable are the samples across the bottom of the graph for the inter-frame scheme. These samples correspond to consecutive frames that contain a fair amount of total squared difference, but not enough difference within each block of the image to change the VQ index for that block. In other words, motion is spread across the image, instead of being concentrated in small areas.

Notice that the performance of intra-frame IBNC does not depend on the amount of motion in the sequence. This is as expected, since the intra-frame technique completely ignores any inter-frame correlation. It is also important to notice how the rate of the inter-frame technique is very low when there is little motion, but grows large quickly as the motion increases. The proposed mixed-frame method combines the best characteristics of both inter- and intra-frame IBNC. The mixed-frame scheme performs almost identically to inter-frame IBNC in sequences with little motion, but approaches a maximum rate as the amount of motion increases.

CONCLUSION

We have implemented a low-complexity video coding system operating in real-time on a standard desktop personal computer. The combination of HTVQ and mixed-frame VL-IBNC is computationally simple and yet has good rate-distortion performance at low bit-rates. We have also demonstrated a video transmission system over a packet network using this coding scheme. Current research is concentrating on adaptively determining an optimal set of prefix length events for encoding each frame. Simple transform coding could also be used during HTVQ to improve image quality, as in [3]. We also intend to explore adaptive schemes to encode to a fixed bit rate, aimed eventually at 57.6 Kbps ISDN.

REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, 1992.

[2] P.C. Chang, J. May, and R. M. Gray, "Hierarchical Vector Quantizers with Table-Lookup Encoders," *Int. Conf. Acoustics Speech and Signal Proc.*, pp. 1452-1455, 1985.
[3] M. Vishwanath and P. Chou, "An Efficient Algorithm for Hierarchical Compression of Video," *Int. Conf. Image Proc.*, vol. 3, pp. 275-279, Nov. 1994.
[4] N. Chaddha, P. Chou and R. M. Gray, "Constrained and Recursive Hierarchical Table-Lookup Vector Quantization," *Proc. Data Compression Conf.*, Snowbird, UT, pp. 220-229, Apr. 1996.
[5] D. L. Neuhoff and N. Moayeri, "Tree-Searched Vector Quantization with Interblock Noiseless Coding,"

*The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government.