VIDEO CODING USING PERCEPTUALLY WEIGHTED VECTOR ZEROTREES AND ADAPTIVE VECTOR QUANTIZATION

James E. Fowler
Department of Electrical and Computer Engineering
Mississippi State University, Starkville, Mississippi

ABSTRACT

A new system for intraframe coding of video is described. This system combines zerotrees of vectors of wavelet coefficients and the generalized-threshold-replenishment (GTR) technique for adaptive vector quantization (AVQ). A data structure, the vector zerotree (VZT), is introduced to identify trees of insignificant vectors, i.e., those vectors of wavelet coefficients in a dyadic subband decomposition that are to be coded as zero. GTR coders are then applied to each subband to efficiently code the significant vectors by way of adapting to their changing statistics. Both VZT generation and GTR coding are based upon minimization of criteria involving both rate and distortion. In addition, perceptual performance is improved by invoking simple, perceptually motivated weighting in both the VZT and the GTR coders. Our experimental findings indicate that the described VZTGTR system handles dramatic changes in image statistics, such as those due to a scene change, more efficiently than a scalar zerotree technique employing a nonadaptive scalar quantizer.

1. INTRODUCTION

A recent flurry of published results has demonstrated that wavelets yield excellent performance for subband image coding. The embedded-zerotree-wavelet (EZW) algorithm by Shapiro [1], as well as variants such as space-frequency quantization (SFQ) by Xiong *et al*. [2], have, by reducing redundancy among wavelet coefficients with tree-based prediction structures (zerotrees), broken previous performance barriers and dramatically advanced the state of the art in still-image coding. In this paper, we extend the concept of zerotrees to vectors, allowing us to capitalize on recently developed techniques [3, 4] for efficient adaptive vector quantization (AVQ).

Specifically, we present here a system for the intraframe coding of video sequences which extends SFQ [2], a rate-distortion-based scalar zerotree algorithm, to vectors and then employs the generalized-threshold-replenishment (GTR) AVQ algorithm [3, 4] for efficient coding of vectors of wavelet coefficients. Our vector-zerotree (VZT) structure efficiently describes which "insignificant" vectors are not to be coded while our GTR coders process the remaining "significant" vectors, all the while adapting to changing statistics of these vectors. In addition, we improve perceptual performance by invoking simple, perceptually motivated weighting in both the creation of the VZT and in the GTR coding.

We are motivated to "vectorize" the zerotree concept for a number of reasons. Primarily, rate-distortion theory dictates that quantization of vectors is more efficient than scalar quantization. Secondly, creating zerotrees of vectors significantly reduces the number of nodes in the trees as compared to the scalar case; consequently, we expect that the burden of side information needed to represent the VZT structures will be significantly less. Finally, although nonadaptive quantization has been successfully applied to the coding of many types of data, including speech, audio, images, and video, such sources can rarely be assumed to be stationary in practice. On the other hand, AVQ in general, and GTR in particular, has been shown to achieve efficient rate-distortion performance for nonstationary sources [4]. Below, we demonstrate that our GTR coders operate more efficiently than the static, scalar quantizers employed by SFQ and other zerotree techniques when the statistics of a video stream change dramatically due to a scene change.

In the following, we overview our proposed VZTGTR video-coding system and describe our implementation of VZT structures and GTR coding for this system. We conclude with some experimental results comparing the performance of our VZTGTR system to that of SFQ on an image sequence containing a scene change.

2. THE VIDEO-CODING SYSTEM

Our VZTGTR video-coding system is depicted in Fig. 1. Our system uses a three-level, dyadic subband decomposition employing the 9/7 wavelet filter described in [5]. The lowpass subband (baseband) is coded independently using scalar DPCM followed by a uniform scalar quantizer and arithmetic entropy coding.

The VZT structure used in our system is similar to the zerotree data structure used in [2], which in turn has its origins in the classic EZW technique developed in [1]. The key structural difference between our VZT and the zerotrees of

This material is based upon work supported by the National Science Foundation under Grant No. INT-9600260. Much of the work was performed while the author held a postdoctoral research position at Laboratoire I3S, Université de Nice-Sophia Antipolis, Sophia Antipolis, France.

[1, 2] is that our VZT is constructed for square vectors of wavelet coefficients rather than scalar values. For the case of 2×2 vectors, each 2×2 vector of wavelet coefficients at decomposition level $l > 1$ of the subband decomposition has four children vectors at level $l - 1$; these children vectors are also size 2×2. Node j at level l, n_{jl}, of our VZT corresponds to a vector i in subband s, v_{is}, of the subband decomposition. Node n_{jl} holds one of two values: S to indicate that each of the four children of vector v_{is} are significant (symbols S *or* Z), or Z to indicate that v_{is} is a vector-zerotree root. The occurrence of a vector-zerotree root in the VZT structure indicates that we will not code any of the descendant vectors of v_{is} (although we will code vector v_{is} itself).

2.1. Vector-Zerotree Pruning

We determine a new VZT structure for each frame of input video by starting with a full tree (i.e., a VZT structure with all nodes labeled S), and "carving" out the VZT over several iterations of a pruning algorithm. Our VZT-pruning algorithm estimates the best VZT given the set of GTR codebooks produced while coding the previous frame. The VZT pruning is based on a rate-distortion criterion that determines the best VZT tree considering the cost, in terms of distortion, of coding sets of vectors as zero (as is implied by the occurrence of a zerotree root) versus the gain, in terms of rate, of not coding a zerotree of vectors. Additionally, our VZT-pruning algorithm compensates for the fact that the sensitivity of human vision to image distortion is highly dependent on the subband in which the distortion occurs. We note that our VZT-pruning algorithm differs from the similar approach presented in [2] in that entropy-constrained VQ (ECVQ) [6] replaces the uniform scalar quantization used in [2], and we modify the rate-distortion criterion to normalize distortion measures with perceptual weighting. Fig. 2 shows our VZT-pruning algorithm in detail.

The perceptual weights, α_s, serve to "normalize" each distortion calculation in view that the perceptual effect of distortion on image quality varies significantly from subband to subband. Use of these perceptual weights is similar to the approach taken in [7] in which values for just-noticeable distortion (JND) were determined following perceptual experiments for base sensitivity: random noise was added to a mid-gray background in each subband and the variance of the noise was increased until the noise was just noticeable against the background. The perceptual weights, α_s, used in our video-coding system are the reciprocals of the JND values reported in [7]. Although in general, it would be possible for the values of α_s to vary from frame to frame, we currently fix the α_s values for all frames.

2.2. Adaptive Vector Quantization with GTR

The vectors that are not "pruned" as indicated by the newly determined VZT are passed to a set of GTR coders. GTR, an online AVQ algorithm based on rate-distortion criteria, has been discussed extensively elsewhere [3, 4], so only a brief review will be presented here.

The GTR algorithm operates as follows. For current vector v_{is}, the rate-distortion-based nearest neighbor is calculated to minimize $J = D(c) + \gamma R(c)$, where codeword c is in codebook $\mathcal{C}_s$, $D(c)$ is the distortion between v_{is} and c, and $R(c) = -\log_2 p_{cs}$ is a probability-based rate estimate. Once the nearest neighbor, c^*, is determined, the algorithm decides whether to update the codebook with the current vector. The update cost function, $\Delta J = \Delta D + \gamma \Delta R$ is calculated, where ΔD is the potential gain in distortion due to an update, ΔR is the cost in side information of the update, and γ is a Laplacian constant dictating the tradeoff between rate and distortion in the GTR coder. If $\Delta J < 0$, v_{is} is added to the codebook; otherwise, merely the index of c^* is sent to the decoder.

We have made a few modifications to the GTR algorithm as originally presented in [3, 4]. In our VZTGTR system, each GTR coder starts coding an input video sequence with a null codebook, i.e., a codebook with no codewords. The codebook is then populated through codebook updates until a maximum of 256 codewords is reached, after which each codebook update necessitates the removal of an existing codeword. This codeword removal is accomplished via the move-to-front variant of the GTR algorithm as described in [3, 4]. Additionally, we incorporate perceptual weighting in each GTR coder; i.e., in the GTR coder for subband s, we use $\gamma = \lambda_s/\alpha_s$, where λ_s and α_s are, respectively, the rate-distortion and perceptual-weight parameters used previously in VZT pruning.

Each GTR coder produces a sequence of VQ indices as well as side information. Each VQ-index sequence is entropy coded independently, producing a separate bitstream for each subband. The side information from each GTR coder consists of a map of binary flags indicating, for each vector coded in the subband, whether an update occurs, and, in the case of an update, the vector components of the updated vectors. The side information for all subbands is multiplexed together and combined with the VZT information. The update-vector components are coded using a uniform scalar quantizer followed by entropy coding. The side-information flags are combined with the VZT symbols, resulting in an stream of symbols from a four-symbol alphabet. This symbol stream is entropy coded.

3. EXPERIMENTAL RESULTS

In this section, we describe experiments conducted to compare the performance of our VZTGTR system with that of the SFQ algorithm [2]. We specifically consider the situation in which the statistics of an image sequence change dramatically due to a scene change. To simulate the scene change, we use a 200-frame image sequence composed of 125 frames from the "Football" sequence followed by 75 frames from the "Susie" sequence. This test sequence is grayscale with a spatial resolution of 352×240 pixels and a temporal sampling of 30 frames/sec. (noninterlaced). We arrange the experiment so that both the VZTGTR system and the SFQ algorithm code the initial 125 frames (the "Football" portion) of the sequence at target bit rate of 0.5 bits/pixel and then examine performance after the scene change. We note that all rate values are calculated from "real" bitstreams produced by arithmetic coders.

Since the SFQ algorithm was originally designed to code single images, we apply SFQ in a frame-by-frame fashion for the results here. The original description of SFQ called for an exhaustive search of a number of scalar-quantizer step sizes to find the best one for a particular image. Since such a search is too computationally expensive to perform for each image of a video sequence, we fix the SFQ scalar quantizer to a preselected step size, $q = 31.5$, determined by the above mentioned exhaustive search on the first frame of the sequence. This step size is used to code the entire sequence. We adjust by trial and error the SFQ rate-distortion parameter, λ, so that the rate, averaged over the initial 125 frames with q fixed at 31.5, is approximately 0.5 bits/pixel; we use this λ for the entire sequence.

Our VZTGTR system uses a uniform scalar quantizer to code the components of the vectors that update the codebook. This step size ($q = 50$ was used) was chosen so as to provide the best rate-distortion performance over the initial 125 frames of the test sequence. Using this q, we adjust by trial and error the rate-distortion parameter λ so that the rate, averaged over the first 125 frames of the sequence, is approximately 0.5 bits/pixel. This λ is then used in the coding of each subband in each image of the test sequence.

The above procedures can be considered to have "optimized" the operation of the two coding techniques to the "Football" portion of our test sequence. That is, the scalar-quantizer step sizes, q, and the rate-distortion parameters, λ, of each technique have been selected to provide optimal distortion performance for an average rate of 0.5 bits/pixel over the "Football" portion of the test sequence. However, these parameters are also used over the "Susie" portion of the test sequence; these latter frames have statistics significantly different from those of the initial "Football" portion. The resulting rate-distortion performance of the two techniques is given in Fig. 3. Frames from both the "Football" and "Susie" portions of the reconstructed output sequences are presented in Fig. 4 for both techniques.

4. DISCUSSION AND CONCLUSIONS

In Fig. 3, we observe that VZTGTR and SFQ achieve very similar rate-distortion performance over the "Football" portion of the test sequence. However, the VZTGTR system achieves significantly superior performance after the scene change—over most of the "Susie" portion of the sequence, VZTGTR achieves slightly greater PSNR at a substantially lower bit rate. From Fig. 4, a similar conclusion can be drawn regarding the visual quality of the two techniques. Over the "Football" portion of the sequence, the visual quality achieved by the two algorithms is nearly identical. However, the VZTGTR system maintains a better looking image after the scene change. In particular, the VZTGTR coder gives better reproduction of edges and areas of detail.

The primary task of a video-coding system is to maintain consistent visual quality at the decoder for the entire sequence. The key difficulty in applying many image-coding algorithms to this task is the selection of algorithm parameters. Even if it is possible to select, *a priori*, parameters yielding suitable performance over one portion of an image sequence, dramatic shifts in statistics due to scene changes inevitably require some form of adaption.

For instance, the performance of the SFQ algorithm is closely tied to the scalar-quantizer step size. However, to code a single frame of our test sequence (resolution 352×240) by first determining the optimal step size using the exhaustive search outlined in [2], our implementation of SFQ requires about 106 secs. of computation on a Pentium II, 266MHz, 128Mb. The coding of this same frame with the step size already specified takes only 0.92 secs. Determining an optimal step size for each image of a video sequence is clearly infeasible from a computational standpoint. It is also unwarranted—our observations indicate that the optimal step size often changes little over a single scene. It is when there is a sudden change in image statistics, i.e., a scene change, that a new scalar-quantizer step size is needed; otherwise, rate-distortion performance, as well as visual quality, will suffer, as is evidenced in Figs. 3 and 4.

By adding AVQ coders to a rate-distortion-based zerotree framework, the VZTGTR system incorporates into SFQ an adaption mechanism necessary for efficiently handling scene changes. The VZTGTR system adds vectors to the codebooks as needed to satisfy rate-distortion criteria. In the experiments outlined above, this codebook updating occurs for an average of 1.7% of the vectors in each frame of the test sequence, while the bits required to transmit the update vectors to the decoder account for only about 15% of the total bit rate. When the scalar quantizer used to code the update vectors is mismatched to the source as is the case during the latter frames of the test sequence, the rate-distortion performance of the VZTGTR system suffers. However, the resulting inefficiency is much less than that incurred by SFQ, whose scalar quantizer, used in the coding of 100% of the wavelet coefficients, is much more crucial to the rate-distortion performance of the algorithm. In addition to superior rate-distortion performance, the VZTGTR also produces better perceptual quality for the latter portion of the test sequence. Our observations indicate that this superior perceptual performance is due to both the perceptual weightings present in the VZTGTR system as well as to the GTR coders which tend to preserve edges and other areas of high detail [4]. Finally, we note that our current (nonoptimized) implementation of the VZTGTR system takes about 1.8 secs. to process a single frame of the test sequence.

In concluding, we make several remarks concerning issues open to future work. First, we note that our VZTGTR system provides natural priority partitioning of the coded bitstream not present in the other video coding methods such as MPEG. For transmission over priority-capable asynchronous networks, we anticipate increasing resilience to cell-loss by sending baseband and side-information data streams at highest priority while sending the highpass subbands with decreasing priorities based on their respective location in the subband tree. Our future plans include the investigation of the performance of VZTGTR over such priority-based transmission under cell-loss conditions. In addition, we note that we have restricted our experiments here to intraframe coding as an efficient intraframe technique is the basis for successful motion-compensated approaches. The incorporation of motion compensation to our VZTGTR system in such a way as to remain resilient to cell loss is nontrivial and remains a topic for future investigation. Finally, the λ_s parameters used in the VZTGTR system determine a balance between rate and distortion achieved by the system; in the experiments presented here, this balance is constant across the entire sequence. However, as stated above, the true aim should be to maintain consistent visual quality over the sequence. We anticipate that allowing time-varying and subband-varying λ_s values will help achieve this goal and plan on incorporating such mechanisms into the VZTGTR system at a later date.

5. REFERENCES

[1] J. M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients,” *IEEE Trans. Signal Proc.*, vol. 41, pp. 3445–3462, Dec. 1993.

[2] Z. Xiong, K. Ramchandran, and M. T. Orchard, “Space-Frequency Quantization for Wavelet Image Coding,” *IEEE Trans. Image Proc.*, vol. 6, pp. 677–693, May 1997.

[3] J. E. Fowler and S. C. Ahalt, “Adaptive Vector Quantization Using Generalized Threshold Replenishment,” in *Proc. IEEE Data Compression Conf.*, pp. 317–326, Mar. 1997.

[4] J. E. Fowler, “Generalized Threshold Replenishment: An Adaptive Vector Quantization Algorithm for the Coding of Nonstationary Sources,” *IEEE Trans. Image Proc.*, 1998. to appear.

[5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image Coding Using Wavelet Transform,” *IEEE Trans. Image Proc.*, vol. 1, pp. 205–220, Apr. 1992.

[6] P. A. Chou, T. Lookabaugh, and R. M. Gray, “Entropy-Constrained Vector Quantization,” *IEEE Trans. ASSP*, vol. 37, pp. 31–42, Jan. 1989.

[7] I. Höntsch, L. J. Karam, and R. J. Safranek, “A Perceptually Tuned Embedded Zerotree Image Coder,” in *Proc. ICIP*, pp. 41–44, Oct. 1997.

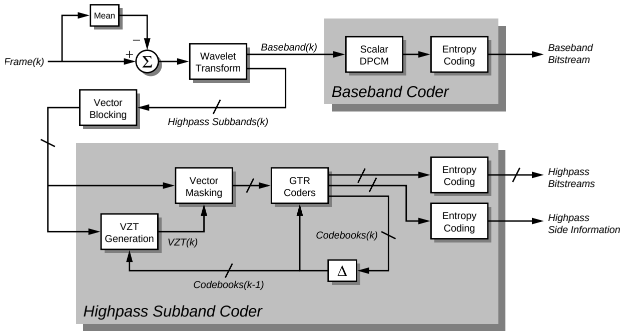
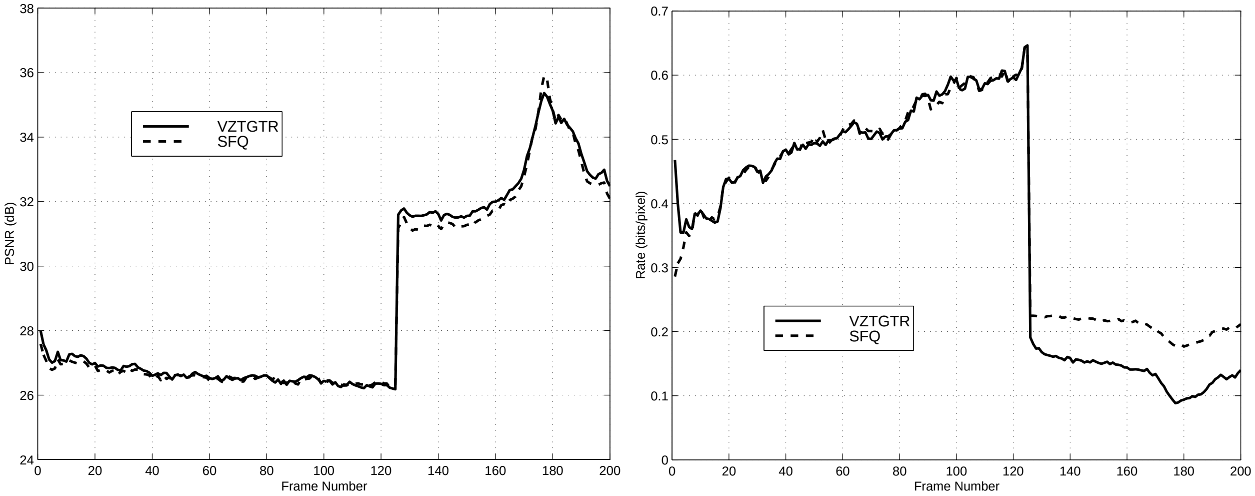


Figure 1: Block diagram of the VZTGTR video-coding system. The block labeled with Δ indicates a frame delay. Lines labeled with a slash indicate multiple data paths (one for each subband).

```
Given: current frame = $k$
       initial VZT $V^{(0)}$, initialized to full tree
       VQ codebooks $\mathcal{C}_s$ from frame $k-1$
       initial probabilities $p_{cs}^{(0)}$ of codewords $c \in \mathcal{C}_s$
       set of rate-distortion parameters, $\lambda_s$
       set of perceptual weights, $\alpha_s$
       initial costs $J(v_{is}) = 0$

set iteration count $m = 0$
do
  set $m = m + 1$
  set $V^{(m)} = V^{(m-1)}$
  for level $l = 1$ to number of levels of decomposition do
    for each subband $s$ in level $l$ do
      for each vector $v_{is}$ with node $n_{jl} \in V^{(m)}$, do
        calculate squared vector norm, $P(v_{is}) = ||v_{is}||^2$
        calculate distortion, $D(v_{is}) = ||v_{is} - \hat{v}_{is}||^2$, where
          $\hat{v}_{is} = \arg\min_c \Big\{ \alpha_s ||v_{is} - c||^2 +$
            $\lambda_s \Big[ -\log_2 \Big( p_{cs}^{(m-1)} \Big) \Big] \Big\}, \quad c \in \mathcal{C}_s$        (1)
      calculate updated probabilities $p_{cs}^{(m)} = N_{cs}/N_s$,
        where $N_{cs} =$ num vectors quantized to $c$,
        and $N_s =$ num vectors in subband $s$
      for each vector $v_{is}$ with node $n_{jl} \in V^{(m)}$, do
        estimate rate $R(v_{is}) = -\log_2 \Big( p_{cs}^{(m)} \Big)$,
          where $c$ is the winning codeword from (1)
        calculate cost $G(v_{is}) = \alpha_s D(v_{is}) + \lambda_s R(v_{is})$
        if level $l > 1$,
          calculate cost $J_1 = \sum_x \alpha_s P(x)$, where vector $x$
            is a descendant of $v_{is}$
          calculate cost $J_2 = \sum_x G(x) + J(x)$,
            where vector $x$ is a child (level $l-1$) of $v_{is}$
          if $J_1 \leq J_2$,
            set $n_{jl} = Z$, $J(v_{is}) = J_1$, and remove from
              $V^{(m)}$ all descendant nodes of $v_{is}$
          else,
            set $n_{jl} = S$ and $J(v_{is}) = J_2$
        else,
          set $n_{jl} = S$ and $J(v_{is}) = 0$
while $V^{(m)} \neq V^{(m-1)}$
```

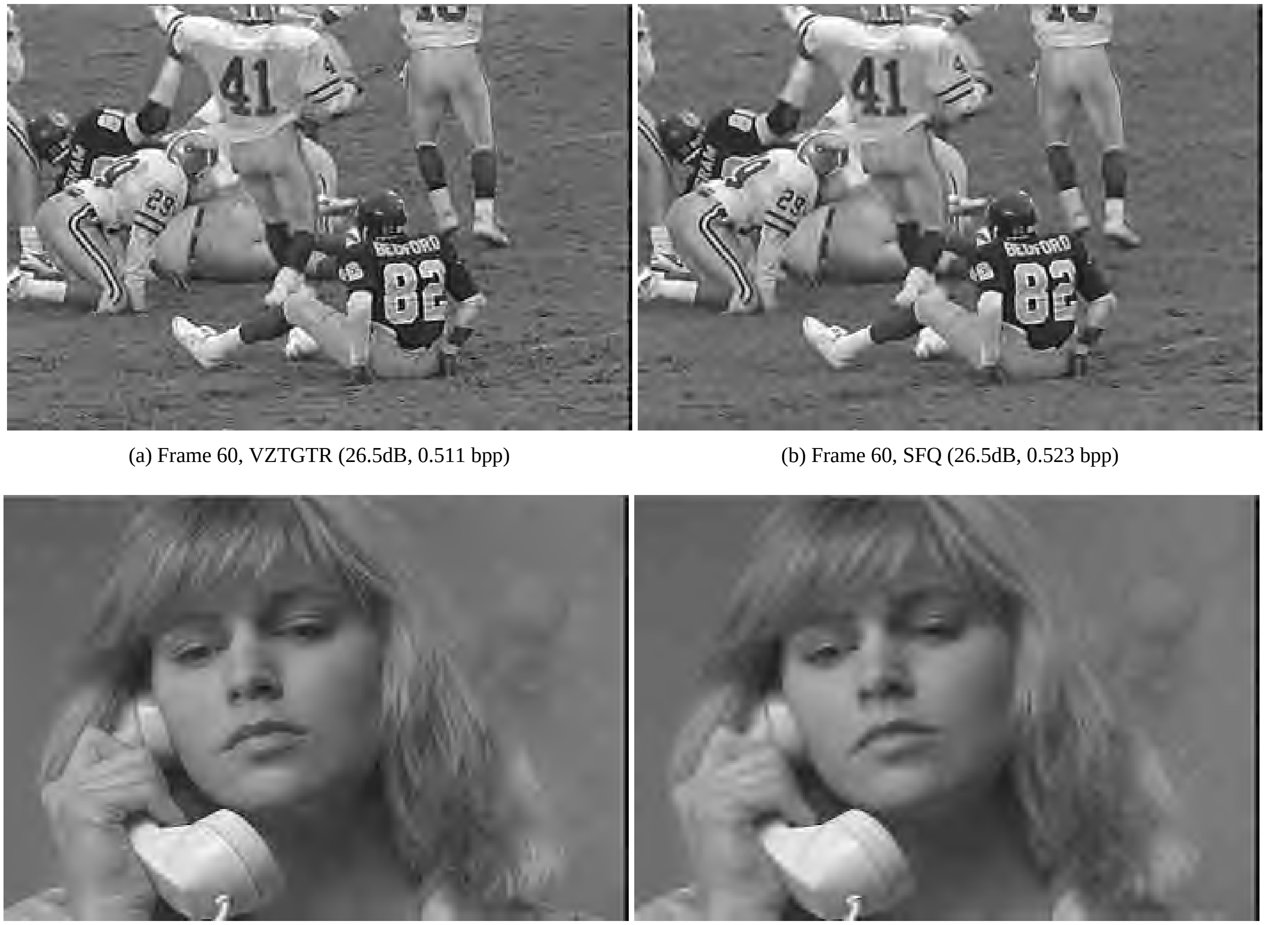
Figure 2: VZT-pruning algorithm.



(a) Distortion

(b) Rate

Figure 3: Performance of the VZTGTR video-coding system vs. that of the SFQ algorithm [2] on the “Football-Susie” sequence consisting of 125 frames from “Football” followed by 75 frames from “Susie.”



(a) Frame 60, VZTGTR (26.5dB, 0.511 bpp)

(b) Frame 60, SFQ (26.5dB, 0.523 bpp)

(c) Frame 160, VZTGTR (32.0dB, 0.141 bpp)

(d) Frame 160, SFQ (31.7dB, 0.214 bpp)

Figure 4: Reconstructed frames from the “Football-Susie” sequence.