within a neighborhood around the reference function G_i. The reduced order warping matrix for the reference function G_i is given by

$$W_i = R_i^{-1}. \tag{12}$$

The warping vector for the expansion coefficient a_i is given by

$$\bar{w}_i = \bar{r}_i W_i \tag{13}$$

where $\bar{r}_i$ is an operator that selects the row associated with a_i from the matrix W_i. The update equation for the coefficient a_i becomes

$$a_i^{l+1} = a_i^l - \beta \bar{w}_i \nabla J_i \tag{14}$$

where ∇J_i is the gradient for the elementary functions included in R_i.

The update term in (14) can be expressed as

$$\bar{w}_i \nabla J_i = -\int\int e(x, y) \left[\sum_j w_i(j) G_j(x, y) \right] dx\, dy \tag{15}$$

where $w_i(j)$ is the jth element of the warping vector $\bar{w}_i$. The new function created by the warping vector is given by

$$P_i(x, y) = \sum_j w_i(j) G_j(x, y). \tag{16}$$

This function P_i is orthonormal to all elementary functions included in R_i, and is referred to as the "locally biorthonormal projection function." The remainder of this section outlines a method for calculating the optimal truncated projection function for a given support width.

Let Q_p denote a pseudo-Hessian matrix whose elements are given by

$$q_p(i, j) = \langle G_i, P_j \rangle_{xy}. \tag{17}$$

If the projection and elementary functions are normalized such that $q_p(i, i) = 1$, then the normalized pseudo-Hessian matrix $\hat{Q}_p$ can be written as

$$\hat{Q}_p = I + V \tag{18}$$

where I is the identity matrix and V is a matrix containing all the off-diagonal elements of $\hat{Q}_p$. If all the elements of V are small, then

$$\hat{Q}_p^{-1} \approx I - V = 2I - \hat{Q}_p. \tag{19}$$

Using (19), an estimate of the locally biorthonormal projection function is recursively defined by

$$P_i^{l+1}(x, y) = 2P_i^l(x, y) - \sum_{j=1}^{N} \hat{q}_p(i, j) P_j^l(x, y). \tag{20}$$

The recursive estimate of P_i will converge if the Gabor expansion is unique (if Q^{-1} exists). The resulting set of projection functions can be used in a descent implementation if G_i in (7) is replaced by P_i. In such an implementation, the lower bound of the optimal convergence factor is estimated by

$$\beta_{LB}^*(i) \approx \frac{1}{\sum_j |q_p(i, j)|}. \tag{21}$$

IV. Comments

The previous section has introduced two methods that improve convergence: the first method uses warping vectors; the second method uses locally biorthonormal projection functions. The overlap characteristics of the members of R_i determine the warping vector w_i and the window of the projection function P_i. In general, a different warping vector or projection function is required for each expansion coefficient a_i. These methods are well suited to a massively parallel implementation which assigns a separate processor/memory group to each elementary function G_i.

For the single processor/memory case, these methods are most efficient for lattices that exhibit overlap invariance: the overlap $q(i, j)$ depends on the distance between elementary functions G_i and G_j, not on their absolute lattice positions. Consider the two lattices presented in [2]: the Cartesian lattice and the log-polar lattice. The overlap characteristics of the Cartesian lattice are completely invariant to shifts in position or frequency. Bastiaans used this invariance to calculate a biorthonormal window [3], [4], which provides a direct solution (no iterations) to the Gabor expansion. If the Cartesian lattice is implemented as an iterative Gabor expansion, the increase in complexity over Daugman's neural network is small because all warping vectors w_i will be identical. The improvement in convergence increases with the size of the neighborhood R_i, approaching Bastiaans' direct solution as the neighborhood size become arbitrarily large.

The log-polar lattice is not invariant to shifts in frequency. Consequently, there is no equivalent to Bastiaan's biorthonormal window. In an iterative implementation, it is possible to improve the convergence by exploiting the position invariant overlap characteristic that exists between elementary functions with a common frequency.[1] If the neighborhood R_i contains only elementary functions with the same frequency, all warping vectors w_i will be identical. As a result, a limited (but noticeable) increase in convergence is obtained with little additional complexity. The rate of convergence can be further increased if the number of unique warping vectors (and the complexity of the implementation) is allowed to increase.

References

[1] R. N. Braithwaite, "The use of the Gabor expansion in computer vision systems," Master thesis, Dep. Elec. Eng., Univ. of British Columbia, Vancouver, Canada, 1989.
[2] J. G. Daugman, "Complete discrete 2-D transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.
[3] M. J. Bastiaans, "Gabor's expansion of a signal into Gaussian elementary signals," *Proc. IEEE*, vol. 68, pp. 538–539, 1980.
[4] M. Porat and Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 452–467, 1988.
[5] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1984.

Image Compression Using the 2-D Wavelet Transform

A. S. Lewis and G. Knowles

***Abstract*—The 2-D orthogonal wavelet transform decomposes images into both spatial and spectrally local coefficients. The transformed coef-**

Manuscript received November 15, 1990; revised August 5, 1991.

The authors were with the Department of Electrical Engineering, Imperial College, London, U.K. They are now with the Department of Information, U.I.B., 07071 Palma de Mallorca, Spain.

IEEE Log Number 9106077.

[1]Common frequency should be interpreted as elementary functions whose modulating waveform has the same radial frequency and the same orientation.

ficients were coded hierarchically and individually quantized in accordance with the local estimated noise sensitivity of the human visual system (HVS). The algorithm can be mapped easily onto VLSI. For the Miss America and Lena monochrome images, the technique gave high to acceptable quality reconstruction at compression ratios of 0.3–0.2 and 0.64–0.43 bits per pixel (bpp), respectively.

I. Introduction

Traditional image compression techniques have been designed to exploit the statistical redundancy present within real world images. The discrete cosine transform (DCT), DPCM, and the entropy coding of subband images are all examples of this statistical approach. Removing redundancy can only give a limited amount of compresssion; to achieve high ratios, some of the nonredundant information must be removed. The statistical coders produce annoying visual degradation when operating in this mode because they introduce errors in visually important parts of the image structure (such as feature edges). By using methods of image decomposition that closely mimic the human visual system (HVS), compression can take into account the importance of each individual coefficient and code accordingly.

The HVS is an information processing system, receiving spatially sampled images from the cones and rods in the eye and deducing the nature of the objects it observes by processing this image data. At a low level, objects can be thought of as structures made up of surfaces of the same color or texture bounded by edges. The color and texture of an object is greatly affected by its orientation and illumination so the edges are usually the most important means of recognition, so a good image compression algorithm should try to minimize edge distortion.

Psychophysicists and visual psychologists have performed many experiments on the HVS to determine how it processes image data. They discovered that the eye filters the image into a number of bands, each approximately one octave wide in frequency. Further, in the spatial domain, the image should be considered to be composed of information at a number of different scales [6]. Marr formulated a *constraint of spatial localization* that *the physical phenomena that give rise to intensity changes in the image are spatially localized.*

In this paper we introduce a new approach to image compression based on decomposing the image using the orthogonal wavelet transform, and then apply a compression algorithm based on Marr's constraint of spatial locality. Firstly, we will introduce the wavelet transform, discuss its implementation by quadrature mirror filtering (QMF), and describe the ideas behind our compression method. In Section III we present an algorithm implementing our method which is suitable for real time use either on a DSP chip, or on dedicated hardware. In Section IV we give some results on the test images of Miss America and Lena for a range of compression ratios.

II. Wavelet Transform and Image Compression

One-dimensional wavelet theory defines a function ψ, the wavelet, and its associated scaling function ϕ, such that the family of functions $\{\psi^j(x)\}_{j\in Z}$, where $\psi^j(x) = \sqrt{2^j}\psi(2^j x)$, are orthonormal. The wavelet transform can be implemented by quadrature mirror filters [11], [12] $G = (g(n))$, and $H = (h(n))$, $n \in Z$, where $h(n) = 1/2\langle\phi(x/2), \phi(x - n)\rangle$, and $g(n) = (-1)^n h(1 - n)$ ($\langle\ \rangle$ denotes L^2 inner product). H corresponds to a low-pass filter, and G is an octave wide high-pass filter. The reconstruction filters have impulse response $h^*(n) = h(1 - n)$, and $g^*(n) = g(1 - n)$. For a more detailed analysis of the relationship between wavelets and QMF see Mallat [7].

For images we use the hierarchical wavelet decomposition suggested by Mallat [7]. The G and H filters are applied to the image in both the horizontal and vertical directions, and the filter outputs subsampled by a factor of two, generating three orientation selective high-pass subbands, GG, GH, HG, and a low-pass subband HH. The process is then repeated on the HH band to generate the next level of the decomposition, etc. Four octaves of decomposition leads to thirteen subbands (Fig. 1). Fig. 2 shows the image of Lena decomposed in this way. This hierarchical approach to image decomposition fulfills the role of scales in Marr's decomposition. QMF subband coding algorithms for images have been explored by Vetterli [8], and for use in compression by Woods and O'Neil [9] using a flat decomposition and by Gharavi and Tabatabai [10] in a pyramidal decomposition akin to the wavelet transform.

In general the wavelet transform requires much less hardware to implement than Fourier methods, such as the discrete cosine transform DCT [5]. However, for the 4 coefficient Daubechies [1] wavelet, it is especially simple to calculate and invert the transform in hardware as no multipliers are needed to calculate the quantized coefficients ($h(0) = 11/32$, $h(1) = 19/32$, $h(2) = 5/32$, $h(3) = -3/32$) [4]. This allows us to incorporate the advantages of a multiresolution approach to decompose and reconstruct the image, without high hardware costs.

The wavelet decomposition is an alternative representation of image data but the number of bits used to store it has not changed. To compress the image data, we must decide which coefficients to send and how many bits to use to code them. Our compression algorithm consists of taking the low-pass subband in full, and then deciding which coefficients within the remaining subbands to keep. The problem is to decide which of the nonzero wavelet transform coefficients correspond to noise and which visually important details of the image, in particular, we want to preserve the edge-like information within the image. Simply thresholding each of the coefficients, would leave extra noise and visually unimportant information in the compressed image. We apply an algorithm based on Marr's constraint of spatial locality when deciding which coefficients to keep. So that if an important detail occurs at some place in the image we expect that the coefficients corresponding to that location will exceed a threshold in **more than one octave**; the orientation of the detail will be determined by which of the GH, HG, or GG subbands we use. In this way we utilize both the frequency and spatial locality of the wavelet transform to detect and code the image data efficiently. By examining the image at low resolutions, and hence, a low number of samples, an initial guess at the location of the edges can be made. These guesses can be confirmed or rejected by examining the higher resolution detail signal at the same spatial location. We shall use the same technique to not only save on processing time but also to compress the picture through rejecting the redundant areas; by controlling which areas are rejected we can aim for high compression and high quality in the final output. The threshold values are determined adaptively from a simple HVS model. Finally, the smaller the support of the wavelet, the less nonzero wavelet coefficients will correspond to an edge, so the more efficient will be our compression scheme. For this reason we chose the Daubechies-4 wavelet. The second stage of the compression process is to quantize the remaining nonzero coefficients; we use a linear mid-step quantizer with step-size derived from a HVS model, and then Huffman code the resulting values.

III. The Compression Algorithm

In this section we describe the algorithm implementing the ideas discussed in the previous section, and combine this with a quantization model based on the HVS.

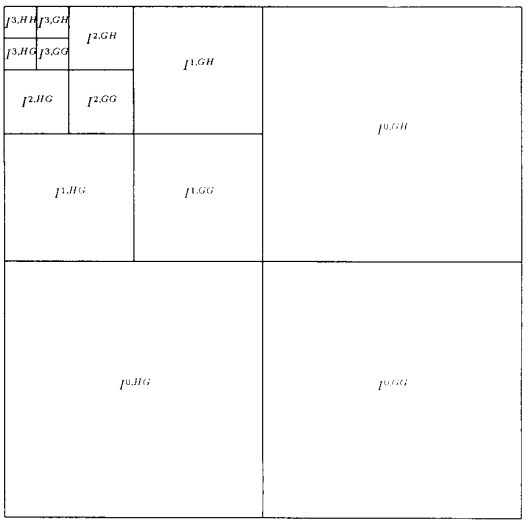

Fig. 1. Wavelet multiresolution image decomposition.

Fig. 2. Wavelet decomposition of Lena image (high-pass coefficients ×4).

Each coefficient in the high-pass bands of the wavelet transform for octaves ($r = 1 \cdots 3$) has four coefficients corresponding to its spatial position in the octave band above it in frequency. This suggests using a four branch tree to represent the structure of the decomposition. This data structure was used successfully in a previous video compression algorithm [2]. The tree is constructed by first considering each coefficient of all orientations ($s = GG, GH, HG$) of the lowest octave in the high-pass bands. The value is coded

and if its absolute value is greater than a fixed threshold value, the process is repeated for the four corresponding coefficients in the octave above. If not, the tree is rejected and zero coefficients are implied by the coding system for the tree above. This technique requires no control signals or addresses to be added by the coder and is efficiently implemented using stack-based hardware.

Our thresholding algorithm was far from optimal; though it followed edges and rejected flat areas very efficiently, mistakes happened too often and blurred parts of the image around edges.

A better measure of the likelihood of an edge at a point within a subband would be to calculate the energy in a small area encompassing the point. By using a small 2 by 2 block as the nodes of the tree structure, rather than single coefficient values, thresholding errors are greatly reduced. In addition, special block codes are introduced to increase the coding efficiency above that of the original method. Fig. 3 shows the new small block-based tree structure and its coding with the aid of the following recursive function, SendTree. Firstly, the low-pass image *HH* is coded in full(8 bits per pixel (bpp)), then the remaining subbands are coded recursively by applying the following subroutine SendTree to all the blocks within the lowest frequency subbands. (In the following, let $I^{r,s}(x, y)$ be the coefficient value of the decomposition I at octave r, orientation s, and position x, y within that subband.)

```
SendTree (I, r, s, x, y)
  If (THRESHOLD < ThresholdFunction(I^{r,s}(x + {0, 1}, y +
  {0, 1}))) {
    SendToken(BlockNotEmpty);
    SendCoefficients(I^{r,s}(x + {0, 1}, y + {0, 1}));
    SendTree(I, r − 1, s, 2x, 2y);
    SendTree(I, r − 1, s, 2(x + 1), 2y);
    SendTree(I, r − 1, s,2x, 2(y + 1));
    SendTree(I, r − 1, s, 2(x + 1), 2(y + 1));
  } else SendToken(BlockEmpty);
```

The ThresholdFunction was chosen to be the sum of squares of the block's coefficients normalized to the HVS perceptual threshold discussed later.

Tree structuring and block coding have exploited interband and intraband correlation to facilitate data compression but so far we have not considered how to code the coefficient data itself (SendCoefficients). With 8 bpp we have 256 different levels to represent our transformed coefficients, but this may not be the optimum value. We can use our knowledge of the HVS to determine the smallest number of levels required at each point within the transformed image. The errors caused by using fewer levels can be considered as a noise source; visual psychophysics states that a number of factors effect the noise sensitivity of the eye: the background luminance, the proximity to an edge, the frequency band, and texture masking.

Background luminance is related to noise sensitivity by Weber's law. The eye is less sensitive to noise for brighter background luminances. The low-pass coefficients can be used to provide values of background luminance.

Edge proximity or spatial masking relates noise sensitivity to distance from and height of the edge. As the coefficients we send are supposed to be part of an edge, the spatial locality of the octave provides distance information, while the energy of the lower frequency coefficients indicates edge height. Sensitivity decreases for increasing edge height and at decreasing distances from the edge.

Band sensitivity is of the octave currently being coded. This is a fixed value for each octave, orientation, and luminance-chrominance channel. We have formulated an empirical model for this based on HVS experiments.

Octave r

$I^{r,s}(x, y)$	$I^{r,s}(x+1, y)$
$I^{r,s}(x, y+1)$	$I^{r,s}(x+1, y+1)$

Octave r-1

$I^{r-1,s}(2x, 2y)$	$I^{r-1,s}(2x+1, 2y)$	$I^{r-1,s}(2x+2, 2y)$	$I^{r-1,s}(2x+3, 2y)$
$I^{r-1,s}(2x, 2y+1)$	$I^{r-1,s}(2x+1, 2y+1)$	$I^{r-1,s}(2x+2, 2y+1)$	$I^{r-1,s}(2x+3, 2y+1)$
$I^{r-1,s}(2x, 2y+2)$	$I^{r-1,s}(2x+1, 2y+2)$	$I^{r-1,s}(2x+2, 2y+2)$	$I^{r-1,s}(2x+3, 2y+2)$
$I^{r-1,s}(2x, 2y+3)$	$I^{r-1,s}(2x+1, 2y+3)$	$I^{r-1,s}(2x+2, 2y+3)$	$I^{r-1,s}(2x+3, 2y+3)$

Fig. 3. Block tree structure.

Texture masking decreases the sensitivity to noise if there is high activity in the locality of the coefficient. The energy of lower frequency coefficients can indicate the texture activity level.

A mathematical model of the HVS can be constructed to allow the estimation of noise sensitivity for any part of the transformed image. Subband coding algorithms that exploit HVS properties usually quantize the whole subband on the basis of spectral response alone. Safranek and Johnston [13] combined band sensitivity, background luminance, and texture masking information to provide a perceptual threshold for each subband coefficient. They then used the minimum threshold value to quantize the entire band. In our algorithm we use a similar calculation to estimate the perceptual threshold of each 2 by 2 block, and then quantize each pixel in the block with this threshold using a linear mid-step quantizer. We also use this perceptual threshold figure to normalize the coefficient value before edge detection. It should be noted that only previously coded values are needed in the quantizer, hence, no additional side information is required.

The quantizer step-size for the coefficients in Fig. 3, *qstep*, is calculated as follows:

$$\text{qstep}(r, s, x, y) = q_0 * \text{frequency}(r, s) * \text{luminance}(r, x, y) * \text{texture}(r, x, y)^{0.034}$$

where q_0 is a normalization constant and

$$\text{frequency}(r, s) = \begin{cases} \sqrt{2}, & \text{if } s = GG \\ 1, & \text{otherwise} \end{cases} * \begin{cases} 1.00, & \text{if } r = 0 \\ 0.32, & \text{if } r = 1 \\ 0.16, & \text{if } r = 2 \\ 0.10, & \text{if } r = 3 \end{cases}$$

$$\text{luminance}(r, x, y) = 3 + \frac{1}{256} \sum_{i=0}^{1} \sum_{j=0}^{1} I^{3,HH}(i + 1 + x/2^{3-r}, j + 1 + y/2^{3-r})$$

$$\text{texture}(r, x, y) = \sum_{k=1}^{3-r} 16^{-k} \sum_{s}^{GG,GH,HG} \sum_{i=0}^{1} \sum_{j=0}^{1} (I^{k+r,s}(i + x/2^k, j + y/2^k))^2 + 16^{3-r} \operatorname{var}(I^{3,HH}(\{1, 2\} + x/2^{3-r}, \{1, 2\} + y/2^{3-r}))$$

where var is the variance of the four coefficient block, and the summation is zero when its lower limit exceeds its upper limit.

TABLE I
VARIABLE LENGTH CODES

Quantized Level	Code Bits
.	.
.	.
.	.
−5	111001
−4	11101
−3	1111
−2	101
−1	011
0	00
1	010
2	100
3	1101
4	11001
5	110001
.	.
.	.
.	.

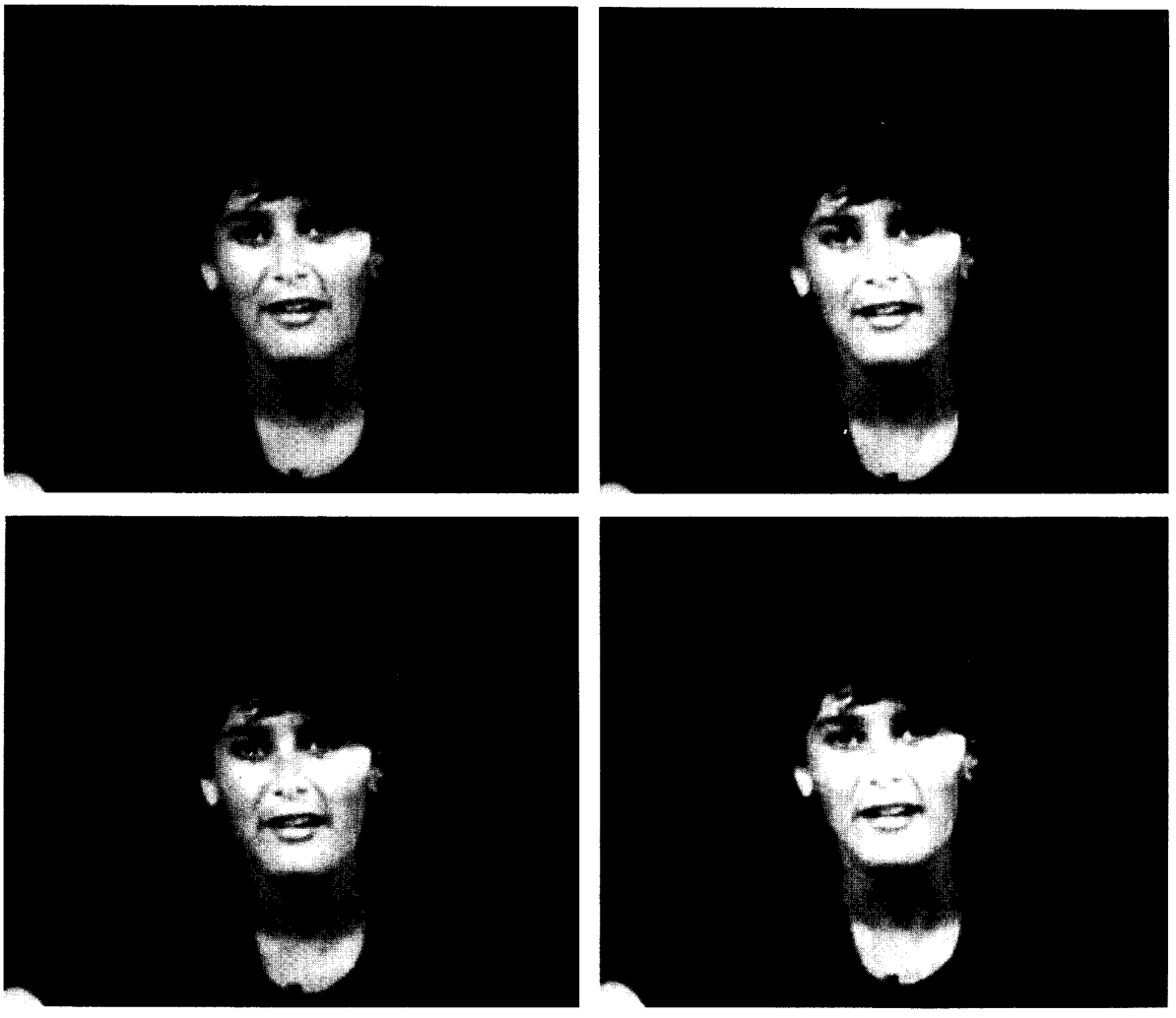

Fig. 4. Coding results for frame #1 of the Miss America test sequence. The original image is in the upper left. The upper right is coded at 0.30 bpp (SNR 37.09), the lower left at 0.25 bpp (SNR 36.56), and the lower right at 0.20 bpp (SNR 36.11).

Fig. 5. Coding results for Lena. The original image is in the upper left. The upper right is coded at 0.64 bpp (SNR 34.76), the lower left at 0.53 bpp (SNR 34.03), and the lower right at 0.43 bpp (SNR 33.18).

Our equations represent a crude approximation to the HVS noise sensitivity; more accurate models could be made to increase compression or picture quality.

After the number of quantization levels has been established and the coefficient has been quantized, further compression can be achieved by using the statistical properties of quantized coefficients. A well-known statistical property of subband coefficients is their Laplacian probability distribution function with a mode value of zero. Even after quantization this property is approximately true. A simple variable length or Huffman code can be constructed on the basis of this assumed property. Further compression can be achieved from the use of the variable length code as it exploits the entropy within the coefficient values. Table I shows some example code strings for a range of quantization levels.

The first two bits indicate the magnitude of the coefficient (00 − 0, 01 − 1, 10 − 2, and 11 − remaining). The third bit indicates the sign (0 − positive and 1 − negative). Trailing zeros terminated by a one indicate the magnitudes of the remaining values above two.

Our compression algorithm uses four "orthogonal" techniques to exploit redundancy within image data. Additional techniques could be used, such as DPCM, on the low-pass band or the entropy of a block's nonzero symbols, but we currently do not consider them to be cost effective.

IV. Results

We have applied our method to two standard monochrome test images, Miss America and Lena; the coded reconstructions are shown in Figs. 4 and 5, respectively. The Miss America image is the first frame of a test sequence and is 352 by 288 pixels (CIF format) in size. The Lena image is 512 by 512 pixels in size. The SNR values are calculated over the whole of the image (including the edges).

Our results achieve better compression ratios for a given picture quality than the standard subband coding methods. Additionally, our algorithm is significantly simpler to implement in hardware to achieve real-time performance. This is important for the coding of video sequences [3]. We find that the errors introduced by our methods are less visually annoying than for DCT compressed images due to the lack of blocking effects. At very high compression

ratios visual degradation is introduced, mainly as blotchiness in flat areas and slight fuzziness around sharp discontinuities. In particular, the background of the Miss America and Lena images, and Lena's face and shoulder become blotchy. Also their hair and Lena's feather lose their sharpness. Finally, our method is equally applicable to the compression of color images in YUV format.

In conclusion, our results indicate that the combination of HVS compatible filters with finite support, and a quantizer which introduces noise in the visually least important and noise insensitive parts of the image gives a significant improvement in compression/image quality over block-based transform methods. Finally, the 4-tap Daubechies filter and the coder we use are much simpler to implement in hardware than DCT, VQ or other subband coding methods [4]. We are currently incorporating this method in a video codec, and implementing it in VLSI([3]).

References

[1] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *IEEE Trans. Inform. Theory*, vol. 36, pp. 961-1005, Sept. 1990.

[2] A. S. Lewis and G. Knowles, "Video compression using 3D wavelet transforms," *Electron. Lett.*, vol. 26, no. 6, pp. 396-397, 1990.

[3] ——, "A 64 Kb/s video codec using the 2-D wavelet transform," presented at IEEE Data Compression Conf., Snowbird, UT, 1991.

[4] ——, "A VLSI architecture for the 2D Daubechies wavelet transform without multipliers," *Electron. Lett.*, vol. 27, no. 2, pp. 171-173, 1991.

[5] G. Knowles, "VLSI architecture for the discrete wavelet transform," *Electron. Lett.*, vol. 26, no. 15, pp. 1184-1185, 1990.

[6] D. Marr, *Vision*. New York: Freeman, 1982.

[7] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 674-693, 1989.

[8] M. Vetterli, "Multi-dimensional sub-band coding: Some theory and algorithms," *Signal Proc.*, vol. 6, pp. 97-112, 1984.

[9] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Aconst. Speech, Signal Processing*, vol. 34, pp. 1278-1288, Oct. 1986.

[10] H. Gharavi and A. Tabatabai, "Sub-band coding of monochrome and color images," *IEEE Trans. Circuit Syst.*, vol. 35, pp. 207-214, Feb. 1988.

[11] A. Croisier, D. Estaban, and C. Galand, "Perfect channel splitting by use of interpolation, decimation, tree decomposition techniques," presented at Int. Conf. on Information Sciences/Systems, Patras, Greece 1976.

[12] M. J. T. Smith and T. P. Barnwell III, "Exact reconstruction techniques for tree structured sub-band coders," *IEEE Trans. Acoust. Speech, Signal Processing*,, pp. 434-441, June 1986.

[13] R. J. Safranek and J. D. Johnston, "A perceptually tuned sub-band image coder with image dependent quantization and post-quantization data compression," in *Proc. IEEE ASSP'89*, vol. 3, 1989, pp. 1945-1948.

An Edge Preserving Differential Image Coding Scheme

Martin C. Rost and Khalid Sayood

***Abstract*—Differential encoding techniques are fast and easy to implement. However, a major problem with the use of differential encoding for images is the rapid edge degradation encountered when using such systems. This makes differential encoding techniques of limited utility especially when coding medical or scientific images, where edge preservation is of utmost importance. We present a simple, easy to implement differential image coding system with excellent edge preservation properties. The coding system can be used over variable rate channels which makes it especially attractive for use in the packet network environment.**

Manuscript received January 18, 1990; revised April 28, 1991. This work was supported by the NASA Goddard Space Flight Center under Grant NAG-5-916.

M. C. Rost is with Sandia National Laboratories, Albuquerque, NM 87185.

K. Sayood is with the Department of Electrical Engineering and the Center for Communication and Information Science, University of Nebraska, Lincoln, NE 68588-0511.

IEEE Log Number 9106076.

I. Introduction

The transmission and storage of digital images requires an enormous expenditure of resources, necessitating the use of compression techniques. These techniques include relatively low complexity predictive techniques such as adaptive differential pulse code modulation (ADPCM) and its variations, as well as relatively higher complexity techniques such as transform coding and vector quantization [1], [2]. Most compression schemes were originally developed for speech and their application to images is at times problematic. This is especially true of the low complexity predictive techniques. A good example of this is the highly popular ADPCM scheme. Originally designed for speech [3], it has been used with other sources with varying degrees of success. A major problem with its use in image coding is the rapid degradation in quality whenever an edge is encountered. Edges are perceptually very important, and therefore, their degradation can be perceptually very annoying. If the images under consideration contain medical or scientific data, the problem becomes even more important, as edges provide position information which may be crucial to the viewer. This poor edge reconstruction quality has been a major factor in preventing ADPCM from becoming as popular for image coding as it is for speech coding. While good edge reconstruction capability is an important requirement for image coding schemes, another requirement that is gaining in importance with the proliferation of packet switched networks is the ability to encode the image at different rates. In a packet switched network, the available channel capacity is not a fixed quantity, but rather fluctuates as a function of the load on the network. The compression scheme must, therefore, be capable of taking advantage of increased capacity when it becomes available while providing graceful degradation when the rate decreases to match decreased available capacity.

In this paper we describe a DPCM-based coding scheme which has the desired properties listed above. It is a low complexity scheme with excellent edge preservation in the reconstructed image. It takes full advantage of the available channel capacity providing lossless compression when sufficient capacity is available, and very graceful degradation when a reduction in rate is required.

II. Notation and Problem Formulation

The DPCM system consists of two main blocks, the quantizer and the predictor (see Fig. 1). The predictor uses the correlation between samples of the waveform $s(k)$ to predict the next sample value. This predicted value is removed from the waveform at the transmitter and reintroduced at the receiver. The prediction error is quantized to one of a finite number of values which is coded and transmitted to the receiver and is denoted by $e_q(k)$. The difference between the prediction error and the quantized prediction error is called the quantization error or the quantization noise. If the channel is error free, the reconstruction error at the receiver is simply the quantization error. To see this, note (Fig. 1) that the prediction error $e(k)$ is given by

$$e(k) = s(k) - p(k) \tag{1}$$