SUBBAND CODING FOR IMAGE DATA ARCHIVING

Daniel Glover
NASA Lewis Research Center
Cleveland, Ohio 44135

S. C. Kwatra
The University of Toledo
Toledo, Ohio 43606

Abstract. The use of subband coding on image data is discussed. An overview of subband coding is given. Advantages of subbanding for browsing and progressive resolution are presented. Implementations for lossless and lossy coding are discussed. Algorithm considerations and simple implementations of subband systems are given.

1. Introduction

Subband coding is a promising strategy for data compression that has been used successfully in recent years. This paper discusses subband coding with the perspective of organizing data for storage and retrieval rather than as just a data compression strategy. An overview of subband coding is given followed by discussions of subbanding for archiving. Finally, some simple algorithms for subbanding image data are shown.

Subband coding of speech was developed in the late 1970's [1] and matured in the 1980's. The key development for subbanding signals was the Quadrature Mirror Filter (QMF) in 1976 [2]. A QMF bank cancels out the aliasing that is generated by filtering providing a good quality reconstruction of the signal. Subband coding was applied to image data in the mid-1980's [3,4,5]. The theory for perfect reconstruction filter banks (which not only cancel aliasing, but do not introduce amplitude or phase distortion) was also developed around this time [6]. An overview of concepts important to subband coding can be found in [7,8,9].

2. Subband Coding

A generic subband coding system is shown in Figure 1. The original signal is split up into N frequency bands (subbands) by the analysis filter bank represented by the matrix $H(z)$. Each subband is then decimated, keeping only every Nth value. If the proper conditions are met, the resulting total number of values in the subbands is the same as in the original signal. No compression has been achieved yet; the subbanding process prepares the data for compression coding. The coders shown in each of the subbands could be the same or (more commonly) tailored to match the characteristics of the data in a particular type of subband.

After the data has been coded, it is transmitted through the channel and received or, equivalently, stored and recalled. The data is decoded and the subbands sent through a synthesis filter bank, $F(z)$, where the original signal is reconstructed. The synthesis filter bank has a special relationship with the analysis bank such that the aliasing caused by the analysis filters is canceled. The filter banks can be designed such that the reconstructed signal is a perfect replica of the original signal, provided that no errors have been introduced in the coding or transmission stages.

To take advantage of the two-dimensional structure of image data, the filters used in the subband system should be two dimensional. Designing such filters is difficult, so separable filters have typically been used to accomplish 2-D processing. Separable filters use 1-D filters and apply them to the data in two directions, i.e., first horizontally and then vertically. This results in straightforward filter design, but requires storage to access horizontally processed data in a vertical direction (see Figure 2).

An important property of image data is the importance of phase as a vehicle for information. A good example is obtained from reconstructions of images using only the Fourier transform phase compared with reconstructions using only the magnitude; the phase-only reconstructions are recognizable whereas the magnitude-only reconstructions are not [10]. Linear phase filters are most commonly used for image data. Since most of the information is in the phase, frequency resolving capability is not as critical an issue with image data as it is with, say, audio data. This means that short kernel filters can be used on image data with good results.

The advantages of using short kernel filters include simple hardware design and speed of algorithm execution. Short kernel implementations are attractive for on-line compression and for fast retrieval of stored data.

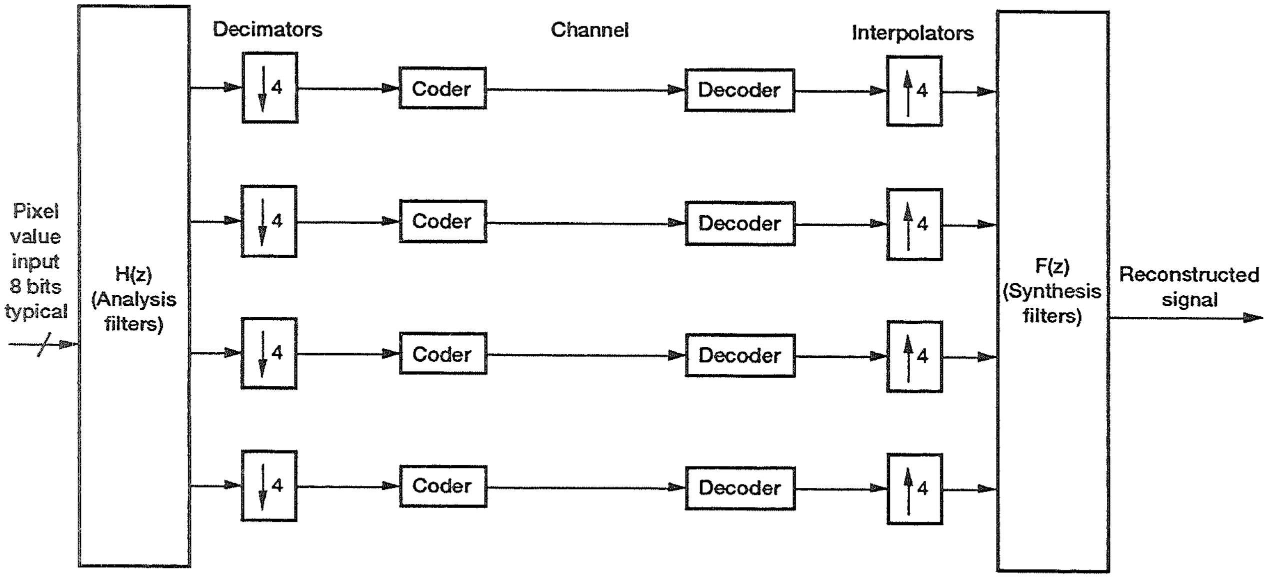


Figure 1. A generic four-channel subband coder.

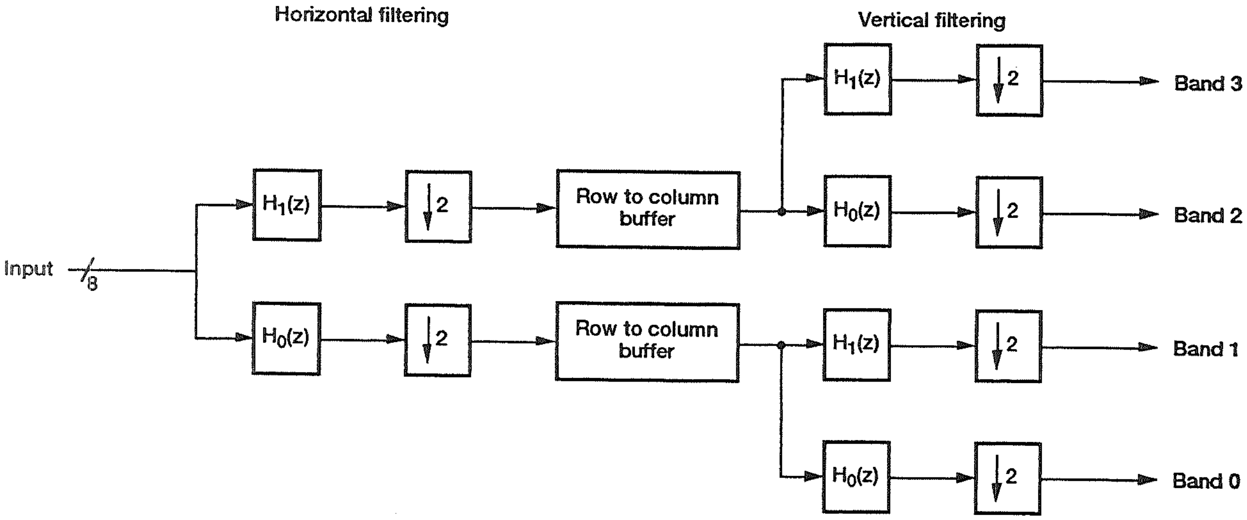


Figure 2. A separable, four-channel subband analysis bank.

3. Browsing and Progressive Resolution

A property of the subband signals that makes the splitting of the original signal into lowpass, bandpass, and highpass bands useful for lossy compression is that most of the original signal energy winds up in the low frequency band. The higher frequency subbands can be compressed quite a bit without compromising the picture quality if the lowest subband is not degraded.

The fact that the low frequency band is a lower resolution version of the original image makes subbanding attractive as a browsing tool. The low frequency band is generated as a consequence of subbanding and can be tracked separately from the higher frequency subbands in a browsing catalog. The low band can be retrieved and viewed separately. If a higher resolution version is desired, the higher frequency bands can be retrieved and image reconstructed.

Color images are handled by subbanding each component separately. These components could be RGB, YUV, etc. For browsing, the most significant component can be used (e.g., luminance for color images) or all the components can be retrieved and viewed separately or recombined into a low resolution color image. Care must be taken when subbanding chrominance data or false colors can be generated in the low resolution (low frequency) band.

Color image data can be stored as color components (RGB) or as luminance and chrominance data (e.g., YUV). The components are usually subbanded and coded separately. A subband filter that averages pixels for the low frequency band can generate false colors around transitions. To avoid this it is possible to use a low frequency subband that is just a subsampled version of the original (see Section 5).

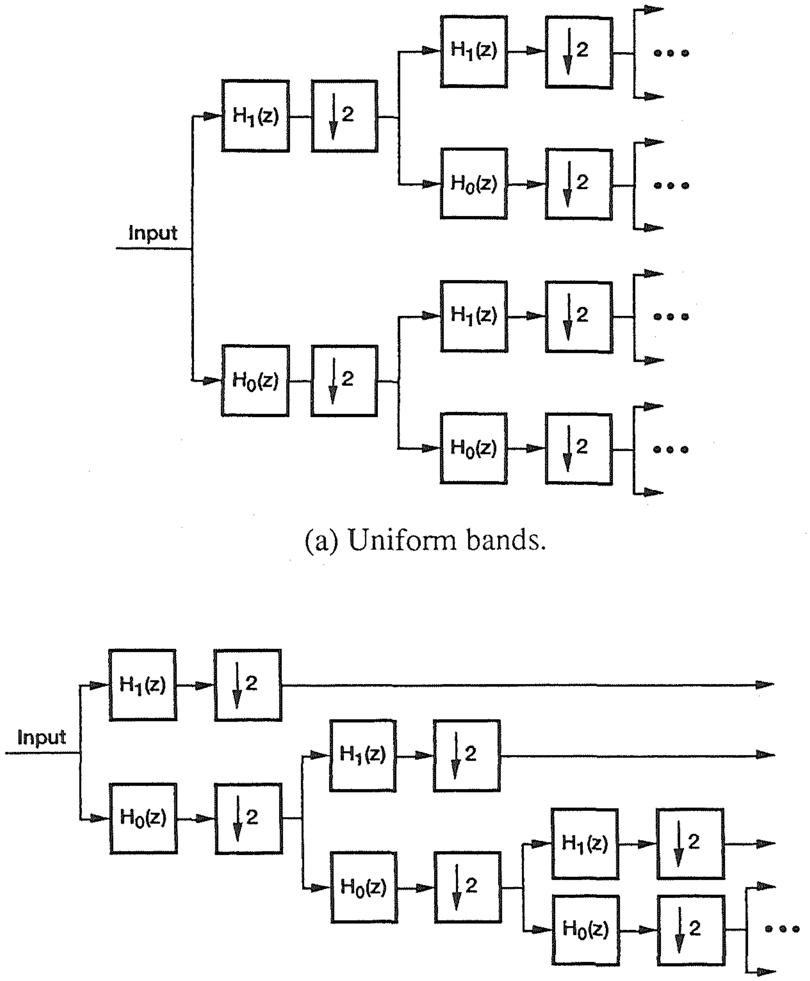
A typical approach to generating multiple subbands is to cascade sets of two-channel filters in a tree structure. Although it is possible to design multiple-band filter banks, it is usually easier to just cascade a simple two-band filter bank to get four, seven, sixteen, or whatever number of bands is appropriate. A corresponding tree structure is used for the synthesis bank and this lends itself to progressive resolution reconstruction.

The two most common tree structures are the octave-band split and the uniform-band split (see Figure 3). Either structure is equivalent from a browsing standpoint since the low frequency bands can be made as small as is practical for viewing using either tree. For progressive resolution, the octave-band split is most efficient in terms of reconstructing a series of increasing resolution images.

4. Lossless and Lossy Coding

Subbanding is a strategy for coding and as such can be used in either a lossless or lossy manner. The subbanding process itself can be made lossless by using perfect reconstruction filter banks. Of course, errors in the coding or transmission of the data will result in losses. The subbanding analysis and synthesis processes do not in themselves result in any data loss if properly designed.

There are two places where losses can be intentionally introduced: 1) in the coders, and 2) in the way subband values are treated in terms of maintaining accuracy. Lossy coders deliberately throw out "data" that is irrelevant or is predominantly noise. The signal reconstructed from lossy coders does not (in most cases) exactly match the original signal. This "loss" is by design and in some cases can result in a reconstructed signal that is better than the original. The term "lossy" is unfortunate in that it conjures up thoughts of lost data in scientists' minds when most of what is usually lost is noise.



(a) Uniform bands.

(b) Octave bands.

Figure 3. Subband analysis tree structures.

The second area where losses can be introduced.is in the accuracy of subband values. A typical original signal sample size might be 8 bits. When that signal is subbanded, it may take larger sample sizes in the subbands to maintain the accuracy needed for perfect reconstruction. For example, a simple two-channel split will require a minimum of 1/2 bit/sample larger average sample size to maintain accuracy. For a more conventional filter, the sample size requirements are much greater. Although the number of samples doesn't increase over the original signal due to subbanding, the total amount of data increases due to the required accuracy if perfect reconstruction is desired.

An advantage of simple subbanding systems, such as the Walsh-Hadamard transform/subband system (WHT), is that the sample size does not have to increase very much to maintain perfect accuracy. For the two-dimensional, 2x2 WHT, the sample size only increases two bits to maintain perfect accuracy. For more complex subband filters double precision accuracy might be needed to avoid quantization errors (due to truncation or rounding). Section 5 presents a subband system which is even more efficient, in terms of sample size for lossless coding, than the WHT.

For lossy coding the increase in subband value accuracy is not needed because these extra bits would just be dropped in the coding stage. A very effective coding scheme for image data uses simple coarse quantization of the high frequency bands for compression [11].

5. Some Subbanding Algorithms

It can be shown that simple, two-dimensional subband filters can be derived from block transforms [12]. The filters are the basis pictures (formed by outer product of the basis vectors) of the corresponding transform matrix. The transform matrix (in the orthogonal case) is made up of vectors formed from the coefficients of the analysis filters. For the Walsh-Hadamard transform (WHT), with the transform matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

the two-dimensional filters are:

$$\boldsymbol{B}_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix};\; \boldsymbol{B}_1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix};\; \boldsymbol{B}_2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix};\; \boldsymbol{B}_3 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}; \qquad (1)$$

To subband an image with these filters, the image component data is broken up into 2x2 blocks and each block is multiplied (inner product) by each of the filters. For example, let $\boldsymbol{D}_1$ be a 2x2 block (matrix) of luminance pixel values. The low band value, S_0, corresponding to that block is:

$$S_0 = \boldsymbol{B}_0 \bullet \boldsymbol{D}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} d_0 & d_1 \\ d_2 & d_3 \end{bmatrix} = d_0 + d_1 + d_2 + d_3 \qquad (2)$$

The other subband values are:

$$S_1 = d_0 + d_1 - d_2 - d_3$$

$$S_2 = d_0 - d_1 + d_2 - d_3$$

$$S_3 = d_0 - d_1 - d_2 + d_3$$

After these subband values have been calculated, a new 2x2 block of data is read in. If the subband values are kept in a block, then a traditional block transform is obtained. If the subband values are grouped separately (e.g., all the S_0 values in one group), then a subband representation is obtained. The subband values should be scaled (by dividing by 4) and shifted to a positive range if it is desired to view them. If all of the low band (S_0) values are viewed as an image that is 1/2 by 1/2 the original size, then a low resolution, low pass version of the original image is obtained. The higher bands contain edge information.

Since the inverse transform matrix is the same as the forward WHT matrix, the filters for reconstruction are also the same. To get the reconstructed data block, Δ_1, the following equations are used (including the overall transform scaling factor of 1/4 here):

$$\delta_0 = 1/4(S_0 + S_1 + S_2 + S_3)$$

$$\delta_1 = 1/4(S_0 + S_1 - S_2 - S_3)$$

$$\delta_2 = 1/4(S_0 - S_1 + S_2 - S_3)$$

$$\delta_3 = 1/4(S_0 - S_1 - S_2 + S_3)$$

Cascading the algorithm is equivalent to performing a high order subband/transform. For example, performing a four channel subbanding on data that has already been split into four channels results in a sixteen channel split that is equivalent to a 4x4 WHT.

Another subband system from [12] uses the following non-symmetric transform:

$$\boldsymbol{G} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}; \quad \boldsymbol{g} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3}$$

The analysis filters are given by the basis pictures of ***G***:

$$\boldsymbol{B}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}; \; \boldsymbol{B}_1 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}; \; \boldsymbol{B}_2 = \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix}; \; \boldsymbol{B}_3 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}; \tag{4}$$

This system has been characteristic that the low band is a subsampled version of the original. This is useful for chrominance data where an averaging function could create false colors in the low band. This algorithm is also simpler, requiring fewer additions. It is not a symmetric transform, however, and the following filters are used for reconstruction:

$$\boldsymbol{b}_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; \; \boldsymbol{b}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}; \; \boldsymbol{b}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}; \; \boldsymbol{b}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; \tag{5}$$

The equations for analysis and synthesis corresponding to (4) and (5) are:

$$\begin{array}{ll} \underline{\text{Analysis}} & \underline{\text{Synthesis}} \\ S_0 = d_3 & \delta_0 = S_0 + S_1 + S_2 + S_3 \\ S_1 = d_1 - d_3 & \delta_1 = S_0 + S_1 \\ S_2 = d_2 - d_3 & \delta_2 = S_0 + S_2 \\ S_3 = d_0 - d_1 - d_2 + d_3 & \delta_3 = S_0 \end{array}$$

6. Simulation Results

Some examples of subbanding an image are shown in Figures 4 through 10. Figure 4 shows the original image "Lenna" which was taken from the USC unofficial standard image set. The image is 512 x 512 pixels (but appears stretched horizontally due to the display system used) and was converted to monochrome from a color original.

Figure 5 shows a four band split of the original using G from Equation (3). The low band in the upper left corner is a subsampled version of the original while the high band in the lower right corner has very little energy. Figure 6 is a close-up of the low band which makes the low resolution artifacts, such as staircase edges, more visible.

The sixteen band split shown in Figure 7 was obtained by cascading two four band splits. The data shown in Figure 5 was subbanded again using the same algorithm. A third cascade of the four band algorithm produced the sixty-four band split shown in Figure 8. A close-up of the low band is shown in Figure 9. The low band is a very low resolution version of the original, but it is usable as a browse picture and is only 64 x 64 pixels in size.

Figure 10 shows a perfect reconstruction of Lenna from a four band split using the inverse transform g from Equation (3).

To investigate the effect of subbanding on lossless compression of images, some subbanded image files were compressed with a public domain program that is based on the Lempel-Ziv coding method. Files subbanded into four, sixteen, and sixty-four bands using both H and G transform matrices were compressed and compared to the original image file size. The original images did not compress well with the LZ coder.

At first, the subbanded values were truncated to 8 bits before compression, resulting in a lossy compression. Since the LZ coder is designed to work on text characters and since the bits that were dropped are expected to contain mostly noise, this approach seemed a logical first step.

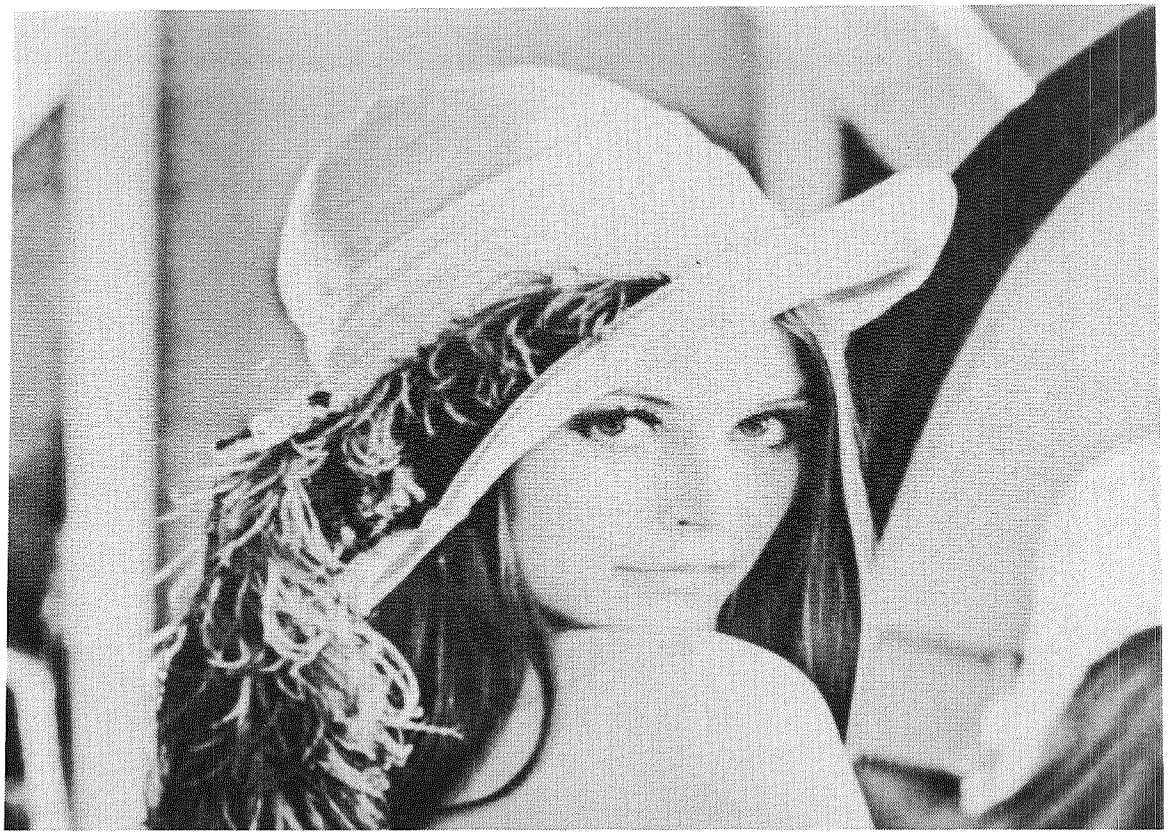


Figure 4. Original image of girl, "Lenna."

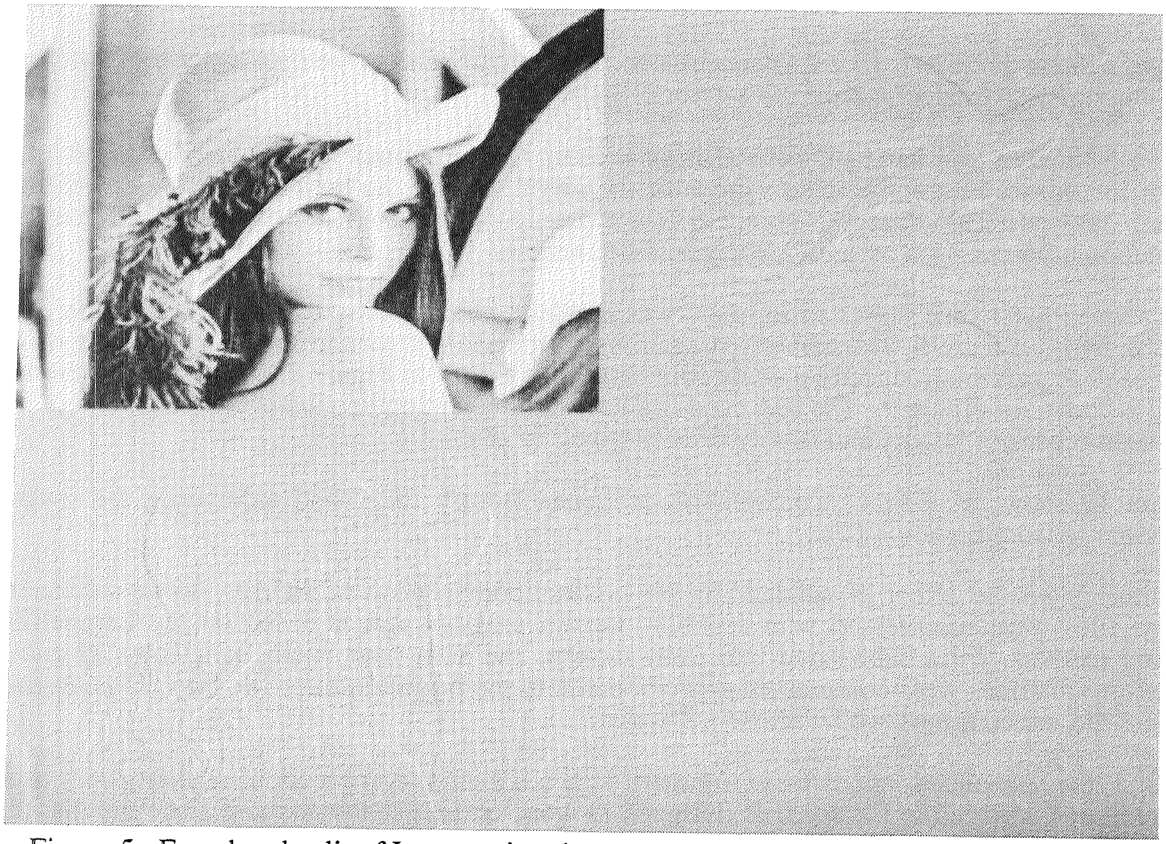


Figure 5. Four-band split of Lenna using the non-orthogonal transform of Equation (3).

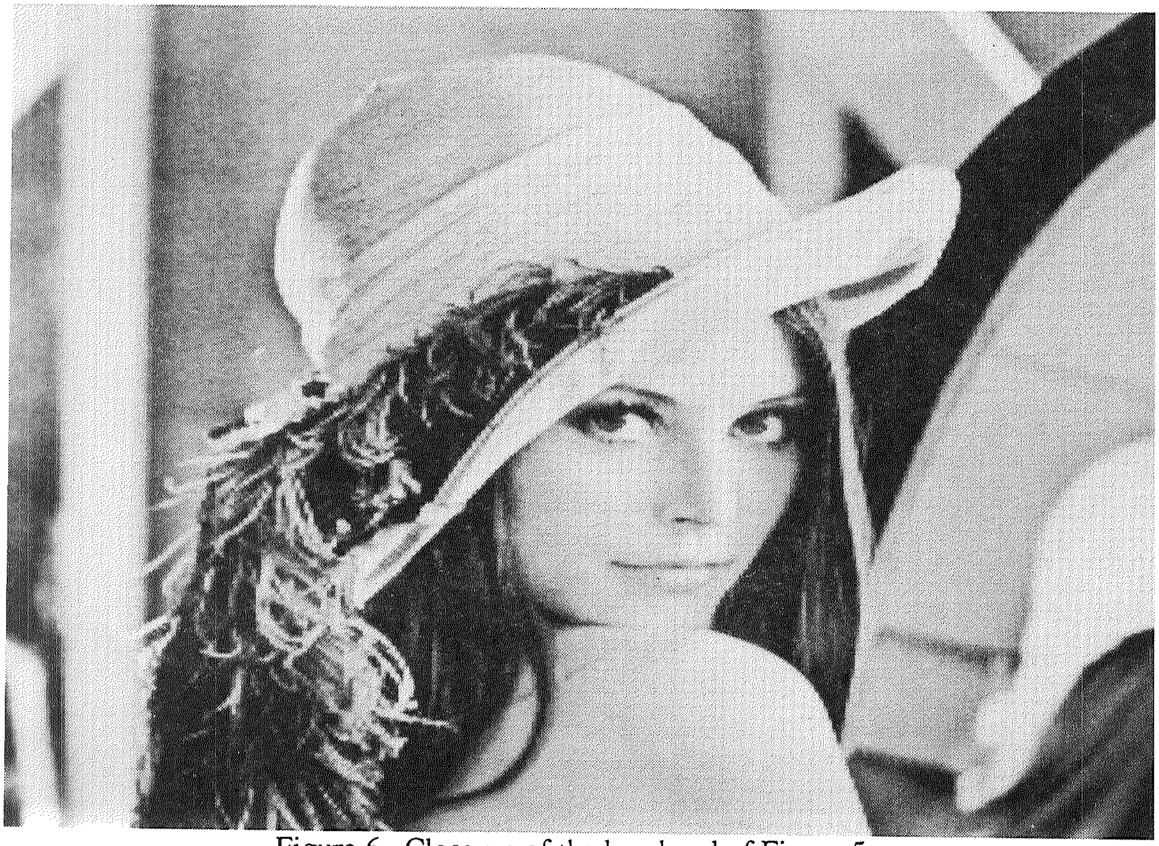


Figure 6. Close-up of the low band of Figure 5.

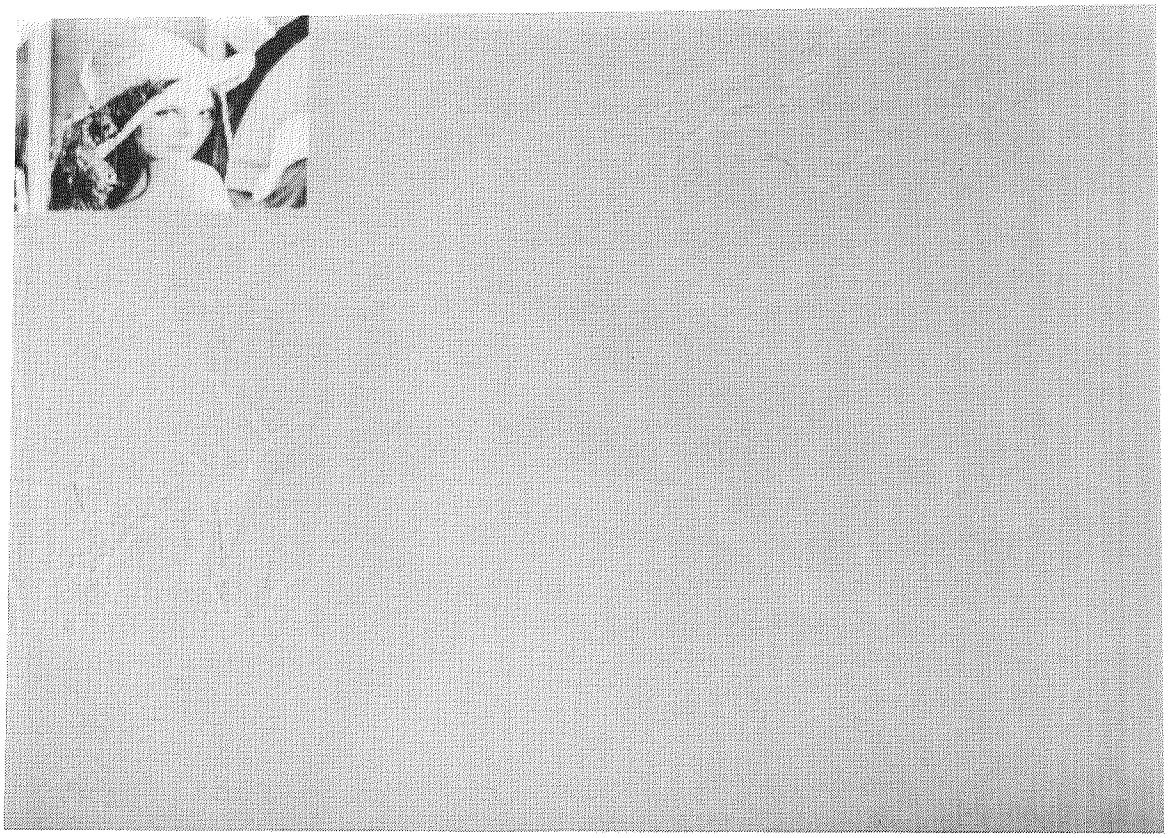


Figure 7. Sixteen-band split of Lenna obtained by applying the transform of Equation (3) to the data shown in Figure 5.

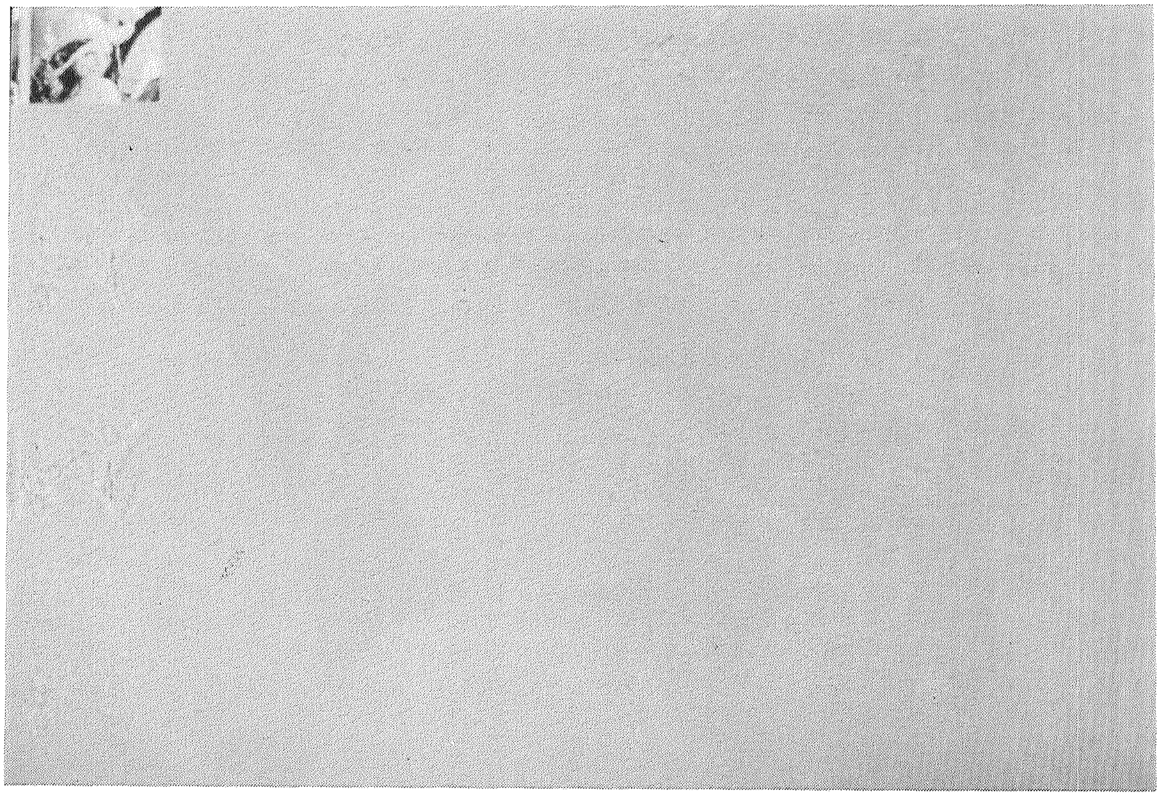


Figure 8. Sixty-four-band split of Lenna obtained by applying the transform of Equation (3) to the data shown in Figure 7.

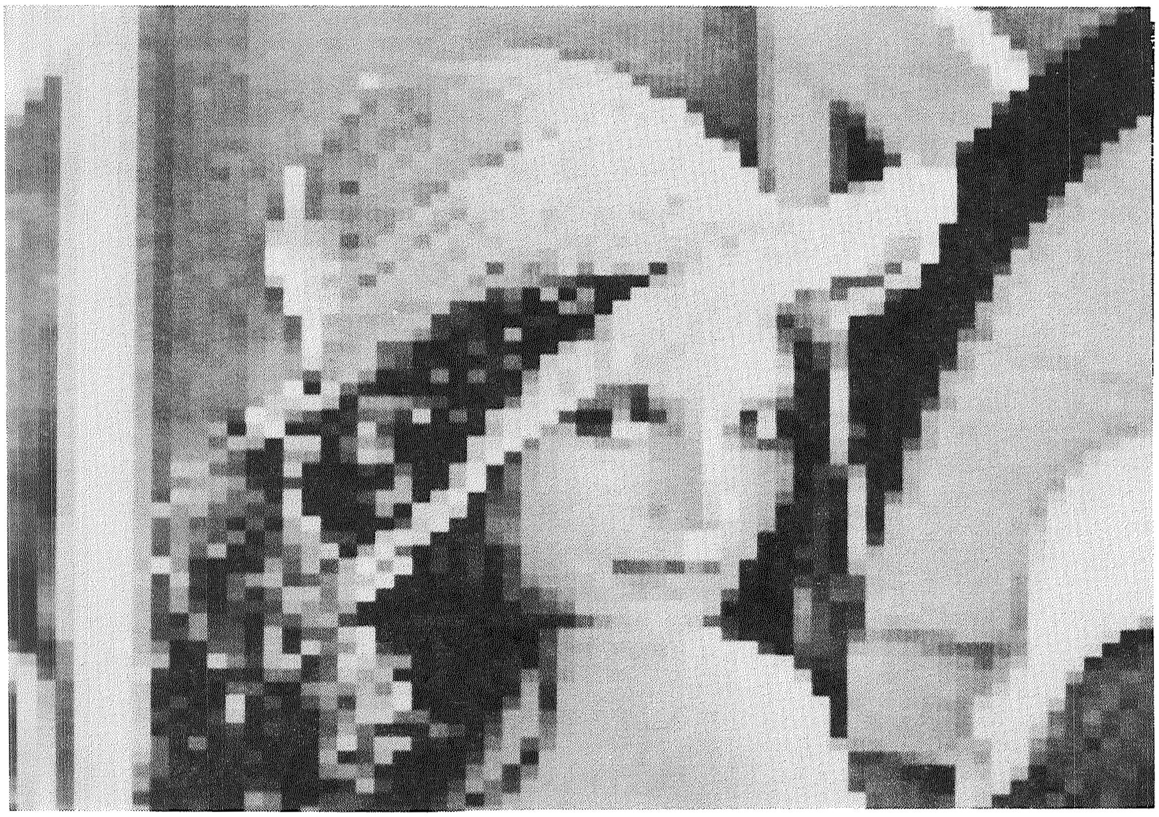


Figure 9. Close-up of the low band of Figure 8.

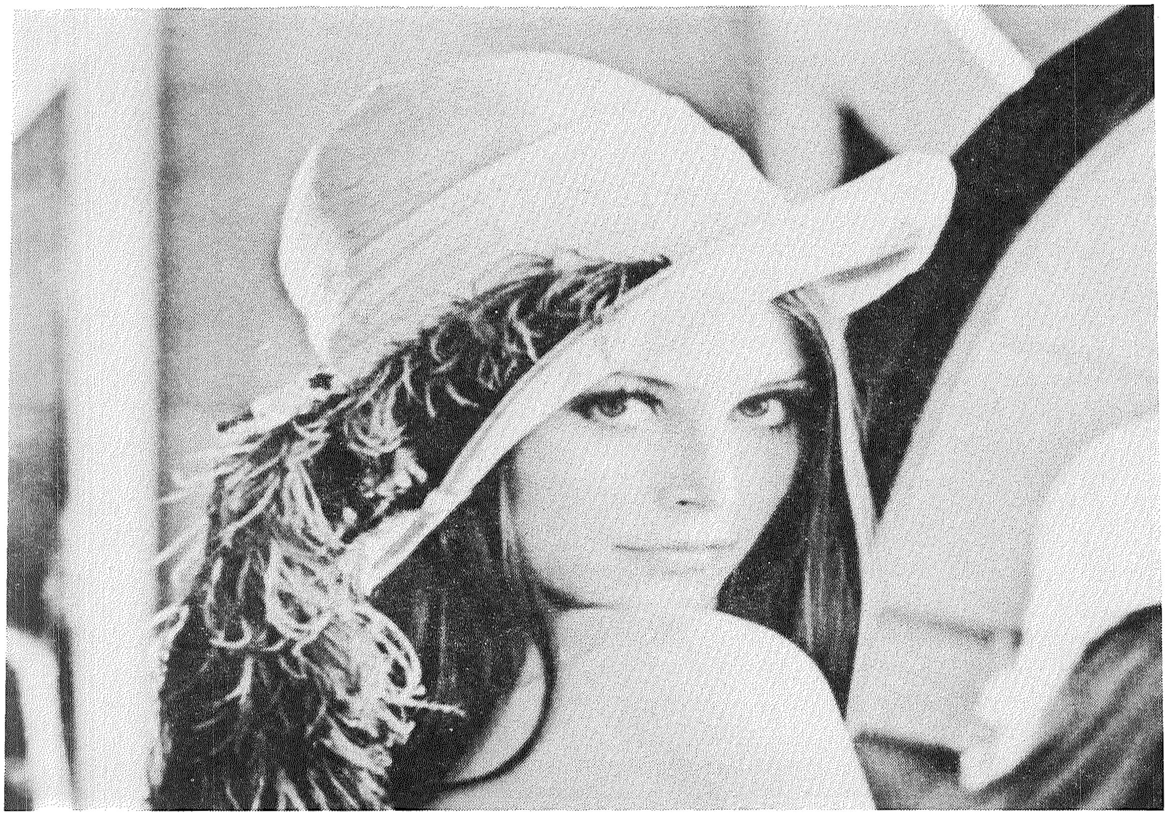


Figure 10. Perfect reconstruction of Lenna following a four-band split.

The results are shown in Table 1. The more subbands an image is separated into, the better the compression. The quality of the reconstructed image remains very good since only a small amount of information is lost in truncation. The ***H*** transform out performed the ***G*** because when the values were truncated, more signal energy was removed in the ***H*** transform case (i.e., there is less loss in the ***G*** case). A comparison of the subband value size between ***H*** and ***G*** is shown in Figure 11.

For true lossless compression the full value for each subband must be kept. This is most easily done by storing each value as a 16 bit word. Since the lowest band of a ***G*** transformed image is still only 8 bits, that band can still be stored in bytes. Using ***H***, all bands have the same value size (dynamic range) and so require an extra 2 bits per value for every subband stage.

Since noise is not generally compressible, we would expect the lossless compression ratios to be worse than the truncated ratios of Table 1. Table 2 shows the lossless case for the four band subbanded images. Although the ***H*** transformed data actually expands instead of contracts, improvements could be easily obtained by packing the data properly before compressing. For example, adding the extra 2 bits per value (for a total of 65 536 bytes) to the four-band truncated compressed file in Table 1 yields a total file size of 263 630 for the Baboon image. The *G* transform does much better, probably due to the 8 bit low band values.

The results of Table 1 show that subbanding image data can improve the effectiveness of LZ compression. The more the data is subbanded, the better the compression ratios. For the truncated values used in these tests, the ***H*** transform matrix outperformed the ***G*** because more noise was removed in truncation.

The results in Table 2 show, in a preliminary way, the advantage of the ***G*** matrix in lossless coding. The ***G*** transform produces an average of 1 bit/pixel smaller value size than *H* which gives it a head start in compression. The values in Table 2 show a trend, but the bits have not been packed in an efficient fashion. However, computer based methods tend to use byte and 16 bit word size values. Dense packing of bits requires more manipulation.

The interesting case of the Baboon image subbanded with ***G*** compressing better when the higher bands are stored as 16 bit values than truncated bytes is probably due to random chance. *G* out performs ***H*** for lossless compression in this simple case. Future research will be into more efficient packing methods.

7. Concluding Remarks

Subbanding image data can be advantageous for browsing and progressive resolution in addition to data compression. Subband systems can be designed to provide perfect reconstruction. The use of short kernel filters allows fast, simple implementations for real-time use or fast retrieval.

Two methods of subbanding image data with simple filters were presented. These methods can be expanded or cascaded to work on larger block sizes easily [12]. Subbanding image data is a promising method of obtaining better performance from lossless coders.

Acknowledgments

This work is based on research that was done in partial fulfillment of the requirements for the Ph.D. degree at the University of Toledo. This work was funded by NASA Lewis Research Center. Thanks to Mr. Paul G. Mallasch of Analex Corp. for coding the algorithms and to Mrs. Linda Gaydosh for preparing the manuscript.

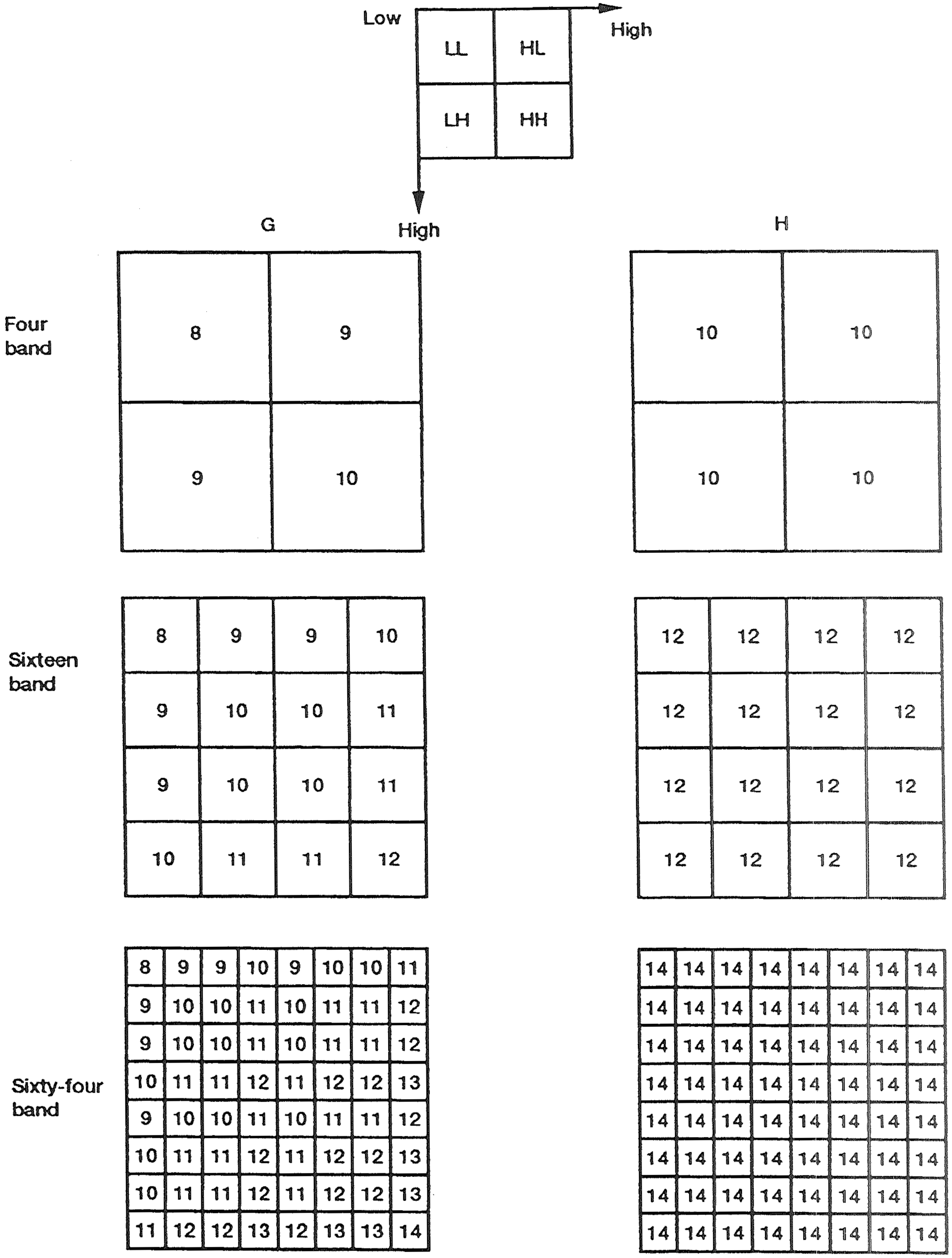
Table 1. Lossy Compression Results using an LZ Coder on subbanded images with all values truncated to 8 bits.

			File Size		
Image	Number of bands	Transform matrix	Original	Compressed	Ratio
Lenna	1	---	262 148	228 196	1.15:1
	4	*H*		166 343	1.58:1
	4	*G*		173 011	1.52:1
	16	*H*		121 418	2.16:1
	16	*G*		142 832	1.84:1
	64	*H*		88 383	2.97:1
	64	*G*		125 100	2.10:1
Baboon	1	—		239 005	1.10:1
	4	*H*		198 094	1.32:1
	4	*G*		202 833	1.29:1
	16	*H*		166 275	1.58:1
	16	*G*		184 933	1.42:1
	64	*H*		136 257	1.92:1
	64	*G*	↓	171 851	1.53:1

Table 2. Lossles Compression Results using an LZ Coder on Subbanded Images Saved as 16 Bit Values Except Low Band of *G* Transform Which Are Saved in Bytes.

				File Size		
Image	Number of bands	Transform matrix	Average pixel size, bits	Original	Compressed	Ratio[a]
Lenna	4	*H*	10	524 292	303 509	0.86:1
		G	9	458 756	254 594	1.03:1
Baboon		*H*	10	524 292	371 813	0.71:1
	↓	*G*	9	458 756	196 535	1.33:1

[a] Ratio of original 262 148 byte file to compressed.



Four band, G

8	9
9	10

Four band, H

10	10
10	10

Sixteen band, G

8	9	9	10
9	10	10	11
9	10	10	11
10	11	11	12

Sixteen band, H

12	12	12	12
12	12	12	12
12	12	12	12
12	12	12	12

Sixty-four band, G

8	9	9	10	9	10	10	11
9	10	10	11	10	11	11	12
9	10	10	11	10	11	11	12
10	11	11	12	11	12	12	13
9	10	10	11	10	11	11	12
10	11	11	12	11	12	12	13
10	11	11	12	11	12	12	13
11	12	12	13	12	13	13	14

Sixty-four band, H

14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14
14	14	14	14	14	14	14	14

Figure 11. Pixel value size by subband for ***G*** and ***H*** transforms, cascading the four-band analysis.

References

[1] R. E. Crochiere, S. A. Webber, J. L. Flanagan: "Digital coding of speech in sub-bands," Bell Sys. Tech. J., vol. 55, pp. 1069-1085, October 1976.

[2] A. Crosier, D. Esteban, C. Galand, "Perfect channel splitting by use of interpolation, decimation and tree decomposition techniques," Proc. of Int. Conf. on Information Sciences/Systems, Patras, Greece, pp. 443-446, August 1976.

[3] M. Vetterli, "Multi-dimensional subband coding: some theory and algorithms," Signal Processing, vol. 6, pp. 97-112, February 1984.

[4] J. W. Woods and S. D. O'Neil, "Sub-band coding of images," IEEE Trans. ASSP, vol. ASSP-34, pp. 1278-1288, May 1986.

[5] A. v. Brandt, "Sub-band coding of videoconference signals using quadrature mirror filters," Applied Sig. Processing and Dig. Filtering, Proc. IASTED Int. Symp, Paris, pp. 212-215, June 17-21, 1985.

[6] M. J. T. Smith, T. P. Barnwell, "A procedure for designing exact reconstruction filter banks for tree structured sub-band coders," Proc. IEEE ICASSP-84, San Diego, pp. 27.1.1-27.1.44, March 1984.

[7] R. E. Crochiere, L. R. Rabiner, Multirate Digital Signal Processing, Englewood Cliffs, NJ: Prentice Hall, 1983.

[8] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," Proc. IEEE, vol. 78, no. 1, pp. 56-93, January 1990.

[9] J. W. Woods (ed), Subband Image Coding, Boston: Kluwer, 1991.

[10] J. S. Lim, Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ: Prentice Hall, 1990.

[11] A. Al-Asmari, S. C. Kwatra, "Bandwidth compression of HDTV images for communication via satellite," to be presented at the 14th AIAA Conf. on Comm. Sat.

[12] D. Glover, "Subband/transform image coding for lossy and lossless compression," Ph.D. dissertation, The University of Toledo, in preparation.