Experimental Studies on a Compact Storage Scheme for Wavelet-Based Multiresolution Subregion Retrieval*

A. Poulakidas, A. Srinivasan, O. Egecioglu, O. Ibarra, T. Yang
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

Wavelet transforms, when combined with quantization and a suitable encoding, can be used to compress images effectively. In order to use them for image library systems, a compact storage scheme for quantized coefficient wavelet data must be developed with a support for fast subregion retrieval. We have designed such a scheme and in this paper we provide experimental studies to demonstrate that it achieves good image compression ratios, while providing a natural indexing mechanism that facilitates fast retrieval of portions of the image at various resolutions.

1 Introduction

Wavelet based algorithms have become popular for image compression. These typically involve three steps: (i) the application of a wavelet transform to an image to create coefficient matrices (ii) the quantization of these floating point matrices to create integer matrices, and (iii) encoding the quantized matrices to obtain the compressed image. The original image can be reconstructed by reversing the steps mentioned above. Some loss in the quality of the image results from the fact that the second step cannot be inverted exactly. The three steps are discussed briefly below.

A 2-dimensional wavelet transform maps a given image to another that resembles the original, with half the resolution. Three coefficient (or detail) matrices are also created. This process is referred to as *decomposition*. The detail matrices can be combined with the resembling image, through the application of a 2-dimensional inverse wavelet transform, to reproduce the original image. This inverse transform is also referred to as *reconstruction*. This process is illustrated in Figure 1. The decomposition described above can be recursively applied to the resembling image to produce smaller images at lower resolutions, as shown in Figure 2. This recursive process is carried out to a prescribed maximum depth, which depends on the application. The smallest resulting image obtained is called the *thumbnail*.

*Supported by NSF/NASA/ARPA Grant No. IRI94–11330

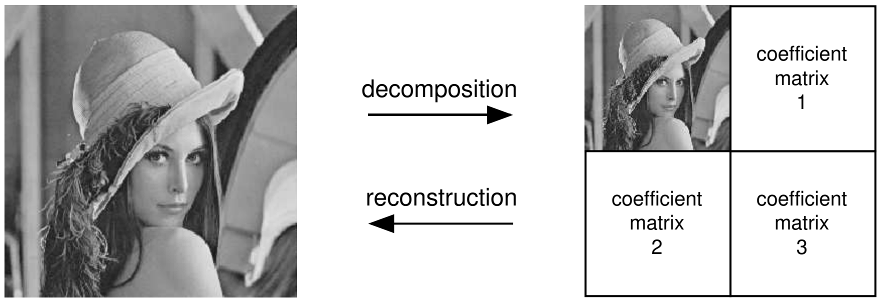

Figure 1: Image decomposition and reconstruction.

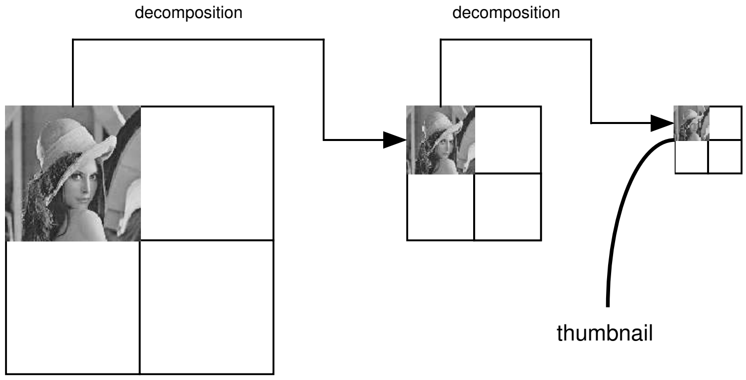

Figure 2: Multi-level decomposition of an image.

The coefficient matrices obtained as a result of the previous step consist of, in general, floating point numbers. These floating point numbers are then quantized to integers in a smaller range. This step enables efficient compression of the coefficient matrices.

In the final step, further compression of the coefficient matrices is achieved by encoding the quantized matrices. The encoding schemes usually utilize the fact that the histogram of the quantized coefficients typically has a narrow distribution.

In the Alexandria Digital Library (ADL) project at UC Santa Barbara, wavelets are used to support the hierarchical image storage and browsing: A typical ADL user will issue a query to access a set of thumbnails. Then the user will browse through the thumbnails, select the ones that he/she wants, and enlarge them to some desired resolution (or, more often, access just a subregion of a thumbnail of interest), using the wavelet reconstruction. We briefly discuss below the utility of a wavelet based compression scheme in the ADL environment.

1. The image collection includes a large number of satellite and aerophoto images with an average size of 30MB. The storage scheme must be able to achieve a good *compression* ratio in order to store these images efficiently. Apart from influencing the storage requirement, effective compression also reduces the time required for reconstructing the image since the time taken for reading from the disk is reduced if the image is stored in less space. Wavelet based schemes have proven effective in yielding substantial compression.

2. The size of the images are usually too large to display in their entirety at the highest

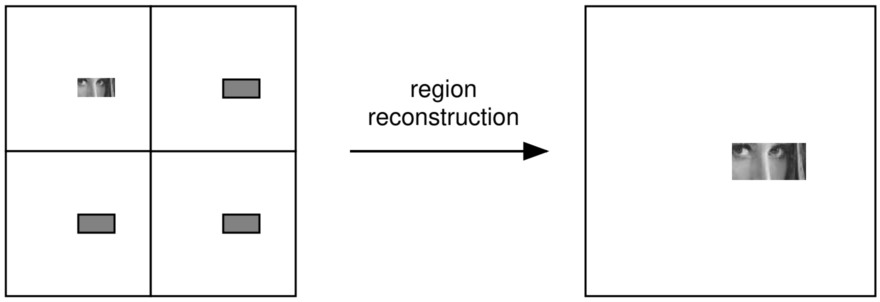

Figure 3: Region reconstruction.

possible resolution. Therefore the user typically either views the whole image at a lower resolution, or views a subregion of the image at a possibly high resolution, as shown in Figure 3. Retrieving a portion of an image from compressed data requires a sophisticated indexing scheme to make access fast. The challenge is to design a storage scheme with a fast indexing mechanism such that the compression ratio is not compromised while trying to reduce the time to access a subregion.

There is extensive literature available on selecting the appropriate wavelet transform (e.g., [1, 7]), and for the quantization that will provide compression by minimizing the bit allocation, while preserving much of the image quality (e.g., [9, 3, 4, 2]). To further reduce storage requirements, often an extra phase of compression is applied to the quantized values, typically using entropy coding, hybrid schemes using run-length/Huffman coding [1], or other methods [8, 6]. In this paper, we describe a scheme that provides an alternative technique for this phase, which results in similar compression performance but with superior subregion retrieval speed.

The storage scheme we implemented works on any naturally quantized coefficient matrix as input. Furthermore, care has been taken in the implementation of the storage scheme to allow flexibility on the data placement. In the experiments reported in this paper, all data are stored on a single disk but the algorithm can be easily extended to multiple disks or on hierarchical storage media.

We describe the storage scheme in section 2 and report on the experimental results in section 3.

2 The storage scheme

The effective compression ratio of our scheme depends on the histograms of coefficient matrices. Theoretically speaking, any distribution of histograms is possible; however, in practice they are symmetrical peaks centered at zero [7], similar to the one shown in Figure 4. Using an appropriate quantization, the resulting matrices will contain many zeros. Furthermore, because of the locality observed in most images and the fact that coefficient matrices consist of difference values among weighted neighboring pixels, it is expected that very often many of the zeros in the quantized matrices will be adjacent and grouped to form 2-dimensional regions consisting of zeros.

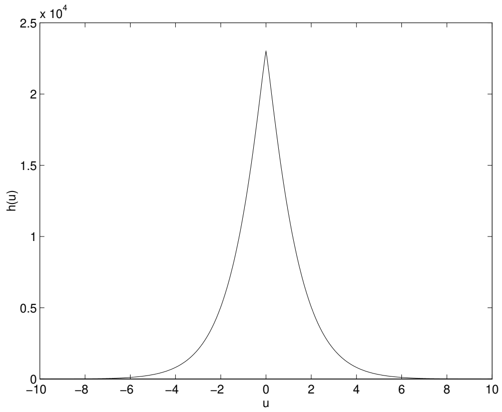

Figure 4: A typical histogram of a coefficient matrix produced by wavelet decomposition.

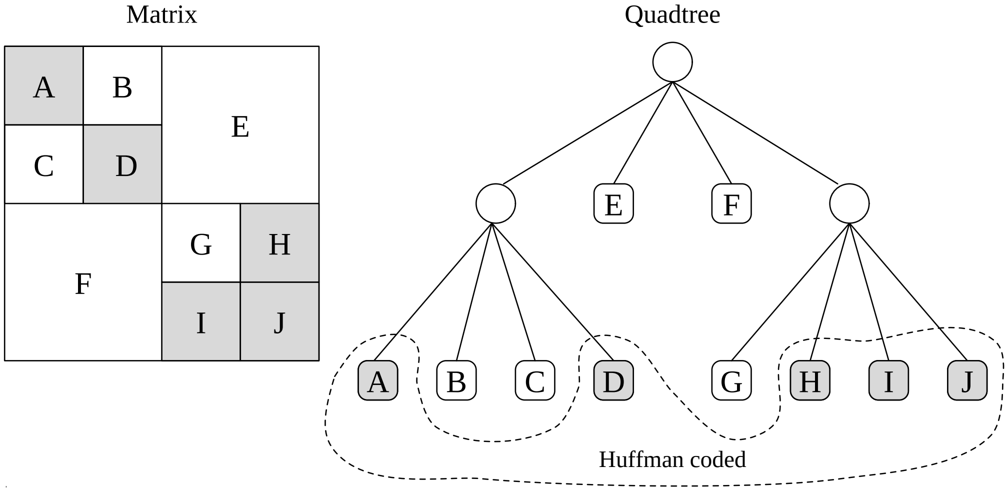

Figure 5: The combination of quadtrees and Huffman coding in our storage scheme.

A classical solution is to use *quadtrees* to compress coefficient matrices. Quadtrees are useful in eliminating unnecessary storage of 2-dimensional blocks of zeros. Furthermore, they provide a natural indexing mechanism to easily access image subregions as depicted in Figure 5. However, quadtrees built on image data usually are large, overhead for accessing and composing subregions is high and also compression ratio is not as effective as other methods such as Huffman coding.

The peaked histograms of coefficient matrices suggest that the non-zero data in quantized matrices will compress well if *Huffman coding* is used[1]. For this reason we have used a hybrid method that takes advantages of both quadtree and Huffman coding: 1) Use the quadtree method to partition an image and 2) compress all the non-zero leaf areas of the quadtree using Huffman coding (see Figure 5).

The peaks of the histograms of coefficient matrices become shorter and wider as the level

[1]Actually, this is not necessarily true if the quantized scheme does not use a constant bin size for each coefficient matrix. We have not experimented with quantization schemes that use many bin sizes for each coefficient matrix.

of the decomposition increases. However, this does not affect the performance of our storage scheme, since the size of coefficient matrices becomes smaller as the level of decomposition increases. Note that just the matrices at the first level of decomposition add up to more than 75% of the total space size of coefficient matrices.

2.1 Remarks on the storage scheme

- We needed to eliminate internal fragmentation at the byte level, since often the compressed size of the non-zero leaf areas is not a multiple of 8 bits.
- A compact representation of a quadtree is used. This is important since in many cases the size of the quadtree takes more than 50% of the total storage required.
- An adaptive control mechanism is provided that automatically prunes a quadtree to improve the compression, while making sure that the desired trade-off with the region retrieval speed is not violated. There is a trade-off in those coefficient matrices that do not compress well under quadtrees, so they require a minimal quadtree in order to compress well. But if the quadtree is too small, insufficient indexing may lead to a slow region retrieval. The desired trade-off is determined by a parameter that specifies the maximum non-zero area that is allowed to be a quadtree leaf.
- Buffering is used to reduce the number of disk reads when retrieving many small non-zero areas.

3 Experiments

We demonstrate the performance of our scheme through experimental results. In particular, we test our scheme against the following three criteria: compression ratio, reconstruction quality, and region retrieval speed.

The wavelet filters used in our experiments were designed to be very fast in reconstruction, since they are just 3-tap and only require integer operations[2]. But they violate the QMF criteria, and provide a slight degradation in reconstruction quality [1].

We tested our scheme with a very simple quantization scheme: the bin sizes are the same for all coefficient matrices at any given level of decomposition, and they have half the width of the bin sizes at the previous level of decomposition. The bin size at the first level of decomposition is user-specified. A more sophisticated quantization should provide better reconstruction quality for a given compression ratio.

We compare the performance of our scheme with that of a typical storage scheme that uses a combined run-length/Huffman entropy coder. Of course, for the comparisons to be legitimate, this scheme is using the same wavelets and quantization scheme as our storage scheme.

We have experimented extensively with several representative images from the ADL (in particular, aerophoto and satellite images) and texture image collections. For the time being,

[2]The reconstruction filters are [1 2 1] and [-1 2 -1].

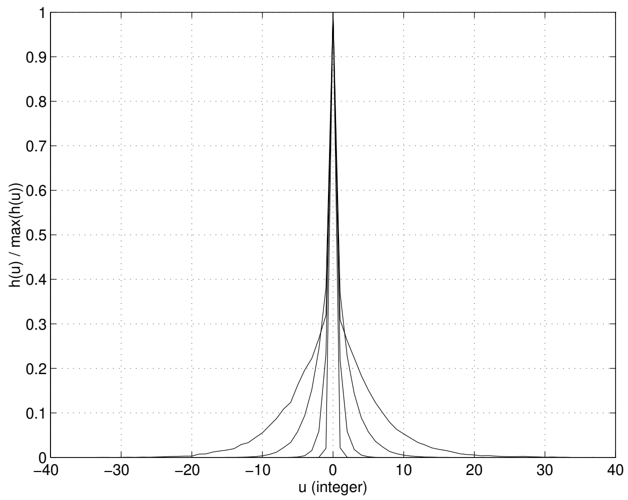

Figure 6: Typical observed histograms of the quantized coefficients, for varying levels of decomposition. The wider the histogram the larger the level of decomposition. The histograms have been normalized so that their height is 1.

we considered images that allocate one byte per pixel, and have sizes varying from 65KB to 4MB.

3.1 Compression ratios

The bin size used during quantization were set such that the distortion in the reconstructed regions was not visually disturbing. In Figure 6 the typical peaks of the observed quantized coefficient matrices are shown. The narrower the peak the better it compresses. Note that although the peaks at deeper levels of decomposition become wider, the size of the corresponding matrices becomes insignificant —the first decomposition level coefficients are more than 75% of the total.

The typical compression we have obtained ranged from 75% to 90%. The worst compression experienced was 72% for some texture images. These images have been shown hard to compress by other research groups and we expect around 70% to be very close to the worst case compression. The compression ratios achieved when using run-length/Huffman were similar. Furthermore, as one might expect, the compression ratio seems to improve as we consider larger images.

3.2 Reconstruction Quality

We measure the reconstruction quality using the peak-signal-to-noise ratio (PSNR) criterion. The PSNR for an $N \times M$ 8-bit image I and its reconstructed equivalent $\hat{I}$ is given in dB by

$$\text{PSNR} = 20 \log_{10}\left(\frac{255}{\sqrt{\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}[I(i,j) - \hat{I}(i,j)]^2}}\right)$$

The higher the PSNR the higher is the reconstruction quality. The compression ratios reported in the previous subsection were achieved with PSNR greater than 32 dB.

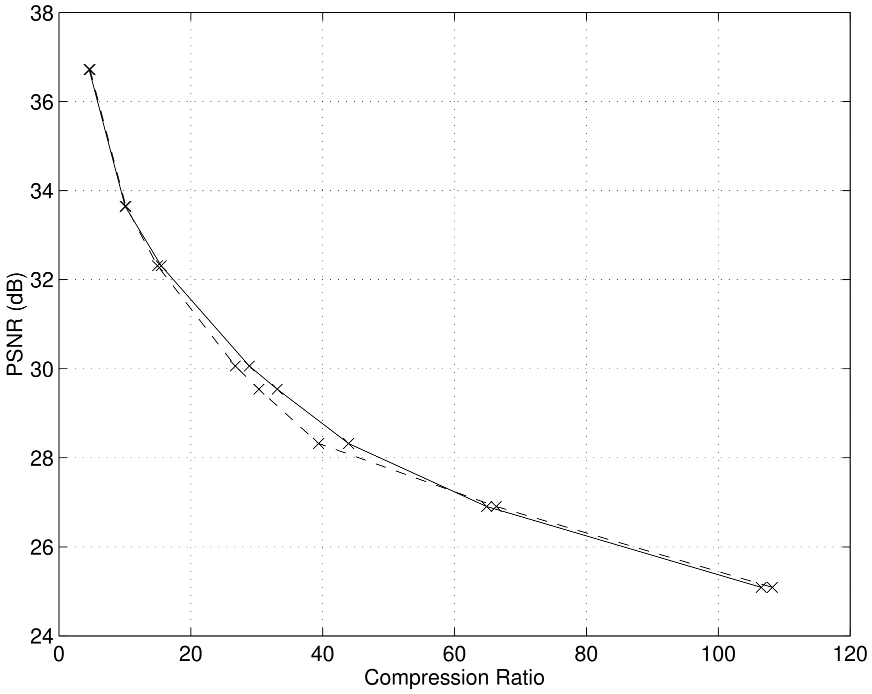

Figure 7: A comparison of the image reconstruction quality versus the compression ratio. The tests were performed on the 512×512, 8-bpp Lena image. The solid line refers to our method and the dashed to the run-length/Huffman based method.

The effect of changing the bin size in compression and reconstruction quality is shown in Figure 7, compared with run-length/Huffman. [5] provides results for other methods.

3.3 Region Retrieval Speed

We run our experiments on a SUN SPARC 5 (75Mhz) workstation with its own local SCSI-2 disk.

To test our region retrieval time we compared with the same run-length/Huffman coding scheme. Its disadvantage lies in that for each region retrieval from a matrix, this scheme requires the whole matrix to be loaded and uncompressed. However, while data loaded are more than that in our scheme, the number of disk seeks required is minimal. Once data is loaded, only those portions needed to reconstruct the desired region are actually used, and the computation is the same for both methods.

In Figures 8 and 9 the two schemes are compared. Figure 8 shows the running times when fully reconstructing 128x128, 256x256, 512x512 and 1024x1024 regions. Figure 9 shows the running times for the reconstruction of a region of typical size (512x512) for 1 to 5 levels.

4 Current Work

We have been extending our scheme to a client/server environment, were coefficient data are progressively delivered by the server to the client, where the reconstruction is performed.

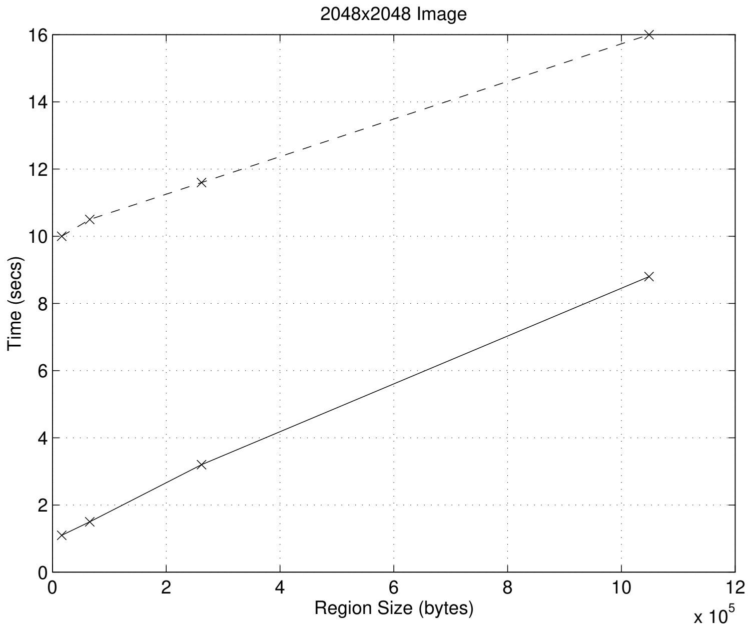

Figure 8: Retrieval+Reconstruction times for various region sizes. The regions were reconstructed for 5 levels. The solid line refers to our method and the dashed to the run-length/Huffman based method.

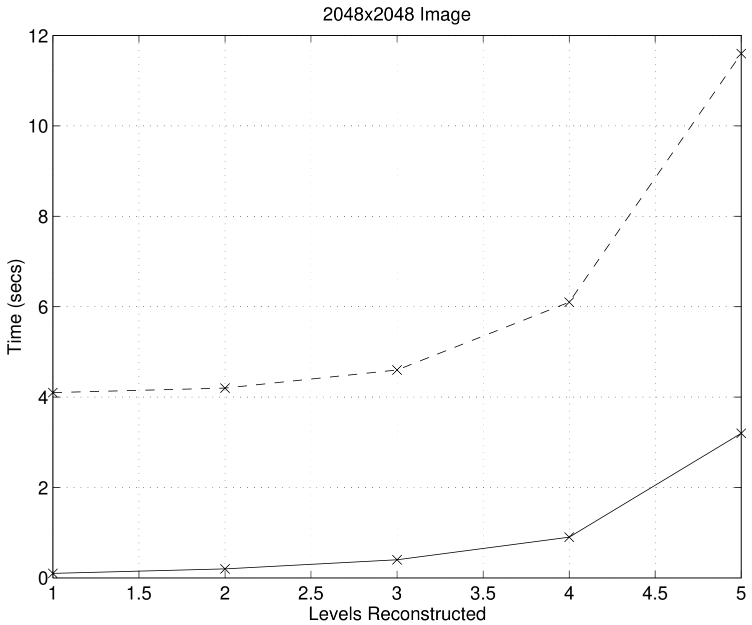

Figure 9: Retrieval+Reconstruction times for a 512x512 region for various levels of reconstruction. The solid line refers to our method and the dashed to the run-length/Huffman based method.

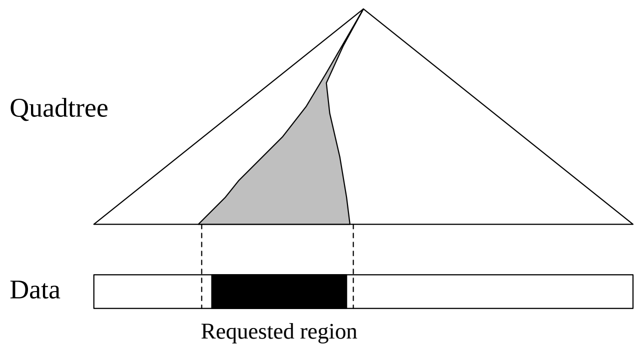

Figure 10: One need only have access to the subtree of the quadtree that "covers" the requested region, in order to be able to reconstruct it. This is demonstrated here for an 1-D region. The 2-D region is a direct generalization.

Our scheme adapts very naturally to this environment, since the server does not have to uncompress the data before sending them. As indicated in Figure 10, only the part of the quadtree that "covers" the requested region and the Huffman-compressed non-zero areas at the region need to be sent to the client. Since the data are sent compressed to the client, the server work is less and network bandwidth is saved.

References

[1] E. H. Adelson and E. Simoncelli, "Subband Image Coding with Three-tap Pyramids," *Picture Coding Symposium*, 1990, Cambridge, MA.

[2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image Coding Using the Wavelet Transform," *IEEE Transactions on Image Processing*, 1(2), April 1992.

[3] M. Barlaud, P. Sole, T. Gaidon, M. Antonini, and P. Mathieu, "Pyramidal Lattice Vector Quantization for Multiscale Coding," *IEEE Transactions on Image Processing*, 3(4):367–381, July 1994.

[4] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image Compression Through Wavelet Transform Coding," *IEEE Transactions on Information Theory*, 38(2):719–746, March 1992.

[5] M. L. Hilton, B. D. Jawerth, A. Sengupta, "Compressing Still and Moving Images with Wavelets," *Multimedia Systems*, 2(3).

[6] A. S. Lewis and G. Knowles, "Image Compression Using the 2-d Wavelet Transform," *IEEE Transactions on Image Processing*, 1(2):244–250, April 1992.

[7] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[8] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[9] N. Strobel, S. K. Mitra, and B. S. Manjunath, "An Approach to Efficient Storage, Retrieval, and Browsing of Large Scale Image Databases," *Photonics East, Proceedings of the SPIE —The International Society for Optical Engineering*, to appear.