WaveMark: Digital Image Watermarking Using Daubechies' Wavelets and Error Correcting Coding

James Ze Wang and Gio Wiederhold

Stanford University, Stanford, CA 94305, USA

ABSTRACT

As more and more digital images are distributed on-line via the Internet and World-Wide Web, many copyright owners are concerned about protecting the copyright of digital images. This paper describes WaveMark, a novel wavelet-based multiresolution digital watermarking system for color images. The algorithm in WaveMark uses discrete wavelet transforms and error-correcting coding schemes to provide robust watermarking of digital images. Unlike other wavelet-based algorithms, our watermark recovery procedure does not require a match with an uncorrupted original image. Our algorithm uses Daubechies' advanced wavelets and extended Hamming codes to deal with problems associated with JPEG compression and random additive noise. In addition, the algorithm is able to sustain intentional disturbances introduced by professional robustness testing programs such as StirMark. The use of Daubechies' advanced wavelets makes the watermarked images more perceptively faithful than the images watermarked with the Haar wavelet transform. The watermark is adaptively applied to different frequency bands and different areas of the image, based on the smoothness of the areas, to increase robustness within the limits of perception. The system is practical for real-world applications, encoding or decoding images at the speed of less than one second each on a Pentium Pro PC.

Keywords: Digital watermark, copyright protection, wavelet transform, error-correcting, color image

1. INTRODUCTION

With the rapid expansion of the Internet, more and more multimedia documents are distributed on-line. Some on-line databases[17–19] have image libraries containing about a million digital images and they are constantly growing. Protecting the copyright of digital images is increasingly a problem that many copyright owners, such as professional photographers and museums, are concerned about.

1.1. Related Work

There are many attempts to solve the problem of illegal image copying in the software industry. Among others, IBM T. J. Watson Research Center[2,11] has created a visible watermarking algorithm. Visible watermarking systems are usually able to sustain all possible image alterations and even intentional disturbances. However, image quality is significantly reduced.

Many proprietary software programs based on digital watermarking technologies such as *Digimarc*, *SureSign*, *InvisibleLink* or *TigerMark* are distributed commercially. They aim to protect the copyright of digital images. There are also software programs designed to hide digital watermarks in audio clips and video clips.

TigerMark watermarking algorithm[5] for multimedia data, developed by NEC Research, has been integrated with Informix Databalde technology. The algorithm uses discrete cosine transform (DCT) to insert marks into the perceptually most significant spectral coefficients of the signal. According to NEC, the

Send correspondence to: wangz@cs.stanford.edu

mark can be recovered after some common operations such as scaling, JPEG compression, dithering and clipping. However, a given image cannot be queried for ownership without the original unwatermarked image.

Recent work by University of Toronto and Delaware University[9,15] uses discrete wavelet transform (DWT) to encode images with invisible watermarks. Pseudo-random codes are added to the large coefficients at the high and middle frequency bands of the DWT of an image. However, the algorithm cannot resolve the ownership of an image without the uncorrupted original image.

StirMark[20] and *unZign*[21] are generic but sophisticated tools distributed freely on the Internet for robustness testing of image watermarking algorithms.[3,5,4] After they are applied to watermarked digital images, watermark of most commercial software cannot be detected or decoded. These tools are designed so that the downgrading of image quality is at the same extent as that of the commercial digital watermark algorithms. Copyright thieves may utilize these available tools to destroy the invisible watermarks before redistributing copyrighted images.

To make such watermarking algorithms practical for the purpose of protecting copyright of images, extremely robustness with respect to common image format changes, intentional disturbances and image processing operations is necessary.

The problem of digital watermarking is extremely difficult due to the following reasons:

- digital watermarking algorithms usually use the lower-order bit-planes of the original image, so do intentional disturbance algorithms;
- digital watermarks cannot be inserted to downgrade the quality of the source image too much;
- copyright thieves are able to determine if the watermark exists in the image because digital watermark readers are usually widely available;
- there is a limited amount of data that can be used to insert digital watermarks in a highly compressed JPEG image;
- noticeable artifacts of image compression usually destroy watermarks easily.

1.2. Overview of Our Work

In the WaveMark project, we have developed a novel wavelet-based multiresolution digital watermarking system for color images. The algorithm in WaveMark uses discrete wavelet transforms and error-correcting coding schemes to provide robust watermarking of digital images. The watermark recovery procedure does not require a match with an uncorrupted original image. The use of Daubechies' advanced wavelets makes the watermarked images more perceptively faithful than the images watermarked with the Haar wavelet transform. The watermark is adaptively applied to different frequency bands and different areas of the image, based on the smoothness of the areas, to increase robustness within the limits of perception. The system is practical for real-world applications, encoding or decoding images at the speed of less than one second each on a Pentium Pro PC.

2. ENCODING ALGORITHM

2.1. Overview

The encoding algorithm of our image watermarking system uses Daubechies' wavelet transforms and error-correcting coding. Figure 1 shows the major steps in the encoding algorithm. The system is designed to place a 32-bit watermark in the image with high redundancy.

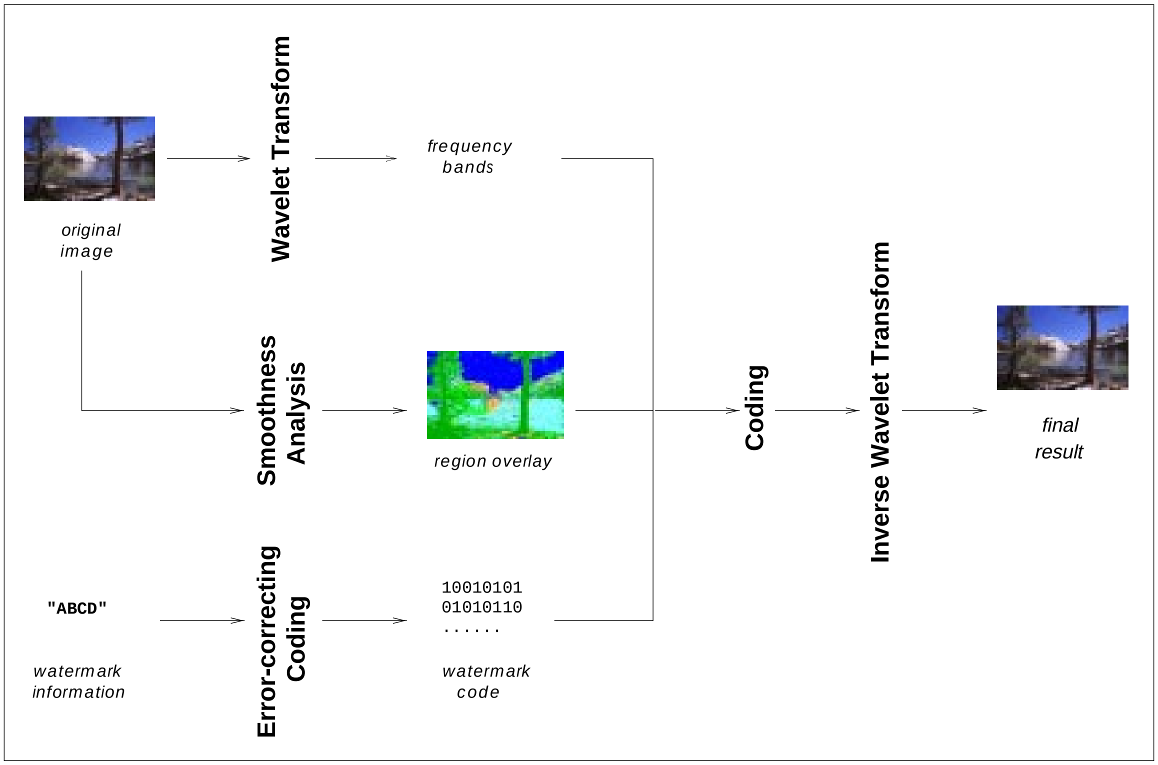


Figure 1. The encoding algorithm of the current experimental system.

2.2. Wavelet Transform

We first convert and store the image in a component color space with intensity and perceived contrasts. We define the new values at a color pixel based on the RGB values of an original pixel as follows:

$$\begin{cases} C_1 = (R+G+B)/3 \\ C_2 = (R+(255-B))/2 \\ C_3 = (R+2*(255-G)+B)/4 \end{cases} \quad (1)$$

Clearly, each color component in the new color space ranges from 0 to 255. This color space is similar to the opponent color axes[1]

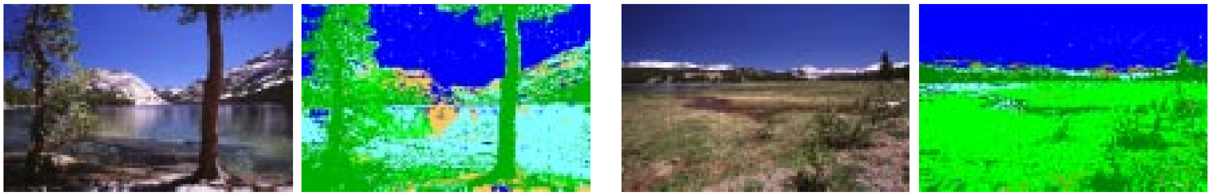
$$\begin{cases} RG = R-2*G+B \\ BY = -R-G+2*B \\ WB = R+G+B \end{cases} \quad (2)$$

For each color component, we perform a 4-level wavelet transform using Daubechies-4 wavelet. For very large images, a wavelet transform with a higher number of levels can be applied.

Daubechies' wavelets give remarkable results in image analysis and synthesis due to the efficiency of the transformation coefficients in representing images.[6–8,10] Daubechies' wavelet transform separates the image into clean distinct low frequency and high frequency parts. Various experiments and studies have shown that Daubechies' wavelets are well suited to deal with general-purpose images.[12,13,16]

2.3. Smoothness Analysis

We extract a rough smoothness region overlay for each image. We use the variances of 4×4 blocks in the intensity band to distinguish five different smoothness classes. Figure 2 shows the classification results on



color image *region overlay* *color image* *region overlay*

Figure 2. Smoothness analysis. Areas with similar smoothness are assigned a same color in the region overlay.

color images. Each of the five different classes is given a unique "pseudo" color in the final result. The classification results are satisfactory for our application.

For watermarking purposes, we apply watermark coding with lower strength to regions classified as highly smooth regions (such as sky) because watermark coding with higher strength in these regions introduces noticeable distortion. We apply watermark coding with higher strength to regions of lower levels of smoothness.

2.4. Error Correcting Coding

A Hamming code[14] is used to add redundancy to the bits so that the errors can be detected or corrected to a certain extent. Hamming code is a linear block code. The main advantage of linear block codes is their simplicity in implementation and low computational complexity. For details about the implementation of linear block codes, readers are referred to the book written by Wicker[14] . A linear block code is usually composed of two parts. The first part contains the information bits, which are the original bits to be transmitted. The second part contains the parity checking bits, which are obtained by summing over a subset of the information bits. A linear block code with length n and k information bits is denoted as a (n, k) code. Hamming codes are the earliest linear block codes appeared. They are well studied and their design is rather straight forward.

In this application, we use a $(8, 4)$ extended Hamming code. It is obtained from the original $(7, 4)$ Hamming code by adding an additional redundant bit. Such a code can correct a one-bit error and detect up to three one-bit errors. Due to the consideration of implementation convenience, we use byte as basic unit to calculate the Hamming code. With four bytes as input, the Hamming code encoder produces eight bytes. Since the first four bytes are the four input bytes, no calculation is needed. The extra four bytes are the parity checking bytes. The bits at the same bit position across the eight bytes form one code word of the extended Hamming code. This process is equivalent to encoding eight groups of bits in parallel, with four bits per group.

2.5. Inserting the Watermark Code

We partition the entire wavelet transform of each color component into 10×10 non-overlapping blocks. We alter the lower bits of the block borders to code '0' in order to assist the decoding process. With 8×8 matrix entries within each block, we are able to *hide* a 64-bit watermark code. As we have shown, the Hamming encoded watermark has 64 bits. We can simply encode each entry in the transformation matrices with one bit of the watermark code.

We alter the lower bits of the coefficients according to the frequency represented in the band, the smoothness region overlay, and the encoded watermark code. For the lower frequency bands of coefficients, we apply watermark coding with higher strength. That is, we alter higher order bits in these bands. Similarly, we alter higher order bits in regions with higher variations.

2.6. Inverse Wavelet Transform

After the watermark code is placed in the transform matrices, we perform a 4-level inverse wavelet transform for each of the three matrices using Daubechies-4 wavelet. Then we use the inverse color transformation

$$\begin{cases} R = C_1 + C_2 + 0.667 * C_3 - (213.0 + 0.333) \\ G = C_1 - 1.333 * C_3 + (170.0 + 0.667) \\ B = C_1 - C_2 + 0.667 * C_3 + (42.0 + 0.667) \end{cases} \tag{3}$$

to obtain a coded color image in the RGB color space. This image is the final image coded with the duplicated watermark codes.

3. DECODING ALGORITHM

3.1. Overview

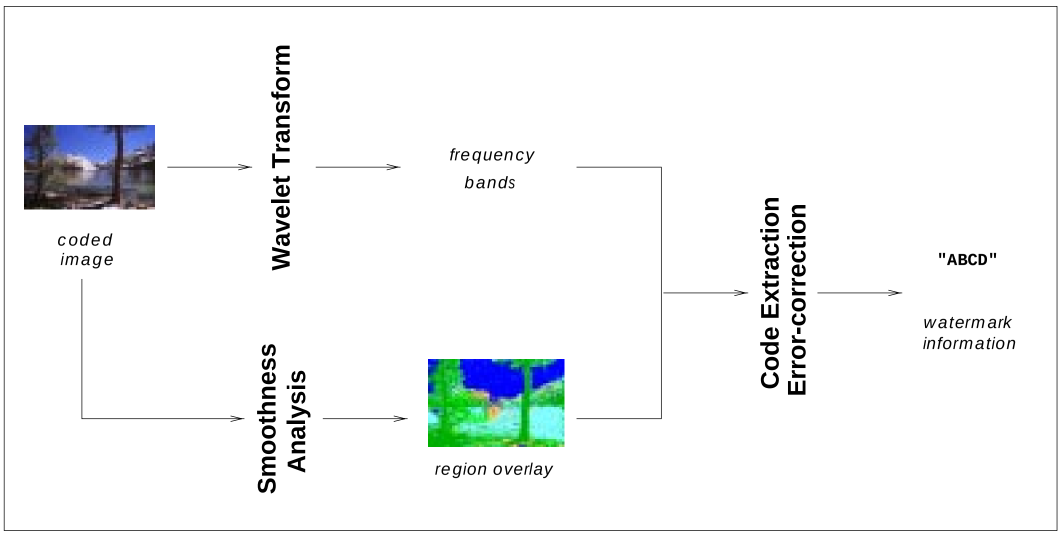


Figure 3. The decoding algorithm of the current experimental system.

The decoding algorithm of our image watermarking system is the inverse function of the encoding algorithm. It uses Daubechies' wavelet transforms and error-correcting coding. Figure 3 shows the major steps in the decoding algorithm.

3.2. Wavelet Transform and Smoothness Analysis

Similar to the encoding process, we first convert the original image into the $C_1C_2C_3$ opponent color space. Then we perform a 4-level wavelet transform on each color component.

We extract a rough smoothness region overlay for the image using the same algorithm as in the encoding process.

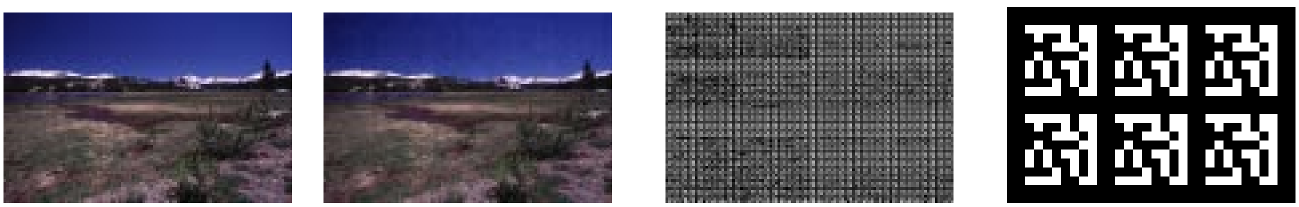
3.3. Extracting Watermark Code

For each coefficient in a transformation matrix, we first determine the strength of watermark coding it has been applied by checking the frequency band and the smoothness region it belongs. Then we can extract the watermark code by decoding the lower-order bit-planes.

After we have processed all the entries in the matrices, we obtain three binary matrices filled with watermark codes. Using the inverse function of the error-correcting coding algorithm we discussed in the previous section, we determine a set of watermark codes with high chance of correctness. Then we analyze the set of codes to obtain the final watermark code. Currently this step is done by voting on all the high-chance codes.

4. RESULTS

4.1. Performance Issues



original image *watermarked image* *watermark codes extracted* *zoom-in*

Figure 4. Results of watermarking using WaveMark.

This algorithm has been implemented on a Pentium Pro 300MHz workstation. We tested the algorithm on a database of about 100 digital photograph images mostly from CD-ROMs. It takes less than one second CPU time to encode or decode each image.

Besides the fast speed, the algorithm has achieved high reliability. Figure 4 shows the result of the algorithm on a sample image. Figure 5 shows the result of the algorithm on the same image under different alterations including compression, intentional disturbances and image processing operations. The algorithm is able to detect the invisible digital watermark within each altered image after performing Hamming error-correcting decoding.

4.2. Limitations of the Algorithm

WaveMark is designed to sustain image alterations such as compression, additive noise and even some intentional disturbances. However, like other wavelet-based watermarking algorithms, it is not suited to handle significant rescaling, aspect ratio changes and rotational transformations.

5. CONCLUSIONS

In this paper, we have demonstrated an efficient wavelet-based watermarking algorithm for digital images. The system uses Daubechies' advanced wavelets and extended Hamming codes to deal with problems associated with JPEG compression and random additive noise. The use of Daubechies' advanced wavelets makes the watermarked images more perceptively faithful. The system is practical for real-world applications, encoding or decoding images at the speed of less than one second each on a Pentium Pro PC.

This paper is still *work-in-progress*. We are attempting to make the algorithm more robust and to conduct a formal performance evaluation.

ACKNOWLEDGMENTS

We would like to thank Oscar Firschein, Visiting Scholar at Stanford University Computer Science Department, and Jia Li of Stanford University Electrical Engineering Department for discussions and help.

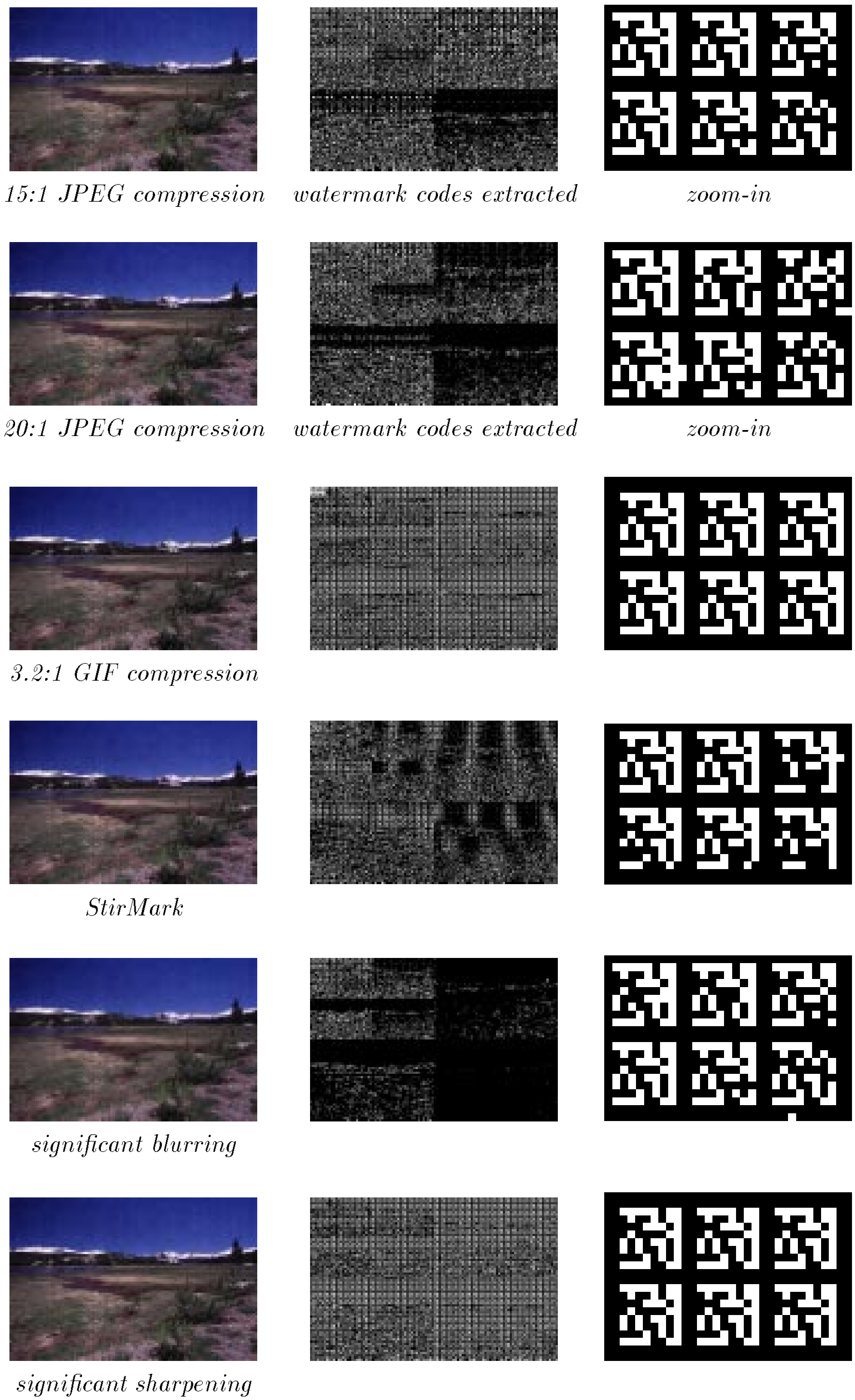


Figure 5. Results of watermarking using WaveMark.

REFERENCES

1. D.H. Ballard, C.M. Brown, *Computer Vision*, Prentice-Hall, Inc., NJ, 1982.
2. G W Braudaway, K A Magerlein, F Mintzer, Protecting Publicly-Available Images with a Visible Image Watermark, *Proc SPIE*, 2659:126-33, 1996.
3. G W Braudaway, Protecting Publicly-Available Images with an Invisible Image Watermark, *Proc IEEE ICIP*, 1:524-7, Oct. 1997.
4. S Craver, N Memon, B Yeo, and M Yeung, Resolving Rightful Ownerships with Invisible Watermarking Techniques, *IEEE J. on Selected Areas of Communications*, Dec 1997.
5. I J Cox, et al., A Secure, Robust Watermark for Multimedia, *Proc Information Hiding First International Workshop*, p. 185-206, Cambridge, UK, 1996.
6. Ingrid Daubechies, Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, 41(7):909-996, Oct. 1988.
7. Ingrid Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, 1992.
8. Gerald Kaiser, *A Friendly Guide to Wavelets*, Birkhauser, Boston, 1994.
9. D Kundur and D Hatzinakos, A Robust Digital Image Watermarking Method using Wavelet-Based Fusion, *Proc ICIP*, 1997.
10. Yves Meyer, *Wavelets: Algorithms & Applications*, SIAM, Philadelphia, 1993.
11. F Mintzer, J Lotspiech and N Morimoto, Digital Watermarking, *D-Lib Magazine*, December, 1997.
12. J Z Wang, G Wiederhold, O Firschein and X W Sha, Content-based Image Indexing and Searching Using Daubechies' Wavelets, *Int J Digital Libraries(IJODL)*, 1(4):311-328, Springer-Verlag, 1998.
13. J Z Wang, J Li, G Wiederhold and O Firschein, System for Classifying Objectionable Images, to appear in *Computer Communications J*, Elsevier Science, 1998.
14. Stephen B Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, NJ, 1995.
15. Xiang-Gen Xia, C G Boncelet, G R Arce, A Multiresolution Watermark for Digital Images, *Proc ICIP*, p. 548-51, 1997.
16. Special Issue on Wavelets and Signal Processing, *IEEE Trans. Signal Processing*, Vol.41, Dec. 1993.
17. Mountain Light Photography, Inc. `http://www.mountainlight.com`.
18. Photo Serve, Inc. `http://www.photoserve.com`.
19. The Stock Solution, Inc. `http://www.tssphoto.com`.
20. StirMark. `http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/stirmark/index.html`.
21. unZign. `http://altern.org/watermark/`.