Resolution and Quality Scalable Spread Spectrum Image Watermarking

Angela Piper[*]
School of IT and CS
University of Wollongong
Wollongong, Australia

Reihaneh Safavi-Naini
School of IT and CS
University of Wollongong
Wollongong, Australia

Alfred Mertins
Faculty of Math. and Science
University of Oldenburg
Oldenburg, Germany

ABSTRACT

If digital watermarking is to adequately protect content in systems which provide both resolution and quality scalability, then the watermarking algorithms used must provide both resolution and quality scalability. Although there exists a tradeoff between resolution and quality scalability, we demonstrate that it is possible to achieve both types by taking advantage of human visual system characteristics to increase quality scalability without compromising resolution scalability. To this end, we present a new algorithm for texture detection, which is specifically designed to avoid the false detection of edges as well as smooth regions. Furthermore, we present a case for texture detection using a single resolution only; noting that, in a watermarking context, this method offers advantages over the more popular multi-scale approach.

Categories and Subject Descriptors

I.4.m [**Image Processing and Computer Vision**]: Miscellaneous; E.m [**Data**]: Miscellaneous

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

digital watermarking, scalable, texture analysis

1. INTRODUCTION

Demand for increased flexibility in content delivery has lead to the development of scalable compression algorithms which offer the ability to easily adapt content to suit the capabilities of the user's system. The widespread use of scalable compression for content delivery, however, requires effective means to secure such content. That is, it requires the development of scalable watermarking algorithms.

Watermarking algorithms considering this problem have been proposed, however they tend to focus on a single type of scalability, resolution [16, 24] or quality [6, 21]. Peng et al. [18] consider both types, but their algorithm deals exclusively with authentication and is not a watermarking algorithm. In this paper we focus on providing a spread spectrum watermarking algorithm which has both resolution and quality scalability, demonstrated through experimental testing using the JPEG2000 compression algorithm.

From a spread spectrum based algorithm, one can obtain either resolution *or* quality scalability by using a constant embedding strength. To fully protect content in systems which are both resolution and quality scalable we require a watermarking algorithm with both resolution *and* quality scalability. This goal is beyond reach for a non-adaptive algorithm, because the low-resolution coefficient selection required for resolution scalability and the high embedding strength required for quality scalability must be traded-off to avoid violating the watermark invisibility constraint.

To alleviate this tradeoff, we begin with a non-adaptive resolution scalable algorithm and exploit the contrast sensitivity and texture masking characteristics of the human visual system (HVS) to construct an HVS adaptive algorithm that has good quality scalability whilst retaining the resolution scalability present in the original algoithm. Contrast sensitivity functions, which model the ability of the human visual system to precieve low-contrast patterns, are quite common in the literature and are simply applied to allow the adjustment of watermark embedding strength according to the resolution and orientation of the embedded coefficient. Finding an acceptable algorithm for texture based embedding strength adjustment, however, is not as simple. Thus we present a new algorithm for texture detection in watermarking.

Existing texture based watermarking algorithms increase the embedding strength in edged regions, which are potentially sensitive to modification, as well as in textured regions. Our algorithm is specifically designed to concentrate on textured regions only, avoiding the visible distortions which may occur when strength increases are applied to edges. Furthermore, our texture algorithm is applied in the wavelet domain but uses only a single resolution for each coefficient to be watermarked. This is perhaps somewhat counter-intuitive, as a common reason for using the wavelet domain for texture detection is the ability to analyse the

[*]Partial funding for this research was provided by the Smart Internet Technology Cooperative Research Centre, Australia.

MM-SEC'05, August 1–2, 2005, New York, New York, USA
Copyright 2005 ACM 1-59593-032-9/05/0008 ...$5.00.

texture at many different scales. Accordingly, we also offer a case for the use of single resolution based texture detection, considering both the implementation advantages and the advantages in watermark quality.

This research is on watermark scalability and, as such, the robustness evaluation is focused entirely on compression in the form of resolution and quality reduction and not on other forms of attack. We would expect that the robustness of our watermarking algorithm to manipulations such as cropping and slight rotation would remain similar to that of other non-blind spread spectrum algorithms. Robustness to filtering will depend on the type of filter but as resolution reduction is essentially a form of low pass filtering we would expect good robustness to such attacks; also the strength increases applied to textured regions should provide some protection against high pass filtering.

The determination of whether or not the proposed algorithm provides improved scalability over existing schemes requires the development of measures of watermark scalability. Furthermore, it is desirable that the same measures can be used to examine either resolution or quality scalability. We present two such measures, corresponding to the two properties of a scalable watermark. The first of these evaluates the minimum level of watermark protection provided to content of 'acceptable' quality. The second assesses how well increases in content 'value' are matched with increases in watermark protection. The calculation of these measures for different watermarking algorithms, across the same set of resolution and quality scaled content, enables us to compare resolution and quality scalability using standard statistical tests.

Experimental comparison of the proposed HVS adaptive algorithm with the non-adaptive algorithm on which it is based shows a substantial increase in the detectability of the watermark in quality-scaled content and no significant decrease in the detectability of the watermark in resolution-scaled content, demonstrating that the techniques proposed are effective in alleviating the resolution/quality scalability tradeoff. The HVS adaptive algorithm is also compared with the widely acknowledged algorithm of Cox et al. [7] and the algorithm of Xia et al. [24] which was designed with edge/texture-based HVS adaptation, resolution scalability and integration with compression algorithms in mind. We will show that the Cox algorithm presents good quality scalability at the expense of resolution scalability, the Xia algorithm does the reverse and the proposed algorithm offers a balance between the two.

2. BACKGROUND

2.1 Digital Watermarking

Digital Watermarking is the act of embedding data in digital content. Digital watermarks can be used in a range of applications for purposes such as copyright protection, authentication and labelling.

Most digital watermarking algorithms must strike a balance between the following three desriable but competing properties: capacity, the amount of data which is embedded; robustness, the ability of the embedded data to survive manipulations of the content (be they malicious or innocent) and invisibility[1], the ability of the distortions caused by embedding remain unnoticed by a human observer.

The watermarking algorithms in this paper are considered for copyright protection of scalably compressed image content. As is generally the case with copyright protection watermarks, the capacity requirement is extremely low - a single bit indicating the presence or absence of the copyright holder's watermark. In this case the tradeoff is between invisibility and robustness.

2.2 A Simple Spread Spectrum Watermarking Algorithm

The following outlines embedding and detection components of the basic spread spectrum watermarking algorithm first introduced by Cox et al.

Setup:

Let $W = (w_1, w_2, ..., w_N)$ be an N-element Gaussian watermark sequence with zero mean and unit variance. Let I represent the original image, I' the watermarked image and I^* a potentially distorted copy of the image which may or may not contain the watermark. Let d be a variable indicating the success or failure of watermark detection. Let $s(W_1, W_2)$ be a statistic, such as similarity

$$sim(W_1, W_2) = \frac{W_1.W_2}{\sqrt{W_2.W_2}}$$

or correlation

$$cor(W_1, W_2) = \frac{W_1.W_2}{\sqrt{(W_1.W_1)(W_2.W_2)}}$$

that measures the closeness of the match between two N-element sequences W_1 and W_2.
A threshold T_{fp} is determined according to behaviour of the chosen detection statistic and the maximum acceptable false positive rate for watermark detection.

Embedding:

```
Input: $I, W$
Output: $I'$
o Select coefficients $V = (v_1, v_2, ..., v_N)$ from $I$
o $I' = I$
o Select corresponding coefficients $(v'_1, v'_2, ..., v'_N)$ from $I'$
o $v'_i = v_i(1 + \alpha w_i)$
```

Detection:

```
Input: $I, I*, W$
Output: $d$
o Select coefficients $V = (v_1, v_2, ..., v_N)$ from $I$
  using the same process as for Embedding
o Select corresponding coefficients $(v^*_1, v^*_2, ..., v^*_N)$ from $I^*$
o Extract $W^* = (w^*_1, w^*_2, ..., w^*_N)$ where $w^*_i = \frac{1}{\alpha}(\frac{v_i *}{v_i} - 1)$
o Calculate the detection statistic, $s(W, W^*)$
o If $s(W, W^*) > T_{fp}$, $d =$ Yes else $d =$ No
```

[1] There is a class of watermark known as the visible watermark, often used to add a visible trademark or logo to image or video content, for which the invisibility constraint does not apply, but the watermarks in this paper are not of this class.

2.3 JPEG2000

JPEG2000 is a new wavelet-based image compression standard, which has been developed to provide higher levels of consistency, performance and flexibility than the old DCT-based JPEG standard. An important feature of the standard as it applies to internet and mobile applications is that JPEG2000 offers greatly improved options for scalability.

During the compression process, a discrete wavelet transform is applied using a dyadic decomposition structure. The lowest resolution layer is formed from the LL subband after n decompositions and each subsequent resolution layer contains the additional LH, HL and HH subbands required for image reconstruction at twice the horizontal and vertical resolutions.

A context adaptive bit plane coder is independently applied to spatially contiguous groups of coefficients from the same subband to produce a number of coding passes; each pass generally encodes a single bit each[2] from some of the coefficients in the group. These passes are then arranged into layers so that those passes which provide the greatest quality improvements are in the lowest layer, those which provide slightly smaller improvements appear in the next layer and so on. The number of passes included in each layer is determined according to the target compression rate for the subimage resulting from the addition of that layer.

2.4 Scalable Watermarking

The need for efficient distribution of content to a wide variety of devices, each with different display and processing capabilities and different connection bandwidths, has given rise to scalable compression schemes, which allow the transmission of only the portions of the content that are required for the best performance on the particular device. However, if scalable distribution schemes are employed, then in order to adequately protect the content using watermarking, we require watermarking algorithms which are also scalable.

Scalable Watermarking algorithms consist of combined watermark embedding and detection schemes intended for use with scalable content and possessing the following two properties:

1. The watermark is detectable in any portion of the scaled content which is of ‘acceptable’ quality.
2. Increased portions of the scaled content provide reduced error in watermark detection appropriate to the improved content quality.

The first property ensures that some degree of protection is given whenever the content is still of value. The second property ensures that more valuable content is given a higher degree of protection.

As there are different types of scalability in compression, there are, correspondingly, different types of scalability in watermarking. The two main types of scalability which are of importance to the compression of still images are resolution and quality scalability.

Resolution scalability (or *spatial scalability*) is the ability to easily display visual data at a number of target spatial resolutions. It is achieved by encoding a low resolution version of the image separately from one or more layers of higher resolution refinement data. This data can be combined with the appropriately scaled low resolution image to produce a higher resolution image. Typically each refinement-layer allows the display of an image at twice the horizontal and twice the vertical resolution previously obtainable.

Quality scalability is the ability to easily display visual data at a number of target quality levels. It is achieved by encoding a coarsely quantized version of the image separately from one or more layers of more finely quantized refinement data at the same resolution. The refinement-layers can be combined with the coarsely quantized version of the image to produce a higher quality image. Quality scalability is also termed *SNR scalability*, however the quality metric used to determine the layers need not be directly related to the signal-to-noise ratio (SNR).

If the definition of scalable watermarking holds when a watermarking algorithm is employed with a resolution scalable compression scheme then the watermarking algorithm can be deemed to provide ‘resolution scalability’. Similarly, if the definition holds when the watermarking algorithm is employed with a quality scalable compression scheme it can be deemed to provide ‘quality scalability’. In order to be truly useful for the protection of still image content within a scalable distribution scheme, a watermarking algorithm must provide both types of scalability.

Watermarking algorithms will satisfy the scalability conditions to varying degrees. A watermarking algorithm which well satisfies property 1 of our scalable watermarking definition will produce watermarks which are still detectable in more highly scaled content. We can compare the resolution or quality scalability of watermarking algorithms in terms of property 1 by comparing the detection statistic values at the lowest resolution or lowest quality subimage.

A watermarking algorithm which well satisfies property 2 will result in an increase in watermark detectability when a new portion of content is received which closely matches the increase in content value provided by the inclusion of that portion. We can evaluate scalability according to property 2 by comparing the amount of watermark which is actually present in each layer with the amount of watermark which would ideally be present layer (based on the estimated ‘value’ of the content at that layer).

2.5 Resolution/Quality Scalability Tradeoff

The simple spread spectrum watermarking scheme described in section 2.2 is non-adaptive, in that it uses a constant embedding strength α regardless of the properties of the selected coefficients V. This results in a tradeoff between resolution and quality scalability.

Both *Embedding* and *Detection* require a set of coefficients to be selected from the transformed image. Depending on the selection scheme used we will achieve varying degrees of scalability in the resulting watermarking algorithm.

Examination of alternative coefficient selection schemes [19] highlights a tradeoff between resolution scalability and quality scalability. Selection schemes which are designed for reduced image distortion, such as that in [7], allow a reasonably high embedding strength; the result of which is that much of the watermark is still available at low quality layers, providing good quality scalability. However, such schemes are limited in their selection of low resolution coefficients due to the distortion inherent in such selection.

[2] If the bit encoded for a given coefficient is the most significant bit of that coefficient, an additional bit representing the sign of the coefficient will also be encoded.

The only mechanism for improving the algorithm's resolution scalability is to allow the selection scheme to include a far greater number of coefficients from low resolution layers. Unfortunately, in order to maintain watermark invisibility, this requires a reduction in embedding strength which results in a reduction in quality scalability.

Increasing the embedding strength α for a resolution scalable algorithm should increase it's quality scalability. This cannot be done for a non-adaptive watermarking algorithm, with a constant α, without compromising invisibility constraints. However, replacing the constant α with a variable α_i may allow us to selectively increase the strength for coefficients that will not contribute to the perceived distortion. If the α_i can be sufficiently increased in this manner, for sufficient number of coefficients, then the algorithm will be both resolution *and* quality scalable.

3. MEASURING SCALABILITY

The construction of measures of scalability enables us to experimentally compare the resolution and quality scalability of the proposed algorithm with that of existing algorithms. These measures must adhere closely to the definition of watermark scalability, be easy to calculate and should be applicable to either type of scaled content.

We propose two measures of scalability, corresponding to the to properties 1 and 2 of a scalable watermark. For property 1 we define *detectability*, which evaluates the minimum level of watermark protection provided to content of 'acceptable' quality. For property 2 we define *graceful improvement*, which assesses how well increases in content 'value' are matched with increases in watermark protection.

Each construction considers the case of a scalably compressed, watermarked image composed of n layers (the same construction applies regardless of whether resolution or quality layers are used). From this image are obtained n subimages, where subimage k consists of layers 1 to k. From each subimage, the similarity statistic sim_k is calculated.

3.1 Property 1 - Detectability

An algorithm satisfying property 1 will produce a watermark which is detectable in any version of the scaled content which is of 'acceptable' quality. What constitutes 'acceptable' is highly subjective, so for the purposes of this paper we assume that the subimages produced in our experiment (section 6) are of acceptable quality. Thus, provided the watermark is detectable in the most highly scaled subimage, the lowest layer, it will be detectable in all scaled versions of the content. The higher the detection statistic at the lowest layer, the better property 1 will be satisfied.

Although the false positive and false negative error rates are the ultimate measures of watermark protection, these rates are generally low, thus a great number of trials are required to obtain enough false positives or false negatives for an approximation of the error rate. To reduce the number of trials required we can use the watermark detection statistic; since, for any detection threshold, higher values of the detection statistic result in lower error rates.

We construct a detectability measure $\mathcal{D}$ to evaluate scalability in terms of property 1 using the calculated values of the similarity detection statistic (sim) at the lowest resolution[3] or lowest quality subimage ($k = 1$)

$$\mathcal{D} = sim_1.$$

3.2 Property 2 - Graceful Improvement

An algorithm satisfying property 2 will produce a watermark in which increased portions of the scaled content provide reduced error in watermark detection to appropriate to the improved content quality. This means that the more a particular layer contributes to the overall image content, the more of the watermark should be embedded in that layer. The better the fit between the value of each layer and the amount of watermark contained in that layer, the better property 2 will be satisfied.

We construct a graceful improvement measure $\mathcal{G}$ to evaluate scalability in terms of property 2 by comparing the *ideal* amount of watermark present in a given layer k with the *actual* amount present in the same layer.

The *ideal* is the number I_k of watermark elements, from an N-element watermark, which would ideally be embedded in layer k given it's value. Let P_k denote the perceptual quality of subimage k relative to the original image, measured using PSNR. The content 'value' contributed by layer k can then be described as the improvement in perceptual quality caused by the addition of layer k, or $P_k - P_{k-1}$. The *ideal* number of watermark elements in layer k is then simply the value of layer k proportional to the full image value $P_n - P_0$ (where subimage n is the full watermarked image and subimage 0 is an empty (mid-grey) image), multiplied by the number of elements N in the full watermark.

$$I_k = N \frac{P_k - P_{k-1}}{P_n - P_0}$$

To calculate the *actual* number A_k of watermark elements present in a subimage that has been formed using a resolution decomposition we could simply count the number of watermarked coefficients present in the subimage, as each of these is a complete coefficient and will thus contain an entire watermark element. Attempting to do the same thing for a quality decomposition is problematic, however, since a subimage formed using a quality decomposition may consist of partial coefficients and, as a result, contain partial watermark elements. A number of partial watermark elements are required to provide an equivalent level of protection to one full watermark element, and this number will vary according to how much of each watermark element is present in the subimage.

We can avoid this problem by noting that, when the extracted watermark consists of complete watermark elements, the expected value of the similarity detection statistic is equal to the square root of the number of watermark elements. Thus for a resolution decomposition, the expected number of complete watermark elements present subimage k is sim_k^2; while for a quality decomposition, the partial watermark elements contained in subimage k provide a level of protection equivalent to that of sim_k^2 complete watermark elements. In either case, we can define *actual* as the expected number of complete watermark elements which

[3] To enable fair comparison between algorithms which prohibit embedding in the LL band and those which do not, the resolution subimage which consists of the LL band only is not included in any of the calculations. Thus the 'lowest resolution' subimage consists of the LL, LH, HL and HH subbands after n decompositions.

would have to be embedded in layer k to cause the observed change in the detection statistic from subimage $k-1$ to subimage k; calculated as follows

$$A_k = (sim_k)^2 - (sim_{k-1})^2.$$

How greatly the actual number of watermark elements differs from the ideal number of watermark elements across all layers is measured by

$$\Delta = \sum_k \frac{(A_k - I_k)^2}{I_k}.$$

The use of squared difference between actual and ideal results in a preference for more evently distributed schemes which obtain values close to the ideal for all layers, over those which are very close to ideal on most layers but far from ideal on other layers. Division by the ideal ensures that the differences between actual and ideal are considered in proportion to the layer value. We form a more convenient measure of how closely the actual matches the ideal by normalising so that the possible values range between 0 and 1 so that a value of 1 indicates a perfect fit to the ideal

$$\mathcal{G} = 1 - \frac{\Delta}{N(\frac{N}{I_m} - 1)}$$

where m is the layer for which $0 < I_m \leq I_k \ \forall k$.

4. HVS ADAPTIVE WATERMARKING

To alleviate the tradeoff between resolution and quality scalability that occurs with non-adaptive watermarking, we adapt the embedding strength according to properties of the human visual system and the image content. Studies of the human visual system suggest that the embedding distortion may be masked depending on the contrast sensitivity at the frequency of the embedded coefficient v_i and the textural properties of the image in the same spatial region as the embedded coefficient.

4.1 CSF Based Masking

Contrast sensitivity is the ability to distinguish a low-contrast pattern from an area of uniform colour. The smallest detectable difference in contrast changes depending on the spatial frequency of the pattern being viewed. The modification of a wavelet coefficient during the watermarking process causes a small change in contrast at the particular location and frequency associated with that wavelet coefficient. If the contrast sensitivity at that frequency is low, then the modification is less likely to cause a visible artifact than if the contrast sensitivity at that frequency is high. Thus a good first step towards improving the quality scalability of our watermarking algorithm would be to increase the value of α_i according to the estimated contrast sensitivity at the subband to which v_i belongs.

These increases are easily achieved through the use of an appropriate *Contrast Sensitivity Function* (CSF), which describes the relative contrast sensitivity of the human visual system for sinusoidal patterns of various frequencies, where the frequency f is measured in cycles per degree of visual angle. While there are many versions of the CSF available, the majority have been developed using luminance based experiments only. The versions in [17] have been developed specifically for use with colour image content, for a number of different colour spaces. We adopt[4] those provided for the $\mathrm{YC_bC_r}$ space, commonly used in JPEG2000 compression.

$$\begin{aligned} \mathrm{CSF_Y} &= 0.997 f^2 e^{-0.970 f^{0.758}} + 0.221 e^{-0.800 f^{1.999}} \\ \mathrm{CSF_{Cb}} &= e^{-0.2041 f^{0.900}} \\ \mathrm{CSF_{Cr}} &= e^{-0.1521 f^{0.893}} \end{aligned}$$

The CSF is then sampled at the midpoint of the frequency range covered by each subband in each resolution level to obtain an estimate $\mathrm{CSF}(r,s)$ for the average sensitivity to modifications in resolution r and subband s. The multiplicative inverse of the resulting value, then, describes the relative amount of modification which can be applied to a subband while maintaining the invisibility watermark.

Thus for embedding in a coefficient v_i from component c_i, resolution level r_i and subband s_i we set

$$\alpha_{\mathrm{CSF},i} = \frac{\alpha}{\mathrm{CSF}_{c_i}(r_i, s_i)} .$$

The calculation of frequency range given the resolution of the image requires an estimate of the viewing distance, with further distances moving the region of greatest sensitivity closer to the lowest resolution. The result of this is that if the viewing distance is further than that used in the calculation of f, the actual sensitivity in the low-resolution subbands of the Y component will be greater than the estimated sensitivity, potentially resulting in watermark visibility. To avoid this problem, we follow the method suggested in [17] and estimate a minimum viewing distance and then set $\mathrm{CSF_Y}(r,s) = 1$ for all r below that in which the peak CSF value is found.

4.2 Texture Based Masking

It has been noted [4,14,15] that the human visual system is less sensitive to modifications in textured regions. Regions which attract the label *textured* are those which are substantially composed of high-contrast areas. The texture masking property can be exploited by increasing α_i at coefficients which lie in textured regions.

In order to achieve this it is necessary to at least identify whether or not a given coefficient lies within a textured region. Further, it is desirable to determine the degree to which the region is textured, where regions composed of areas of higher contrast or in which the high-contrast areas are more densely packed are considered more highly textured, in order to provide a greater increase to embedding strength in highly textured regions.

A drawback associated with the use of texture based masking is the problem of edges. Edge regions are also composed of high contrast areas. As a result, texture detection algorithms used for watermarking [2,3,23] often detect edges as well. That is, rather than distinguishing textured regions from edged and smooth regions, they distinguish smooth regions from edged and textured regions. This problem is often not addressed, with [23] citing decreased sensitivity to ‘edged and textured’ regions as a whole.

While there does exist an edge based masking effect, it is highly limited [9]. First, it requires that the site of modification be very near the edge (within 2 or 3 pixels). Second, and more importantly, it requires that the modification and

[4] The function $\mathrm{CSF_Y}$ presented here is a corrected version of the one found in [17] in which the values 0.997 and 0.221 have been switched.

the edge be of the same orientation. The extremely limited orientation sensitivity of the critically sampled wavelet transform employed in JPEG2000 compression means that this second criterion is unlikely to be satisfied. Furthermore, there is the potential for increased sensitivity to modification near edges due to their importance in high-level vision tasks [12,13,22]. In contrast to previous algorithms, our algorithm is designed exclusively for texture detection rather than edge and texture detection. This allows us to embed more strongly in textured regions without risking visibility at edges.

By determining a texture score t_i = Texture(v_i) for the coefficient v_i we can adjust the embedding strength accordingly:

$$\alpha_i = (1 + t_i)\alpha_{\mathrm{CSF},i}$$

To avoid distortion to smooth and edged regions we threshold the texture score, setting $t_i = 0$ whenever Texture(v_i) $<$ $T_{Texture}$ this maintains the embedding strength for those regions at $\alpha_{\mathrm{CSF},i}$.

In an image composed largely of smooth regions, very few coefficients will be identified as textured and receive the associated increase in embedding strength. Thus for such images we will expect the texture based strength adjustment step to make little or no difference to algorithm performance.

4.2.1 Texture Scoring Algorithm

This algorithm provides a texture score t indicating the degree of texture present in the neighbourhood surrounding a given coefficient v at the resolution r to which the coefficient belongs.

Preconditions:
The resolution r must satisfy $r > \lfloor \frac{R}{3} \rfloor$, where R is the number of resolution levels in the decomposed image.

Setup:
Let B_0 be an $n \times n$ block of coefficients centred at v, excluding those coefficients which are not within the subband boundary. The nominal blocksize n should be small enough that any texture detected is reasonably local to the coefficient examined, yet sufficiently large to allow discrimination between textured and edged regions; sizes ranging from 15 to 19 show good performance.

Define blocks B_1 and B_2 similarly for the remaining two subbands at resolution r. Let S_0, S_1 and S_2 be the average coefficient magnitudes of the subbands containing blocks B_0, B_1 and B_2 respectively, and let $|B_i|$ be the number of coefficients in block i [5]. The threshold adjustment parameter q and the weights w_1 and w_2 are constants.

For a blocksize $n = 17$ and threshold adjustment parameter $q = 0.\dot{6}$ we found the weight values $w_1 = 7.7\dot{2}$ and $w_2 = 0.4$ to work well.

Texture:

```
Input: $B_0, B_1, B_2, S_0, S_1, S_2$
Output: $t$
○ For i = 0 to 2
        $e(B_i) = \sum_{b \in B_i} \frac{|b|}{|B_i|}$
        $c(B_i) = \sum_{b \in B_i \wedge |b| > qS_i} \frac{1}{|B_i|}$
○ $E = \sqrt{\max(\ e(B_0)e(B_1),\ e(B_0)e(B_2),\ e(B_1)e(B_2)\ )}$
○ $C = \sqrt{\max(\ c(B_0)c(B_1),\ c(B_0)c(B_2),\ c(B_1)c(B_2)\ )}$
○ $t = w_1 C + w_2 E$
```

The Texture algorithm is designed to produce a texture score t that is large if the region around a coefficient v, at the same resolution, is textured but is small if the same region is smooth or edged. This region is comprised of the 3 blocks of wavelet coefficients which correspond to the spatial neighbourhood of the coefficient v at the same resolution but at 3 different orientations (horizontal, vertical and diagonal). High-magnitude coefficients within the blocks represent areas with high horizontal, vertical or diagonal contrast, so blocks in textured regions can be expected to contain many high-magnitude coefficients.

The energy e of a block is the average magnitude of coefficients within that block. It will be large when the block contains many high-magnitude coefficients (ie. is textured) and small when the block contains no high-magnitude coefficients (ie. is smooth). Edged regions often contain sections with higher contrast than textured regions. Thus, if a block contains a strong edge then e may also be large, even when many of the coefficients contained in the block are of low-magnitude.

The count c of high-magnitude coefficients, proportional to block size, will be high in textured blocks, which contain many high-magnitude coefficients, but will generally be low in edged blocks, which contain relatively few high-magnitude coefficients. We do not use a fixed value above which a coefficient is considered 'high-magnitude' because this will vary according to the image, resolution and subband orientation. Instead, we count all coefficients greater than some fraction q of the average coefficient magnitude for the subband as high-magnitude. The value chosen for q effects the magnitude at which a coefficient is considered to be of high-magnitude for the purposes of detecting texture. It must be small enough that a textured region will contain many coefficients which exceed the threshold but large enough that the threshold is not exceeded by many coefficients in smooth regions. If we assume the wavelet coefficients of each subband to have a laplacian distribution, then with a value of $q = 0.\dot{6}$ approximately 50% of coefficients examined will be considered high-magnitude for the purposes of inclusion in the count. In some cases, this value will be too low, causing large values of c in regions which appear fairly smooth.

To further reduce the chance of large values in edged regions, we note that an edge is generally high contrast in a single orientation only, while a textured region often contains areas of high contrast in several orientations. We define the E and C values using the maximum pairwise multiples of the e and c component values at various orientations, ensuring that the E and C are large only if their component values are large for at least two orientations; taking the square root restores the E and C values to the same

[5] The number of coefficients $|B_i|$ in a given block may be less than the intended size of $n \times n$ if the block is too close to the subband boundary.

range as their components. As the E values will distinguish textured regions from smooth regions but not from edged regions, and the C values will distinguish textured regions from edged regions but not from smooth regions, we use a weighted sum of both C and E to produce a final texture score t, which is capable of distinguishing textured regions from both edged and smooth regions.

5. THE CASE FOR SINGLE RESOLUTION BASED TEXTURE SCORES

There are many texture algorithms [2, 3, 5, 8, 10] which employ a wavelet transform. A major advantage of wavelet-based algorithms in texture-related applications is the multi-scale nature of the wavelet transform, which allows texture features to be computed at a number of different resolutions and thus improves accuracy over single-scale techniques. Given that this is the case, it seems counter-intuitive to offer a wavelet-based texture algorithm which operates on a single resolution only. There are, however, two reasons we may wish to do this. The first is purely the implementation advantage of reduced computational overheads. The second is the watermark quality advantage which results from increasing the embedding strength of coefficients only in those resolutions which are themselves able to mask the modifications.

5.1 Implementation

For maximum usefulness in a scalable watermarking context, the texture estimation algorithm must be able to function as part of a watermarking scheme within a scalable compression system. Given that the texture score produced by the algorithm is used to adjust the watermark strength on the embedding side, the texture score should be obtainable on the detection side. As a result, the texture algorithm should be suitable for use by a device with potentially significant limited capabilities and hence the fewer resolutions required, the better. In a typical six-level decomposition, up to five resolution levels might be involved in calculation of a texture score, reducing this to one resolution level can thus reduce the requirements to one fifth. However, this will be inconsequential if the use of only a single resolution significantly reduces our ability to determine the degree of texture in each region, resulting in inappropriate embedding strengths.

5.2 Quality

While the many wavelet-based texture algorithms make use of the multi-scale nature of the wavelet transform, these algorithms are commonly designed for tasks such as texture classification where using a number of different resolutions allows textures which are similar at some resolutions to be distinguished using other resolutions.

In a watermarking context, however, we are only interested in texture because of its ability to mask the visual distortion introduced by coefficient modification. We simply require a single score which can be used to determine if a coefficient can tolerate a higher embedding strength or if it cannot. By calculating this score from a block in the same resolution as the coefficient we determine the amount of texture which occurs in the same resolution as the noise which will be added by watermarking that coefficient. Including other resolutions risks artificially raising the score as a result of high amounts of texture detected in resolutions which may have little ability to mask the noise being added.

This problem can be seen in figures 1 and 2. In figure 1 a coefficient chosen from a resolution with a high texture score has had it's embedding strength increased by a factor of 20. In figure 2 a coefficient, chosen from the same spatial location[6] but from a resolution with a low texture score, has been modified in the same manner. The modification is clearly visible in figure 2, where the texture score in the same resolution as the modified coefficient was low.

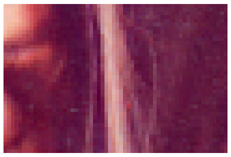

Figure 1: High Texture Score

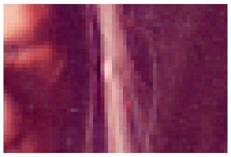

Figure 2: Low Texture Score

Figures 3 and 4 are similar to figures 1 and 2 but with the modified coefficients from a different region of the image. So whereas for the previous pair the high texture score occurred in resolution level 5 and the low texture score in resolution level 4, for this pair the high texture score occurs in resolution level 4 and the low texture score in resolution level 5. That in this pair of images, also, the change to the low texture score coefficient is visible but the change to the high texture score coefficient is not indicates that it is not the resolution level at which the coefficient resides which is important, but the texture score associated with it.

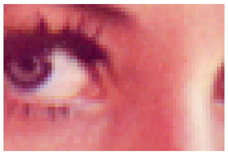

Figure 3: High Texture Score

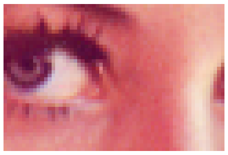

Figure 4: Low Texture Score

Thus we can see that even though texture in a particular resolution, or number of resolutions, may be high, if the texture score in the resolution of the coefficient being modified is low, the texture does not mask the changes. Similarly,

[6] The modified coefficients in Figs. 1 and 2 can be roughly located at 365 257, and those in Figs. 3 and 4 at 283 267, in the 512x512 Lena image.

even though texture scores at other resolutions may be low, if the texture in the resolution of the coefficient being modified is high then the modification is masked. This suggests that basing the texture score on the resolution of the coefficient being modified will provide more appropriate strength adjustment than would a score including additional resolutions.

6. EXPERIMENTAL RESULTS

6.1 Objectives

The primary aim of this experiment is to determine whether the proposed HVS adaptive algorithm demonstrates a significant increase in quality scalability over the non adaptive algorithm, on which it is based, without showing a significant decrease in resolution scalability. As a secondary aim, we wish to examine the scalability of the HVS adaptive algorithm relative to the existing algorithms of Cox et al. [7] and Xia et al. [24]. The first of these has been chosen because it is well known by the watermarking community. The second has been chosen because it, like the proposed algorithm, has been designed to allow detection from reduced resolution and compressed content and has texture-based human visual system adaptation (although, unlike that of the proposed algorithm, this form of texture based HVS adaptation also increases the embedding strength in edge regions).

Both of these objectives may be achieved by comparing the resolution and quality scalability properties of our proposed algorithm with other algorithms. To do this we must examine the scalability performance, as measured by detectability and graceful improvement, of each algorithm on resolution or quality scaled content. This can be done by using each algorithm to watermark a set of images, reducing the resolution or quality of the watermarked images using a scalable compression scheme and calculating the two scalability measures (sections 3.1 and 3.2). Whether there is a significant difference in average performance between the proposed algorithm and other algorithms can then be determined by using a standard statistical test such as the paired t-test. The details of the experiment can be found in the following section.

6.2 Setup

Each of 65 images undergoes JPEG2000 compression using 6 resolution layers, precincts of size 128 × 128, and quality layers with rates 0.01, 0.02, 0.04, 0.06 and 0.9999. As is usual, a transformation from RGB to $\mathrm{YC_bC_r}$ space is applied. The wavelet transformation used is the Daubechies 9,7 filter, provided by the core of the JPEG2000 standard.

Directly preceding the quantization stage, a sequence V of $N = 1000$ wavelet coefficients is selected, and a watermark is embedded in these coefficients. In order to provide consistent grounds for comparison, the embedding strength α is adjusted to ensure the 99th percentile of the S-CIELAB CIEDE 2000 error[7] [20,25] of the watermarked image at full resolution and a compression rate of 0.9999 is $4\Delta E$, which should ensure visual undetectability.

From the watermarked image are produced a series of 5 subimages which comprise the first k resolution or quality layers of the full image. These subimages represent what might be received by various devices with different resolution or bandwidth capabilities. Although there are in fact 6 resolution layers, we do not count the very lowest resolution layer as many schemes specifically prohibit embedding in the LL band and would result in zero similarity for this layer, this gives us 5 subimages for each decomposition type. The watermark is extracted from each subimage and a similarity value calculated as described in section 2.2. To obtain a more accurate estimate of this value we perform the above procedure using 10 different Gaussian watermarks and record the average similarity value sim_k for the k-layer subimage. From the 5 similarity values obtained from each image and decomposition type we calculate a detectability measure $\mathcal{D}$ (section 3.1), for scalability property 1, and a graceful improvement measure $\mathcal{G}$ (section 3.2), for scalability property 2.

The above process is repeated using four different watermarking algorithms: Cox, Xia, `nohvs` and `hvs`. The embedding and detection procedures used for all algorithms are those described in section 2.2; however the algorithms use different embedding strengths and different coefficient selection methods.

- Cox

  The Cox algorithm is a version of the well-known algorithm presented in [7]. Embedding is performed using a fixed embedding strength α. In the experiments in [7], the coefficients V, selected from a discrete cosine transformed image, were the highest magnitude coefficients from the greyscale component, excluding the DC coefficient. The algorithm implemented here uses a version of the same selection scheme suitable for our wavelet transformed image, the coefficients V are the highest magnitude coefficients from the greyscale component, excluding the LL subband.

- Xia

  The Xia is a version of the algorithm presented in [24], which was designed to allow detection from compressed, reduced resolution content. The embedding strength is adapted according to the HVS using the formula $\alpha_i = v_i\alpha$, which increases the strength of the watermark at "edges and textures". In their paper, the coefficients V, selected from a four-level wavelet transformed image, were the highest magnitude coefficients excluding the LL subband. Our implementation uses the same coefficient selection procedure but applied to our six-level wavelet transformed image.

  Also, in the original algorithm of Xia et al., detection of the extracted watermark utilised the corellation statistic cor, rather than the similarity statistic, and watermark presence was determined by computing the correlation between the candidate watermark W and all shifted versions of the extracted watermark W^* in search of a distinct peak in the correlation. To allow comparison, our implementation uses the similarity detection statistic sim and detection threshold T_{fp}.

- `nohvs`

  The `nohvs` algorithm is a non-adaptive, resolution scalable algorithm presented in [19]. Embedding is performed using a fixed embedding strength α. The co-

[7] The settings used for the calculation of the S-CIELAB CIEDE 2000 error were those of a Dell 1702FP (Analog) monitor, 96dpi, viewed at 46cm.

efficients V, selected from a six-level wavelet transformed image, are those with magnitude greater than two fifths the maximum magnitude coefficient at each resolution level (including the LL subband).

- `hvs`
  The `hvs` algorithm is the same as the `nohvs` algorithm except that it employs the HVS-adapted embedding strength $\alpha_i = (1 + t_i)\alpha_{\text{CSF},i}$, described in section 4.2.

For each decomposition type, a paired t-test is used to compare the average detectability and graceful improvment of the proposed `hvs` scheme with each of the other schemes. For each of n images, the difference in detectability, or in graceful improvement, between the `hvs` scheme and another scheme is calculated. The mean m and standard error SE of these differences are used to calculate the test statistic

$$t = \frac{m}{SE}$$

which, provided the differences are normally distributed, follows a student's-t distribution with $\nu = n - 1$ degrees of freedom. This allows us to determine the p-value which is the probability of obtaining the observed mean difference, either positive or negative, should the differences be due to random variation only. The number of images n has been chosen as 65 to allow detection of substantial differences with at least 99% confidence and 90% power. We consider a mean difference of 0.863 for a resolution decomposition and 1.37 for a quality decomposition to represent a substantial change in mean detectability[8].We consider a mean difference of 0.0356 for a resolution decomposition and 0.0342 for a quality decomposition to represent a substantial change in mean graceful improvement[9]. The 65 images used are natural images with horizontal and vertical dimensions ranging from 311 to 768; they have been obtained from the gimp-savvy photo archive [11], an online database of public domain images sourced from the National Aeronautics and Space Administration, the National Oceanic and Atmospheric Administration, and the U.S. Fish and Wildlife Service.

We hope to find an improvement in quality scalability for the `hvs` algorithm when compared to the `nohvs` algorithm, without the presence of a corresponding reduction in resolution scalability. The Cox and Xia algorithms are presented to allow comparison with existing schemes.

6.3 Detectability

Table 1 shows the paired t-test results for the detectability measure. A positive mean difference indicates that the `hvs` scheme shows improved detectability over the given scheme, with larger values indicating a greater level of improvement. Figure 5 shows the mean detectability values for each scheme over all 65 images used.

Table 1: Paired t-test results - Detectability

scheme	decomposition	p-value	mean difference
`nohvs`	resolution	3.6×10^{-11}	-0.00288
Cox	resolution	2.2×10^{-16}	10.4853
Xia	resolution	3.6×10^{-10}	-2.8327
`nohvs`	quality	4.4×10^{-10}	4.7591
Cox	quality	5.9×10^{-15}	-5.0401
Xia	quality	1.5×10^{-6}	3.6001

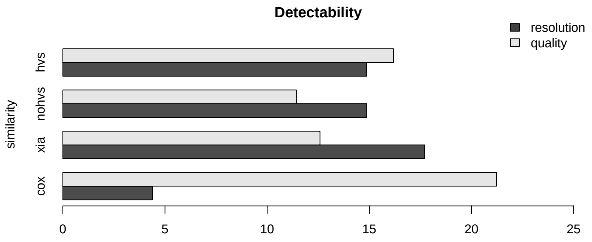

Figure 5: Average Detectability Values

By assuming that the similarity statistic follows a normal distribution [7] we can estimate the false negative rate for each algorithm at the lowest resolution or quality layer. This is done by using the mean and standard deviation of the detectability values for each algorithm to fit a normal distribution and then calculating the proportion of this distribution which lies below the detection threshold. Table 2 shows the estimated false negative rates for each algorithm at the lowest resolution and the lowest quality layer for a threshold value of 6. The Cox algorithm has by far the fewest false negatives for the quality decomposition and the Xia scheme has by far the fewest false negatives for the resolution decomposition. The `hvs` algorithm is the only one for which false negative error rates for both resolution and quality scaled content are below 0.05.

Table 2: Estimated Rate of False Negative Errors

scheme	decomposition	estimated false negative rate
`nohvs`	resolution	0.00033
`hvs`	resolution	0.00033
Cox	resolution	0.74190
Xia	resolution	2.951×10^{-6}
`nohvs`	quality	0.13175
`hvs`	quality	0.020007
Cox	quality	1.3426×10^{-6}
Xia	quality	0.05811

That the tradeoff between quality and resolution detectability has not been entirely eliminated is apparent from the relative performances of the Cox, Xia and `hvs` schemes. There is a substantial increase in resolution detectability relative to the Cox scheme, but a substantial decrease in quality detectability relative to the same scheme; there is a substantial increase in quality detectability relative to the Xia scheme, but a substantial decrease in resolution detectability.

Despite this, the lessening of this tradeoff due to the introduction of human visual system based techniques is clear.

[8] An increase in mean detectability of this magnitude would reduce the false negative detection error for a threshold of 6 by at least 25% for schemes with mean detectability at or above the detection threshold and standard deviations of 2.71 for a resolution decomposition or 4.28 for a quality decomposition.

[9] An increase in graceful improvement of this magnitude is a change of 0.72× the estimated standard deviation of the average scheme.

The `hvs` scheme has substantially improved quality detectability over the `nohvs` scheme, with a mean difference as high as 4.76, and yet the resolution detectability[10] barely suffers at all, with a mean difference of just -0.0029, well below the 0.863 which we take to represent a substantial difference.

6.4 Graceful Improvement

Table 3 shows the paired t-test results for the graceful improvement measure. A positive mean difference indicates that the `hvs` scheme shows increased graceful improvement over the given scheme, with a larger value indicating a greater increase. Figure 6 shows the average detectability values for each scheme over all 65 images used.

Table 3: Paired t-test results - Graceful Improvement

scheme	decomposition	p-value	mean difference
`nohvs`	resolution	0.6596	-0.001769
Cox	resolution	0.2103	0.007108
Xia	resolution	2.2×10^{-16}	0.1041
`nohvs`	quality	0.5118	-0.002666
Cox	quality	9.351×10^{-5}	-0.01725
Xia	quality	2.693×10^{-10}	0.03407

The paired differences obtained using the resolution decomposition do not appear to be normally distributed, thus conclusions drawn from the resolution decomposition rows of table 3 may not be reliable. Thus we also provide (table 4) results for the sign test, which does not rely on the normality of the paired differences, on the graceful improvement measure for the resolution decomposition.

Table 4: Sign test results - Graceful Improvement

scheme	decomp.	p-value	median difference
`nohvs`	resolution	1	3.94×10^{-5}
Cox	resolution	0.002626	0.0169
Xia	resolution	1.163×10^{-16}	0.1088

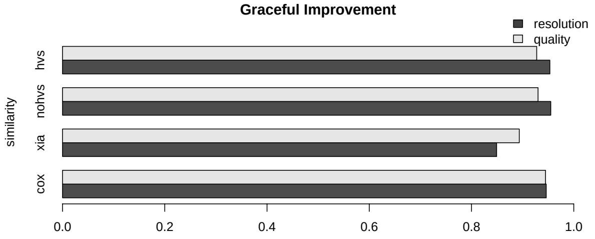

Figure 6: Average Graceful Improvement Values

The graceful improvement comparisons highlight a problem with the Xia scheme. Both resolution and quality comparisons show the Xia scheme to have substantially worse performance than the `hvs` scheme (and, by transitivity, all other schemes). This may be the result of an overemphasis of the watermark in large and low-resolution coefficients and an underemphasis in small and high-resolution coefficients, caused by weighting the embedding strength by the coefficient magnitude.

The differences between the `hvs` and Cox schemes are qualitatively the same as those found in the detectability comparison. The `hvs` scheme outperforms the Cox scheme for a resolution decomposition, while the Cox scheme outperforms the `hvs` scheme for a quality decomposition. However, these differences do not appear to be substantial.

As expected, the use of HVS adaptation did not affect the graceful improvement performance using a resolution decomposition. However, nor did it affect the graceful improvement performance using a quality decomposition. This means that the improvements in quality detectability gained through the use of HVS adaptation have not translated into improvements in quality graceful improvement. This is does not appear to be too great a problem since, as is not the case for detectability, the `nohvs` scheme already shows good graceful improvement for both resolution and quality decompositions.

7. CONCLUSION

While a quality scalable algorithm, such as that proposed by Cox et al., aids image distribution to devices with differing bandwidth capabilities and a resolution scalable algorithm, such as that proposed by Xia et al., aids image distribution to devices with different display resolutions. Only an algorithm which is both resolution and quality scalable is suitable for image distribution across a range of devices differing in both bandwidth and display capabilities.

By taking advantage of human visual system features, we can reduce the tradeoff between resolution and quality scalability, enabling a watermarking algorithm which is both resolution and quality scalable. This is demonstrated through the modification of a wavelet based spread spectrum scheme to employ an HVS-adaptive embedding strength using frequency and texture based masking. While the non-adaptive algorithm has good resolution scalability, it has poor quality scalability. In contrast, the adaptive watermarking algorithm not only maintains the resolution scalability results of the basic algorithm but also demonstrates good quality scalability.

The algorithm which determines the texture component of the variable embedding strength is slightly unconventional in that it uses only a single resolution of the multiresolution wavelet decomposition. We argue that while multiresolution texture analysis may be ideal in contexts such as texture based image segmentation, this is not the case in the context of scalable watermarking. This is, firstly, because a multiresolution approach has increased computational complexity and memory requirments which makes it unsuited to use on target devices with limited memory or processing capabilities and, secondly, because the evaluation of texture at resolutions other than that containing the coefficient undergoing modification may cause an unwarranted increase in embedding strength and compromise the watermark invisibility constraint.

[10] The paired differences for resolution detectability of the `hvs` and `nohvs` schemes are not normally distributed, thus the use of the paired t-test is questionable; however a sign test confirms the result given, with a median difference of -0.0018.

8. REFERENCES

[1] M. Adams and R. K. Ward. Jasper: A portable flexible open-source software tool kit for image coding/processing. In *Proc. of the IEEE ICASSP*, volume 5, pages 241–244, Montreal, May 2004.

[2] M. Barni, F. Bartolini, V. Cappellini, A. Lippi, and A. Piva. A DWT-based technique for spatio-frequency masking of digital signatures. In *SPIE International Conf. on Security and Watermarking of Multimedia Contents*, volume 3657, pages 31–39, San Jose, CA, January 1999.

[3] M. Bertran, J.-F. Delaigle, and B. Macq. Some improvements to HVS models for fingerprinting. In *IEEE International Conf. on Image Processing*, volume 3, pages 1039–1042, Thessaloniki, Greece, October 2001.

[4] A. P. Bradley. A wavelet visible difference predictor. *IEEE Trans. Image Processing*, 8(5):717–730, May 1999.

[5] T. Chang and C. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Trans. Image Processing*, 2(4):429–441, October 1993.

[6] T. P.-C. Chen and T. Chen. Progressive image watermarking. In *Proc. IEEE Intl. Conf. on Multimedia and Expo*, pages 1025–1028, July 2000.

[7] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Processing*, 6(12):1673–1687, 1997.

[8] G. V. de Wouwer, P. Scheunders, and D. V. Dyck. Statistical texture characterization from discrete wavelet representations. *IEEE Trans. Image Processing*, 8(4):592–598, April 1999.

[9] J. Delaigle, C. Devleeschouwer, B. Macq, and I. Lagendijk. Human visual system features enabling watermarking. In *Proc. IEEE International Conference on Multimedia and Expo*, volume 2, pages 26–29, August 2002.

[10] N. Fatemi-Ghomi. *Performance measures for wavelet-based segmentation algorithms*. PhD thesis, University of Surrey, U.K, 1997.

[11] The gimp-savvy photo archive [online]. `http://gimp-savvy.com/PHOTO-ARCHIVE/index.html`. last access: 5 May 2005.

[12] B. Girod. The information theoretical significance of spatial and temporal masking in video signals. In *SPIE International Conf. Human Vision, Visual Processing and Digital Display*, volume 1077, pages 178–187, 1989.

[13] B. T. Hannigan, A. Reed, and B. Bradley. Digital watermarking using improved human visual system model. In *Proc. SPIE Security and Watermarking of Multimedia Contents III*, volume 4134, pages 468–474, August 2001.

[14] N. S. Jayant, J. D. Johnston, and R. J. Safranek. Signal compression based on models of human perception. *Proc. of the IEEE*, 81(10):1385–1422, October 1993.

[15] S. Lewis and G. Knowles. Image compression using 2-D wavelet transform. *IEEE Trans. Image Processing*, 1(2):244–250, April 1992.

[16] S. I. M. Steinder and P. D. Amer. Progressively authenticated transmission. In *MILCOM*, pages 641–645, November 1999.

[17] M. J. Nadenau. *Integration of Human Colour Vision Models into High Quality Image Compression*. PhD thesis, Ecole Polytechnique federale de lausanne, Switzerland, 2000.

[18] C. Peng, R. H. Deng, Y. Wu, and W. Shao. A flexible and scalable authentication scheme for jpeg2000 image codestreams. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 433–441. ACM Press, 2003.

[19] A. Piper, R. Safavi-Naini, and A. Mertins. Coefficient selection methods for scalable spread spectrum watermarking. In *Lecture Notes in Computer Science: Digital Watermarking: Second International Workshop*, pages 235–246, October 2003.

[20] G. Sharma, W. Wu, and E. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *To appear in Color Research and Application*, 30(1), February.

[21] P.-C. Su, H.-J. M. Wang, and C.-C. J. Kuo. An integrated approach to image watermarking and jpeg2000 compression. *Journal of VLSI Signal Processing*, 27:35–53, 2001.

[22] B. Tao and B. Dickinson. Adaptive watermarking in the DCT domain. In *International Conf. on Acoust., Speech, Signal Processing*, volume 4, pages 2985 – 2988, April 1997.

[23] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun. A stochastic approach to content adaptive digital image watermarking. In *Lecture Notes in Computer Science: Third International Workshop on Information Hiding*, volume 1768, pages 211–236, October 1999.

[24] X.-G. Xia, C. G. Boncelet, and G. R. Arce. Wavelet transform based watermark for digital images. *Optics Express*, 3:497–511, December 1998.

[25] X. Zhang and B. Wandell. A spatial extension of CIELAB for digital color image reproduction. In *Proc. Soc. Inform. Display 96 Digest*, pages 731–734, San Diego, 1996.