Hidden Bits: A Survey of Techniques for Digital Watermarking

Chris Shoemaker
Independent Study
EER-290
Prof Rudko
Spring 2002

Table of Contents

Introduction

The earliest forms of information hiding can actually be considered to be highly crude forms of private-key cryptography; the "key" in this case being the knowledge of the method being employed (security through obscurity). Steganography books are filled with examples of such methods used throughout history. Greek messengers had messages tattooed into their shave head, concealing the message when their hair finally grew back. Wax tables were scraped down to bare wood were a message was scratched. Once the tablets were re-waxed, the hidden message was secure [15]. Over time these primitive cryptographic techniques improved, increasing both speed, capacity and security of the transmitted message.

Today, crypto-graphical techniques have reached a level of sophistication such that properly encrypted communications can be assumed secure well beyond the useful life of the information transmitted. In fact, it's projected that the most powerful algorithms using multi kilobit key lengths could not be comprised through brute force, even if all the computing power worldwide for the next 20 years was focused on the attack. Of course the possibility exists that vulnerabilities could be found, or computing power breakthroughs could occur, but for most users in most applications, current cryptographic techniques are generally sufficient.

Why then pursue the field of information hiding? Several good reasons exist, the first being that "security through obscurity" isn't necessarily a bad thing, provided that it isn't the only security mechanism employed. Steganography for instance allows us to hide encrypted messages in mediums less likely to attract attention. A garble of random characters being transmitted between two users may tip off a watchful 3rd party that

sensitive information is being transmitted; whereas baby pictures with some additional noise present may not. The underlying information in the pictures is still encrypted, but attracts far less attention being distributed in the picture then it would otherwise.

This becomes particularly important as the technological disparity between individuals and organizations grows. Governments and businesses typically have access to more powerful systems and better encryption algorithms then individuals. Hence, the chance of individual's messages being broken increases which each passing year. Reducing the number of messages intercepted by the organizations as suspect will certainly help to improve privacy.

Another advantage hinted at by A. Tewfik [16] is that information hiding can fundamentally change the way that we think about information security. Cryptographic techniques generally rely on the metaphor of a piece of information being placed in a secure "box" and locked with a "key". The information itself is not disturbed and anyone with the proper key can gain access. Once the box is open, all of the information security is lost. Compare this to information hiding techniques where the key is embedded into he information itself.

This difference can be better illustrated by current DVD encryption methods. The CSS algorithm takes digitally encoded video and wraps it in an encrypted container. When the DVD player provides the proper key, the video is decrypted and played. Once the video has been decrypted even once, it becomes trivial to trans-code the content and distribute it with no mark of the original author present. Compare this approach to that of an ideal watermark, where despite encryption the watermark remains with the video

despite various alteration and trans-coding attempts. With this the need for a combination of the two approaches becomes clear.

This paper will begin with a quick background on cryptography and steganography, which form the basis for a large number of digital watermarking concepts. The paper will then move on to a discussion of what requirements a watermarking system must meet, as well as methods for evaluating the strengths of various algorithms. The remainder of the paper will focus on various watermarking techniques and the strengths and weaknesses of each. This paper will focus almost exclusively on the watermarking of digital images, however most of these same ideas could easily be applied to the watermarking of digital video and audio.

Background

First we start with a few definitions. *Cryptography* can be defined as the processing of information into an unintelligible (encrypted) form for the purposes of secure transmission. Through the use of a "key" the receiver can decode the encrypted message (decrypting) to retrieve the original message.

Stenography improves on this by hiding the fact that a communication even occurred. The message *m* is imbedded into a harmless message *c* which is defined as the *cover-obect*. The message *m* is then embedded into *c*, generally with use of a key *k* that is defined as the *stego-key*. The resulting message is then embedded into the cover-object *c*, which results in *stego-object* s. Ideally the stego-object is indistinguishable from the original message c, appearing as if no other information has been encoded [7]. This can all be seen below in figure 1.

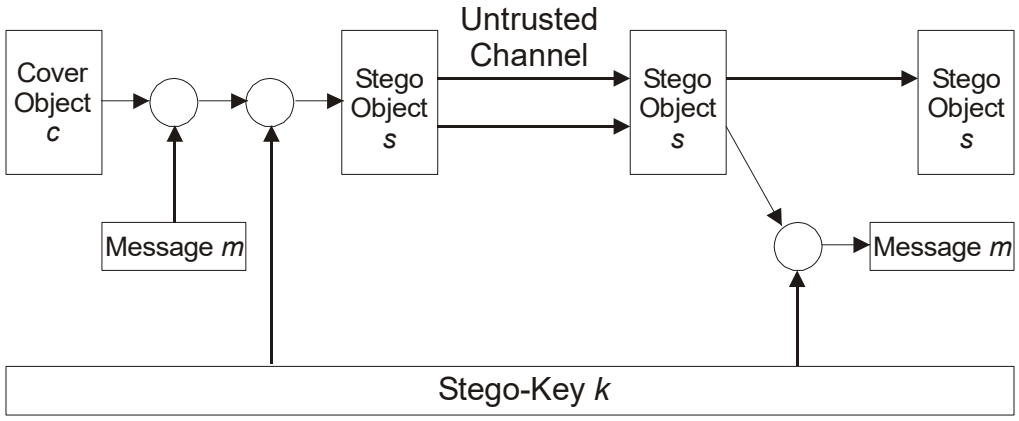

Figure 1- Illustration of a Stegographic System

The cover object is only used for the stego-object generation and is then discarded. The hope of the system is that the stego-object will be close enough in appearance and statistics to the original such that the presence of information will go

undetected. As mentioned previously, for the purposes of this report we will assume the stego-object to be a digital image, keeping in mind that concepts may be extended to other cover objects as well.

Watermarking is very similar to steganography in a number of respects. Both seek to embed information inside a cover message with little to no degradation of the cover-object. Watermarking however adds the additional requirement of robustness. An ideal steganographic system would embed a large amount of information, perfectly securely with no visible degradation to the cover object. An ideal watermarking system however would embed an amount of information that could not be removed or altered without making the cover object entirely unusable. As a side effect of these different requirements, a watermarking system will often trade capacity and perhaps even some security for additional robustness.

What requirements then might an ideal watermarking system have? The first requirement would clearly be that of **perceptibility** [8]. A watermarking system is of no use to anyone if it distorts the cover image to the point of being useless, or even highly distracting. Ideally the watermarked imaged should look indistinguishable from the original even on the highest quality equipment.

The ideal watermark must also be highly **robust**, entirely resistant to distortion introduced during either normal use (unintentional attack), or a deliberate attempt to disable or remove the watermark present (intentional, or malicious attack). Unintentional attacks involve transforms that are commonly applied to images during normal use, such as cropping, resizing, contrast enhancement…etc.

A particularly interesting form of unintentional attack is that of image *compression*. Meerwald [13] points out that lossy compression and watermarking are inherently at odds; watermarking seeks to encode information in extra bits that compression hopes to remove. Thus, ideal watermarking and compression systems are most likely inherently exclusive.

In *malicious* attacks, an attacker deliberately tries to disable the watermark, often through a geometric distortion or the addition of noise. A final note is that robustness can include either resilience to attack, or complete fragility. It may be the case that some watermarking systems may require the watermark to totally destroy the cover object if any tapering is present [2].

Another property of an ideal watermarking system is that it implement the use of **keys** to ensure that the approach is not rendered useless the moment that the algorithm becomes known [8]. It may also be a goal that the system utilizes an asymmetric key system such as in public / private key cryptographic systems. Although private key systems are fairly easy to implement in watermarking, asymmetric key pairs are generally not. The risk here is that embedded watermarking systems might have their private key discovered, ruining security of the entire system. This was exactly the case when a single DVD decoder implementation left it's secret key unencrypted, breaching the entire DVD copy protection mechanism.

Slightly less important requirements of an ideal watermarking system might be **capacity,** and **speed**. A watermarking system must allow for a useful amount of information to be embedded into the image. This can range from a single bit all the way up to multiple paragraphs of text. Furthermore, in watermarking systems destined for

embedded applications, the watermark detection (or embedding) may not be overly computationally intensive as to preclude its use on low cost micro-controllers.

The last possible requirement of an ideal watermarking system is that of **statistical imperceptibility** [14]. The watermarking algorithm must modify the bits of the cover in such a way that the statistics of the image are not modified in any telltale fashion that may betray the presence of a watermark. This requirement is not quite as important here as it is in steganography, but some applications may require it.

How then do we provide metrics for the evaluation of watermarking techniques? Capacity and speed can be easily evaluated using the number of bits per cover size, and computational complexity, respectfully. The systems use of keys is more or less by definition, and the statistical imperceptibility by correlation between the original images and watermarked counterpart.

The more difficult task is providing metrics for perceptibility and robustness. Petitcolas as well as others suggest the scheme listed below in table 1 for the evaluation of perceptibility [14].

Level of Assurance	Criteria
Low	- Peak Signal-to-Noise Ratio (PSNR) - Slightly perceptible but not annoying
Moderate	- Metric Based on perceptual model - Not perceptible using mass market equipment
Moderate High	- Not perceptible in comparison with original under studio conditions
High	- Survives evaluation by large panel of persons under the strictest of conditions.

Table 1 - Summery of Possible Perceptibility Assurance Levels [14]

You may note above that the only rigorously defined metric above is PSNR, shown below in figure 2. The main reason for this is that no good rigorously defined metrics have been proposed that take the effect of the Human Visual System (HVS) into account. PSNR is provided only to give us a rough approximation of the quality of the watermark. Further levels of evaluation rely strictly on observation under varied conditions, as shown in table 1.

$$PSNR = \frac{XY \max_{x,y} p_{x,y}^2}{\sum_{x,y} \left(P_{x,y} - \overline{P}_{x,y} \right)^2}$$

Figure 2 - Determination of Peak Signal-To-Noise Ratio [8]

Petitcolas also provides us with a rough set of reliability or robustness metrics, shown below in table 2.

	Level Zero	Low Level	Moderate
Standard JPEG Compression Quality	100 - 90	100 - 75	100 - 50
Color Reduction (GIF)	256	256	16
Cropping	100 - 90%	100 - 75%	100 - 50%
Gamma Correction		0.7-1.2	0.5-1.5
Scaling		$\frac{1}{2} - \frac{3}{2}$	$\frac{1}{3} - 2$
Rotation		+/- 0 - 2 deg.	+/- 0 - 5 deg. 90 deg.
Horizontal Flip		Yes	Yes
Uniform Noise		1-5%	1-15%
Contrast		+/- 0 - 10%	+/- 0 – 25%
Brightness		+/- 0 – 10%	+/- 0 – 25%
Median Filter			3 x 3

Table 2 - Basic Robustness Requirements [14]

Low level is the bare minimum requirements that a watermark must meet in order to be considered useful. Watermarks at this level should be resistant to common modifications

that non-malicious users with inexpensive tools might do to images. As the robustness increases more specialized and expensive tools become required, as well as more intimate knowledge of the watermarking system being used. At the very top of the scale is *provable* reliability in which it is either computationally or mathematically impossible to remove or disable the mark [14].

This section has introduced the background information, requirements and evaluation techniques required for the implementation and evaluation of watermarking techniques. In the next section various watermarking systems will be discussed and their possible strengths and weaknesses considered.

Choice of Watermark-Object

The first question we need to ask with any watermarking or stenographic system, is what form will the embedded message take? The most straight-forward approach would be to embed text strings into an image, allowing an image to directly carry information such as author, title, date…and so forth. The drawback however to this approach is that ASCII text in a way can be considered to be a form of LZW compression, which each letter being represented with a certain pattern of bits. By compressing the watermark-object before insertion, robustness suffers.

Due to the nature of ASCII codes, a single bit error due to an attack can entirely change the meaning of that character, and thus the message. It would be quite easy for even a simple task such as JPEG compression to reduce a copyright string to a random collection of characters. Rather then characters, why not embed the information in an already highly redundant form, such as a raster image? Not only do images lend themselves to image watermarking applications, but the properties of the HVS can easily be exploited in recognition of a degraded watermark. Consider Figure 3 below:

Figure 3 - Ideal Watermark-Object vs. Object with 25% Additive Gaussian Noise

Note that despite the high number of errors made in watermark detection, the retrieved watermark is still highly recognizable.

Least Significant Bit Modification

The most straight-forward method of watermark embedding, would be to embed the watermark into the least-significant-bits of the cover object [6]. Given the extraordinarily high channel capacity of using the entire cover for transmission in this method, a smaller object may be embedded multiple times. Even if most of these are lost due to attacks, a single surviving watermark would be considered a success.

LSB substitution however despite its simplicity brings a host of drawbacks. Although it may survive transformations such as cropping, any addition of noise or lossy compression is likely to defeat the watermark. An even better attack would be to simply set the LSB bits of each pixel to one…fully defeating the watermark with negligible impact on the cover object. Furthermore, once the algorithm is discovered, the embedded watermark could be easily modified by an intermediate party.

An improvement on basic LSB substitution would be to use a pseudo-random number generator to determine the pixels to be used for embedding based on a given "seed" or key [6]. Security of the watermark would be improved as the watermark could no longer be easily viewed by intermediate parties. The algorithm however would still be vulnerable to replacing the LSB's with a constant. Even in locations that were not used for watermarking bits, the impact of the substitution on the cover image would be negligible. LSB modification proves to be a simple and fairly powerful tool for stenography, however lacks the basic robustness that watermarking applications require.

Correlation-Based Techniques

Another technique for watermark embedding is to exploit the correlation properties of additive pseudo-random noise patterns as applied to an image [9]. A pseudo-random noise (PN) pattern W(x,y) is added to the cover image I(x,y), according to the equation shown below in figure 4.

$$I_w(x, y) = I(x, y) + k * W(x, y)$$

Figure 4 - Addition of Pseudo-Random Noise to Cover Image

In figure 4, *k* denotes a gain factor, and I_W the resulting watermarked image. Increasing *k* increases the robustness of the watermark at the expense of the quality of the watermarked image.

To retrieve the watermark, the same pseudo-random noise generator algorithm is seeded with the same key, and the correlation between the noise pattern and possibly watermarked image computed. If the correlation exceeds a certain threshold T, the watermark is detected, and a single bit is set. This method can easily be extended to a multiple-bit watermark by dividing the image up into blocks, and performing the above procedure independently on each block.

This basic algorithm can be improved in a number of ways. First, the notion of a threshold being used for determining a logical “1” or “0” can be eliminated by using two separate pseudo-random noise patterns. One pattern is designated a logical “1” and the other a “0”. The above procedure is then performed once for each pattern, and the pattern with the higher resulting correlation is used. This increases the probability of a correct detection, even after the image has been subject to attack [9].

We can further improve the method by pre-filtering the image before applying the watermark. If we can reduce the correlation between the cover image and the PN sequence, we can increase the immunity of the watermark to additional noise. By applying the edge enhancement filter shown below in figure 5, the robustness of the watermark can be improved with no loss of capacity and very little reduction of image quality [9].

$$F_{edge} = \frac{1}{2}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure 5 - FIR Edge Enhancement Pre-Filter [9]

Rather then determining the values of the watermark from "blocks" in the spatial domain, we can employ CDMA spread-spectrum techniques to scatter each of the bits randomly throughout the cover image, increasing capacity and improving resistance to cropping. The watermark is first formatted as a long string rather then a 2D image. For each value of the watermark, a PN sequence is generated using an independent seed. These seeds could either be stored, or themselves generated through PN methods. The summation of all of these PN sequences represents the watermark, which is then scaled and added to the cover image [9].

To detect the watermark, each seed is used to generate its PN sequence, which is then correlated with the entire image. If the correlation is high, that bit in the watermark is set to "1", otherwise a "0". The process is then repeated for all the values of the watermark. CDMA improves on the robustness of the watermark significantly, but requires several orders more of calculation.

Frequency Domain Techniques

An advantage of the spatial techniques discussed above is that they can be easily applied to any image, regardless of subsequent processing (whether they survive this processing however is a different matter entirely). A possible disadvantage of spatial techniques is they do not allow for the exploitation of this subsequent processing in order to increase the robustness of the watermark.

In addition to this, adaptive watermarking techniques are a bit more difficult in the spatial domain. Both the robustness and quality of the watermark could be improved if the properties of the cover image could similarly be exploited. For instance, it is generally preferable to hide watermarking information in noisy regions and edges of images, rather then in smoother regions. The benefit is two-fold; Degradation in smoother regions of an image is more noticeable to the HVS, and becomes a prime target for lossy compression schemes.

Taking these aspects into consideration, working in a frequency domain of some sort becomes very attractive. The classic and still most popular domain for image processing is that of the Discrete-Cosine-Transform, or DCT. The DCT allows an image to be broken up into different frequency bands, making it much easier to embed watermarking information into the middle frequency bands of an image. The middle frequency bands are chosen such that they have minimize they avoid the most visual important parts of the image (low frequencies) without over-exposing themselves to removal through compression and noise attacks (high frequencies) [9].

One such technique utilizes the comparison of middle-band DCT coefficients to encode a single bit into a DCT block. To begin, we define the middle-band frequencies ($F_{M)}$ of an 8x8 DCT block as shown below in figure 6.

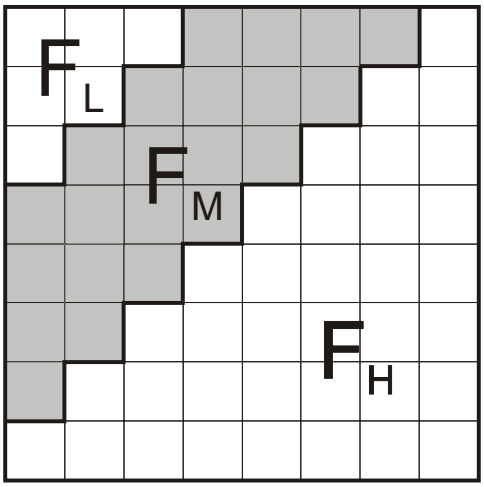

Figure 6 - Definition of DCT Regions [9]

F_L is used to denote the lowest frequency components of the block, while F_H is used to denote the higher frequency components. F_M is chosen as the embedding region as to provide additional resistance to lossy compression techniques, while avoiding significant modification of the cover image [5].

Next two locations $B_i(u_1,v_1)$ and $B_i(u_2,v_{2)}$ are chosen from the F_M region for comparison. Rather then arbitrarily choosing these locations, extra robustness to compression can be achieved if we base the choice of coefficients on the recommended JPEG quantization table shown below in table 2. If two locations are chosen such that they have identical quantization values, we can feel confident that any scaling of one coefficient will scale the other by the same factor…preserving their relative size.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 2 - Quantization values used in JPEG compression scheme [6]

Based on the table, we can observe that coefficients (4,1) and (3,2) or (1,2) and (3,0) would make suitable candidates for comparison, as their quantization values are equal. The DCT block will encode a "1" if $B_i(u_1,v_1) > B_i(u_2,v_2)$; otherwise it will encode a "0". The coefficients are then swapped if the relative size of each coefficient does not agree with the bit that is to be encoded [6].

The swapping of such coefficients should not alter the watermarked image significantly, as it is generally believed that DCT coefficients of middle frequencies have similar magnitudes. The robustness of the watermark can be improved by introducing a watermark "strength" constant *k*, such that $B_i(u_1,v_1) - B_i(u_2,v_2) > k$. Coefficients that do not meet this criteria are modified though the use of random noise as to then satisfy the relation. Increasing *k* thus reduces the chance of detection errors at the expense of additional image degradation [6].

Another possible technique is to embed a PN sequence W into the middle frequencies of the DCT block. We can modulate a given DCT block x,y using the equation shown below in figure 7.

$$I_{W\,x,y}(u,v) = \begin{cases} I_{x,y}(u,v) + k * W_{x,y}(u,v), & u,v \in F_M \\ Ix,y(u,v), & u,v \notin F_M \end{cases}$$

Figure 7 - Embedding of CDMA watermark into DCT middle frequencies [9]

For each 8x8 block x,y of the image, the DCT for the block is first calculated. In that block, the middle frequency components F_M are added to the pn sequence W, multiplied by a gain factor k. Coefficients in the low and middle frequencies are copied over to the transformed image unaffected. Each block is then inverse-transformed to give us our final watermarked image I_W [9].

The watermarking procedure can be made somewhat more adaptive by slightly altering the embedding process to the method shown below in figure 8.

$$I_{W\,x,y}(u,v) = \begin{cases} I_{x,y}(u,v) * (1 + k * W_{x,y}(u,v)), & u,v \in F_M \\ Ix,y(u,v), & u,v \notin F_M \end{cases}$$

Figure 8 - Image dependant DCT CDMA watermark [9]

This slight modification scales the strength of the watermarking based on the size of the particular coefficients being used. Larger *k*'s can thus be used for coefficients of higher magnitude…in effect strengthening the watermark in regions that can afford it; weakening it in those that cannot [9].

For detection, the image is broken up into those same 8x8 blocks, and a DCT performed. The same PN sequence is then compared to the middle frequency values of the transformed block. If the correlation between the sequences exceeds some threshold T, a “1” is detected for that block; otherwise a “0” is detected. Again k denotes the strength of the watermarking, where increasing k increases the robustness of the watermark at the expense of quality [9].

Wavelet Watermarking Techniques

Another possible domain for watermark embedding is that of the wavelet domain. The DWT (Discrete Wavelet Transform) separates an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components. The process can then be repeated to computes multiple "scale" wavelet decomposition, as in the 2 scale wavelet transform shown below in figure 9.

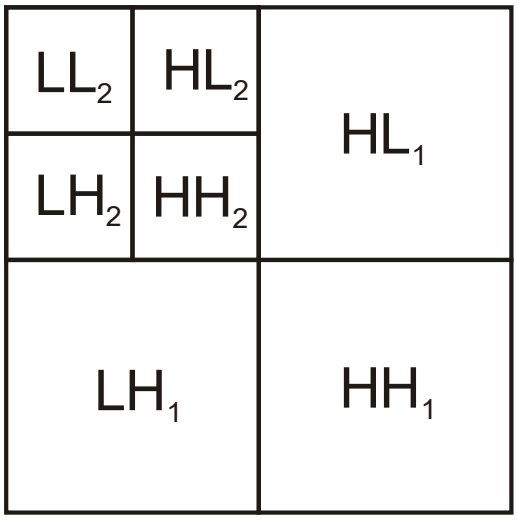

Figure 9 - 2 Scale 2-Dimensional Discrete Wavelet Transform

One of the many advantages over the wavelet transform is that that it is believed to more accurately model aspects of the HVS as compared to the FFT or DCT. This allows us to use higher energy watermarks in regions that the HVS is known to be less sensitive to, such as the high resolution detail bands {LH,HL,HH). Embedding watermarks in these regions allow us to increase the robustness of our watermark, at little to no additional impact on image quality [9].

One of the most straightforward techniques is to use a similar embedding technique to that used in the DCT, the embedding of a CDMA sequence in the detail bands according to the equation shown below in figure 10.

$$I_{W\,u,v} = \begin{cases} W_i + \alpha |W_i| x_i, & u,v \in HL, LH \\ W_i & u,v \in LL, HH \end{cases}$$

Figure 10 - Embedding of a CDMA Watermark In the Wavelet Domain

where w_i denotes the coefficient of the transformed image, x_i the bit of the watermark to be embedded, and α a scaling factor. To detect the watermark we generate the same pseudo-random sequence used in CDMA generation and determine its correlation with the two transformed detail bands. If the correlation exceeds some threshold T, the watermark is detected.

This can be easily extended to multiple bit messages by embedding multiple watermarks into the image. As in the spatial version, a separate seed is used for each PN sequence, which are then added to the detail coefficients as per figure 10. During detection, if the correlation exceeds T for a particular sequence a "1" is recovered; otherwise a zero. The recovery process then iterates through the entire PN sequence until all the bits of the watermark have been recovered.

Furthermore, as the embedding uses the values of the transformed value in embedded, the embedding process should be rather adaptive; storing the majority of the watermark in the larger coefficients. The author [13] claims that the technique should prove resistant to JPEG compression, cropping, and other typical attacks.

Results

We'll begin with a few notes on the results to follow. First, robustness evaluations were limited to testing against JPEG compression and the addition of random noise. Evaluating each of the algorithms against all attacks across a full range of gain values is well beyond the scope of this report. The other robustness metrics described in table 2 will only be touched on briefly, should the algorithm prove exceptionally resistant or exceptionally vulnerable to the attack.

The PSNR of each watermarked image will be given below each figure, however these figures are only to be taken lightly. PSNR does not take aspects of the HVS into effect so images with higher PSNR's may not necessarily look better then those with a low PSNR. This will prove particularly true in the case of the DCT and DWT domain techniques.

Performance requirements shown in table 3 are similarly only to be used as a rough guideline. In general, algorithms were implemented in the most straightforward way, not the most computationally optimal. Furthermore, MATLAB may handle certain programming constructs differently from other languages, thus the best performing algorithm may vary for each language and implementation.

Lastly, in cases where the watermarking algorithm was altered significantly from that described in section III, the modifications will be quickly explains and results from both pre and post modification presented. Smaller implementation alterations however will not be covered.

Three different watermarks were used, based on the theoretical and experimental information capacity of the watermarking algorithm, as shown in figures 11 and 12.

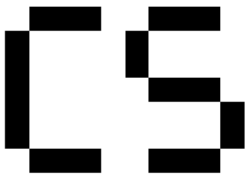

Figure 11 - Small Watermark (12 x 9 pixels)

Figure 12 - Normal Watermark (50 x 20 pixels)

Not shown above is the large watermark created for the LSB embedding algorithm, which uses the normal watermark and titles it out to full image size. For our reference image, the ever-popular miss November (Lena) image is used, as shown below in figure 13.

Figure 13 - Lena Reference Image (512 x 512 Pixels)

Results from LSB substitution were as expected. The watermarked image shows little not noticeable degradation, while the large watermark was recovered perfectly.

Least Significant Bit Substitution

Figure 14a - Watermarked Image PSNR= 102 dB

Figure 14b - Recovered Watermark

Although the watermark was recovered perfectly in the ideal case, the addition of any amount of noise, or compression of the image using JPEG fully destroys the embedded watermark, leaving nothing but noise. Even worse, the watermark can be removed with

Least Significant Bit Substitution

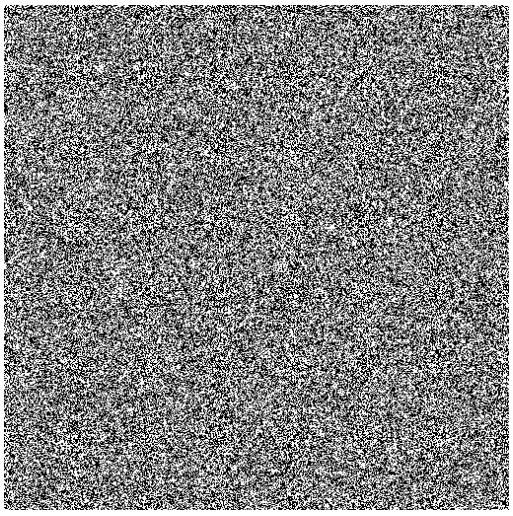

Figure 15a – Recovered Watermark after addition of 1% Gaussian Noise

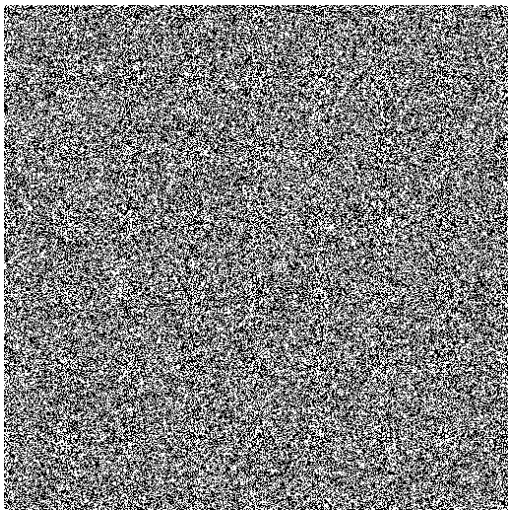

Figure 15b - Recovered Watermark after JPEG Compression with Quality 95

no perceivable change to the watermarked image. The message capacity of LSB embedding however is quite good, a 1:1 correlation with the size of the image.

The results of threshold-based correlation showed a vast improvement over LSB substitution in terms of robustness. Several parameters however must be discussed before we move on to results of this technique. A gain factor of $k = 5$ was chosen experimentally, however larger factors might be used for increased robustness at the expense of visual quality.

Another issue with threshold-based techniques is the choosing of a suitable threshold for detection. One method is to store the correlation of each PN sequence and then use the mean of all the correlations as the threshold T. For watermarks with relatively equal numbers of zero's and ones, this technique should prove somewhat adaptive to a range of image types, as well as varying levels of noise.

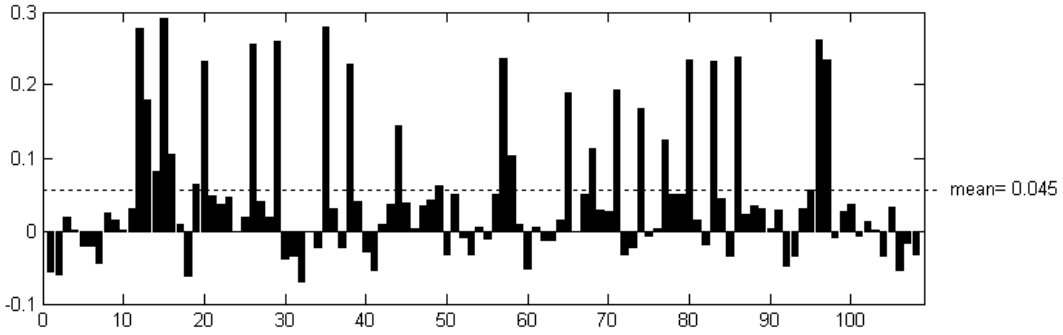

Figure 16 - Choice of Threshold by Mean Value

A final consideration is the size of the watermark being embedded. Use of a smaller watermark will allow larger blocks to be used, increasing the strength of correlation and thus system robustness. Using the normal sized watermark, the largest possible block size $\{8,16,32\ldots\}$ is determined by: $1000 \le \frac{512 * 512}{16^2}$, for a maximum block size of 16.

Threshold-Based Correlation
K = 5 blocksize=16

Figure 17a - Watermarked Image
PSNR = 94.0 dB

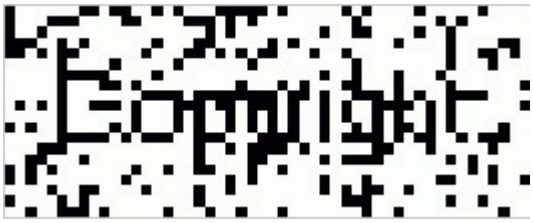

Figure 17b - Recovered Watermark

Although with a gain of 5 the watermark can still be recognized, the results are not spectacular. Increasing the gain does improve watermark recovery, however beyond a k of 5, the blocky regions of noise become visible in the watermarked image, as shown in figure 18. Also note the severe drop in PSNR between the two watermarked images.

Threshold-Based Correlation
K = 40 Blocksize=16

Figure 18a – Heavily Watermarked Image
PSNR = 57.9 dB

Figure 18b - Recovered Watermark

Although the watermark was not perfectly recovered, threshold-based correlation fares much better then LSB in the presence of noise and compression. Using a gain of 5, the watermark is still slightly distinguishable after light levels of noise and compression. As expected, increasing the gain to 40, improves the watermark's robustness significantly.

Threshold-Based Correlation Robustness

Blocksize=16

Figure 19a - 5% Gaussian Noise (k = 5)

Figure 19c - JPEG Compression Q=75 (k = 5)

Figure 19b - 5% Gaussian Noise (k = 40)

Figure 19d - JPEG Compression Q=75 (k = 40)

It's worth noting that watermark recovery could be improved by using a smaller watermark and increasing the block size used for embedding. This should reduce the number of errors from normal detection, as well as improving watermark robustness.

One possible improvement to the threshold-based correlation technique discussed above, is to use two separate PN sequences for embedding; one to encode a “1” and another the “0”. This approach has the advantage of not requiring a “blind” choice of threshold, as the pattern with the higher correlation is chosen. Furthermore, by careful choice of these two patterns to be as un-correlated as possible, we can reduce the change of false detection significantly.

A more subtle advantage is that the approach makes better use of the HVS in spreading its noise throughout the image. The eye is more sensitive to abrupt changes in quality, hence blocky regions of noise will tend to disturb viewers more then a constant level of noise would. Although the PSNR in figure 20 has decreased by a factor of 10, the image remains nearly identical to the reference.

Comparison-Based Correlation

K = 5 blocksize=16

Figure 20a - Watermarked Image
PSNR = 72.5 dB

Figure 20b - Recovered Watermark

Also note that the comparison-based watermark with gain 5 performed marginally better then even the threshold-based with gain 40; with less impact on the cover image.

Robustness is improved as well in the comparison-based watermark. Again we see the comparison-based approach at a gain of 5 rivaling or even besting threshold-based with gain 40.

Comparison-Based Correlation Robustness

Blocksize=16

Figure 21a - 5% Gaussian Noise (k = 5)

Figure 21b - JPEG Compression Q=75 (k = 5)

A disadvantage of these block-based techniques in relation to LSB embedding is that they are highly fragile to flips, crops and rotations. These transformations alter the coordinate systems of the image, making the task of matching up blocks in embedding and recovery quite difficult. The technique however should prove fairly resistant to contrast, brightness and any other sort of per-pixel transform.

Early experimentation with CDMA demonstrated exceptional robustness with relation to noise and high-level JPEG compression, with flawless recovery of the embedded watermark from the pristine image. CDMA in the spatial domain however suffers from several problems that limit its usefulness.

The main drawback of CDMA is that its message capacity is more limited then similar correlation-based techniques. One reason for this is that watermark recovery drops off quickly at higher message sizes. Good results were obtainable using the small watermark, however results with the normal-sized watermark were disappointing. Also, processing time for spatial-domain CDMA watermarking increases exponentially with increasing message sizes. CDMA for the normal message size took by far the longest processing time out of any of the techniques tested (see table 3).

That being said, CDMA performed wonderfully using the smaller message. Through experimentation, the gain factor k=2 was arrived at as a good balance between visual quality and watermark robustness.

CDMA Spread-Spectrum
K = 2

Figure 22a - Watermarked Image
PSNR = 62.3 dB

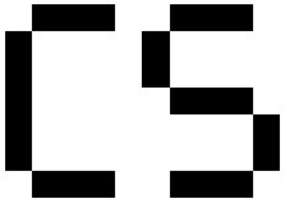

Figure 22b - Recovered Watermark

CDMA Spread-Spectrum Robustness

K=2

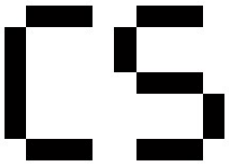

Figure 23a - 15% Gaussian Noise

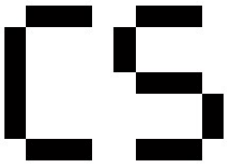

Figure 23c – JPEG Compression Q=75

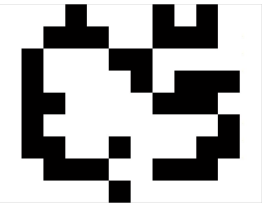

Figure 23b - 50% Gaussian Noise

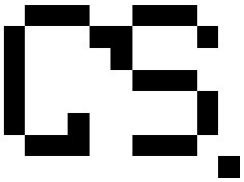

Figure 23d – JPEG Compression Q=50

Based on the results of figure 23a and 23b, we can conclude that CDMA in the spatial domain easily meets the requirements for "moderate" robustness, provided that the encoding messages are relatively small. The results are particularly impressive when you consider that the watermarked image used for figure 23b was entirely unrecognizable after the addition of 50% gaussian noise. Furthermore, CDMA should in theory be resistant to small amounts of cropping and limited degrees of rotation.

The main limitations of CDMA in the spatial domain however remain it's limited capacity and high processing requirements. The embedding of large watermarks using CDMA requires the embedding gain k to be lowered to preserve the visual quality of the image. As more PN sequences are added to the cover image however, larger gains are required to preserve correlation between like sequences. This underlying conflict is the reason that CDMA in the spatial domain will remain more limited in capacity then other techniques.

Comparison of mid-band DCT Coefficients

K = 50

Figure 24a - Watermarked Image
PSNR = 67.6 dB

Figure 24b - Recovered Watermark

Results of the comparison of mid-band DCT coefficients were encouraging. Note that K in this case is not a gain factor such as correlation-based techniques, but rather a threshold. When the two differences in magnitude between the two coefficients being compared does not exceed K, they coefficients are scaled such that they meet this requirement. As in previous techniques, k was chosen experimentally however larger k's may be used for increased robustness at the expense of quality.

In figure 24 we can observe that the technique works perfectly for un-altered images, with good visual quality of the watermarked image. The blocksize for each of the DCT-based techniques was kept constant at 8 x 8, in anticipation of JPEG compression. Better results could be obtained using larger block sizes at the expense of message capacity.

Comparison of mid-band DCT Coefficients Robustness

K=50

Figure 25a - 5% Gaussian Noise

Figure 25b - 15% Gaussian Noise

Figure 25c – JPEG Compression Q=50

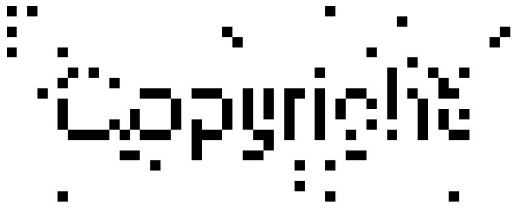

Figure 25d – JPEG Compression Q=20

The comparison DCT-coefficients proved to be both moderately robust against gaussian noise, and extremely robust against JPEG compression. Good recovery results were still possible with a watermarked image that had been compressed with a JPEG quality factor of 20. The watermarked image at this point was showing heavy JPEG artifacts, reducing quality of the attacked image beyond usability.

The best comparison can be made with CDMA in the spatial domain. Although CDMA was more resistant to gaussian noise, comparison of DCT coefficients proved far more resistant to JPEG compression. This would tend to indicate that embedding the watermark in the same domain as expected transformations is clearly advantageous. By predicting which DCT coefficients would be altered using JPEG, an extremely high level of JPEG robustness was achieved.

Results of the correlation-based DCT techniques were fairly similar. Correlation-based DCT appeared to be slightly weaker for lower levels of distortion, yet stronger for the higher levels. Although threshold-based correlation in the DCT mid-band was implemented, the results obtained were even worse then the comparison-based approach shown below. We already know from the spatial correlation results that the use of two PN sequences is better in almost all aspects, and thus results of the threshold-based approach will not be shown here.

Comparison-based Correlation in the DCT mid-band

K = 15

Figure 26a - Watermarked Image
PSNR = 65.7 dB

Figure 26b - Recovered Watermark

Even without any modification to the watermarked image, we can observe a number of detection errors already present in the recovered watermark. A gain of 25 was required before the watermark would detect without bit errors, however at that gain the watermark embedding distortions had become quite noticeable. The zigzag like pattern introduced by this technique under high gain is highly regular and more distracting to the viewer then the slight noise introduced using the coefficient comparison from above. It is

worth noting however that this zigzag pattern would be eliminated if separate PN sequences were chosen for each block, however this would require a change back to threshold-based detection and so over-all performance would most likely suffer.

Comparison-Based Correlation in DCT mid-band Robustness

K=15

Figure 27a - 5% Gaussian Noise

Figure 27c – JPEG Compression Q=50

Figure 27b - 15% Gaussian Noise

Figure 27d – JPEG Compression Q=20

The most interesting result is that the recovered image after JPEG compression with quality factor 50 is actually **better** then the watermark recovered from a pristine cover. This would indicate that the detection errors in the un-modified source are right on the correlation boundary, and the addition of noise is enough to push them over the edge. What is strange however is that the modification doesn't produce the same bit errors as it does correction. Is this just a result an act of random chance, or does it suggest some sort of asymmetry in the detection process? Since the system is based on the correlation between two patterns, an asymmetry in detection probabilities between the two patterns would be a source of constant detection error during watermark recovery.

If we plot the correlation of both patterns over the entire length of the message sequence, everything appears to be normal. As the message contains more white (ones) then black (zeros), the asymmetry between their integrals can be expected. What is important however is that the average magnitude of detection is equal, which they appear to be. Most bits are detected with a correlation of close to one, with only a smaller number falling between {0,1}. It would appear that the improvement in detection accuracy through the addition of noise was merely chance, and not indicative of any underlying detection asymmetry.

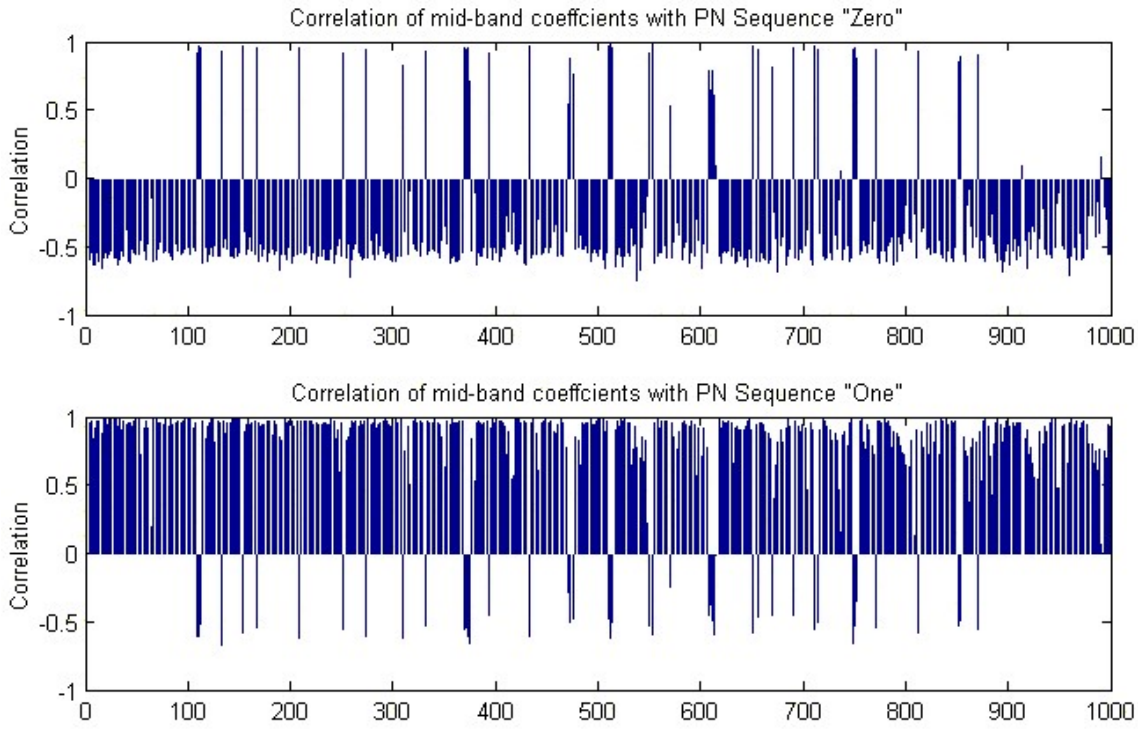

Figure 28 - Correlation Strength of PN Sequences

The block-based DCT techniques discussed share a common flaw with the spatial-domain correlation based techniques described previously; a fragility to scaling, cropping

and other geometric transforms. Cropping the image, or scaling to a non-multiple of the watermarked image offsets the coordinate system that block-based DCT depends on.

Possible improvements to this limitation would be insertion of a independent PN sequence as an “origin” through which all other locations are referenced. If this pattern were inserted in the center of the image, the first step of recover would be to slide the same PN sequences along the image and record the location at which the correlation peaks. The original orientation of the image could then be determined by rotating the PN sequence around the origin until the correlation peaked. Knowing the location of the origin and original orientation of the image, the image could be rotated back to its original orientation, and all blocks could be referenced from the calculated origin.

The technique could be further improved by employing CDMA techniques on the embedded blocks and reducing the message size, rather then the 1:1 ratio currently used. Employing this technique, the recovered watermark would degrade evenly as localized geometric damage was spread out across the entire watermark. The robustness of the watermark could thus be improved significantly.

The problem with this approach however is that it increases watermark recovery requirements significantly over the base technique. An exhaustive search of a PN sequences over each pixel of an M x N image requires a large number of calculations in addition to the calculations required by the base technique. Rotation calculations and recovery requires even more still. It’s possible that the exhaustive search required for an origin PN sequence of any significant size will outweigh any possible improvements in robustness.

Due to it's computationally efficient modeling of the HVS, the wavelet domain offers perhaps the most promising environment for robust watermarking. CDMA in the wavelet domain was first tested using the smaller message size, and then next using the normal message as per most of the other implementations.

CDMA Spread-Spectrum In the Wavelet Domain

K = 2 with small watermark

Figure 29a - Watermarked Image
PSNR = 68.1 dB

Figure 29b - Recovered Watermark

The algorithm seemed to have no problem retrieving the small watermark from the watermarked image with only minimal degradation of the cover image during embedding. Even with a minimal gain of k=2, the algorithm was still able to provide moderate robustness to gaussian noise and JPEG compression as shown in figure 30 below. The recovered watermark was even recognizable under heavy degradation of the cover such as 50% gaussian noise or JPEG compression with quality factor 20.

CDMA Spread-Spectrum In the Wavelet Domain Robustness

K=2

Figure 30a - 15% Gaussian Noise

Figure 30c – JPEG Compression Q=50

Figure 30b - 50% Gaussian Noise

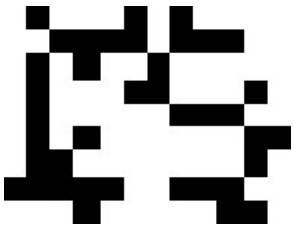

Figure 30d – JPEG Compression Q=20

While CDMA in the spatial domain degraded with increased image size, CDMA in the wavelet domain was able to encode the normal watermark, still with good results. In figure 31 below we see the 1000-bit normal message inserted into the cover with moderate gain, with only a single bit error during recovery.

CDMA Spread-Spectrum In the Wavelet Domain

K = 2 with normal watermark

Figure 31a - Watermarked Image
PSNR = 55.2 dB

Figure 31b - Recovered Watermark

Robustness results using the normal message size were also positive. In figure 32 below we see that the watermark was able to survive moderate levels of gaussian noise, while still be recognizable at detection. JPEG robustness faired even better, with the watermark still be recognizable after JPEG compression with quality factor 20.

CDMA Spread-Spectrum In the Wavelet Domain Robustness

K=2

Figure 32a - 5% Gaussian Noise

Figure 32c – JPEG Compression Q=50

Figure 32b - 15% Gaussian Noise

Figure 32d – JPEG Compression Q=20

The approach appears to combine the message capacity of block based techniques, with the noise immunity of spatial-domain CDMA. Furthermore, as the PN sequences are only half the size in this approach, the computational complexity of the algorithm is halved over spatial CDMA.

CDMA in the wavelet domain appears to show the most promise of the tested watermarking techniques. The algorithm described here is one of the most simplistic available in the wavelet domain, and yet the results are still excellent. These results tend to reinforce the common belief in wavelet domain as the most promising domain for digital watermarking [9].

Shown below in table 3 are approximate processing times for each of the algorithms described. Note that this table is only provided for a rough comparison. The algorithms were implemented in the most straightforward method possible…and not the most computationally efficient. Furthermore, the results shown here will not necessarily scale linearly over systems of varying architecture and speed.

		Inserted Watermark		
		Small	Normal	Large
Spatial Domain				
LSB Substitution	Embed			49
	Recover			3
Threshold-Based Correlation	Embed	0.5	0.6	
	Recover	0.2	0.4	
Comparison-Based Correlation	Embed	1.4	0.5	
	Recover	1.2	0.5	
CDMA Spread-Spectrum	Embed	14.9	132	
	Recover	27.2	249	
DCT Domain				
Comparison of mid-band DCT Coefficients	Embed	4.0	4.0	
	Recover	2.0	2.0	
Threshold-Based Correlation in DCT mid-band	Embed	3.8	3.8	
	Recover	3.9	4.0	
Comparison-Based Correlation in DCT mid-band	Embed	5.4	5.8	
	Recover	4.5	4.5	
Wavelet Domain				
CDMA Spread-Spectrum In the Wavelet Domain	Embed	9.0	67.5	
	Recover	13.6	119	

Table 3 - Processing Time in Seconds

That being said, CDMA in the spatial domain was clearly the most computationally intensive, requiring twice the processing time of its closest competitor and an order of magnitude above the average. CDMA in the wavelet domain is an improvement over the spatial domain, however the processing requirements are still quite high. Also note the highly non-linear behavior of the two CDMA sequences with increasing message sizes.

Conclusion

This study has introduced a number of techniques for the watermarking of digital images, as well as touching on the limitations and possibilities of each. Although only the very surface of the field was scratched, it was still enough to draw several conclusions about digital watermarking.

LSB substitution is not a very good candidate for digital watermarking due to its lack of even a minimal level of robustness. LSB embedded watermarks can easily be removed using techniques that do not visually degrade the image to the point of being noticeable. Furthermore if one of the more trivial embedding algorithms is used, the encoded message can be easily recovered and even altered by a 3rd party. It would appear that LSB will remain in the domain of steganography due to its tremendous information capacity.

Another observation is that transform domains are typically better candidates for watermarking then spatial, for both reasons of robustness as well as visual impact. Embedding in the DCT domain proved to be highly resistant to JPEG compression as well as significant amounts of random noise. By anticipating which coefficients would be modified by the subsequent transform and quantization, we were able to produce a watermarking technique with moderate robustness, good capacity, and low visual impact. This holds true in general for watermarking; robustness can be improved significantly when the subsequent degradation techniques are known. This holds particularly true in the case of compression techniques, were the compression algorithms are well known.

The wavelet domain as well proved to be highly resistant to both compression and noise, with minimal amounts of visual degradation. This is all the more impressive when one considers that the wavelet technique described here is one of the most primitive currently known. More sophisticated wavelet-domain techniques will almost certainly improve on both of these, and hopefully lower it's computational requirements. The wavelet domain may be one of the most promising domains for digital watermarking yet found.

A final note is that of geometric transforms. Geometric transforms are one of the most difficult for a watermarking technique to deal with. Embedding domains may be chosen that display both shifting or rotational invariance such as Cartesian or Polar DCT, however these domains are typically are resistant to only a specific geometric distortion and not the complete set. Furthermore, this greatly reduces our flexibility, as promising domains such as the DWT may no longer may be considered.

Although not discussed here, the counters proposed to these attacks typically rely on discovering the exact rotation, or shifting used in the attack, and then transforming the image back into its pre-attack state. Typically these techniques are computationally pricey, and unpredictable. This remains one of the major problems in the development of robust digital watermarking for digital images.

References

[1] J.A. Bloom, I.J. Cox, T. Kalker, J.M.G. Linnartz, M.L. Miller, C.B.S. Traw, "Copy Protection for DVD Video" in Proceedings of the IEEE, vol. 87, pp 1267,1272-1275, July 1999

[2] I.J. Cox, M.L. Miller, J.M.G. Linnartz, T. Kalker, "A Review of Watermarking Principles and Practices" in *Digital Signal Processing for Multimedia Systems*, K.K. Parhi, T. Nishitani, eds., New York, New York, Marcel Dekker, Inc., 1999, pp 461-482

[3] J. Dugelay, S. Roche, "A Survey of Current Watermarking Techniques" in *Information Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Northwood, MA: Artec House, Dec. 1999, pp 121-145

[4] R.C. Gonzalez, R.E. Woods, "Digital Image Processing", Upper Saddle River, New Jersey, Prentice Hall, Inc., 2002

[5] J.R. Hernandez, M.Amado, and F. Perez-Gonzalez, "DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis And a New Structure", in IEEE Trans. Image Processing, vol. 9, pp 55-68, Jan. 2000

[6] N.F. Johnson, S.C. Katezenbeisser, "A Survey of Steganographic Techniques" in *Information Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Northwood, MA: Artec House, Dec. 1999, pp 43-75

[7] S.C. Katzenbeisser, "Principles of Steganography" in *Information Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Northwood, MA: Artec House, Dec. 1999, pp 2-40

[8] M. Kutter, F. Hartung, "Introduction to Watermarking Techniques" in *Information Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Northwood, MA: Artec House, Dec. 1999, pp 97-119

[9] G. Langelaar, I. Setyawan, R.L. Lagendijk, "Watermarking Digital Image and Video Data", in IEEE Signal Processing Magazine, Vol 17, pp 20-43, September 2000

[10] J. Meel, "Spead Spectrum" De Nayer Institute, October 6th, 1999

[11] P. Meerwald, A. Uhl, "Watermark Security Via Wavelet Filter Parameterization", Internation Conference on Image Processing, Thessaloniki, Greece, 2001

[12] P. Meerwald, A. Uhl, "A Survey of Wavelet-Domain Watermarking Algorithms" EI San Jose, CA, USA, 2001

[13] H. Inoue, A. Miyazaki, T. Katsura "An Image Watermarking Method Based on the Wavelet Transform", Kyushu Multimedia System Research Laboratory.

[14] F.A.P. Petitcolas, , "Watermarking Schemes Evaluation" ", in IEEE Signal Processing Magazine, Vol 17, pp 58-64, September 2000

[15] F.A.P. Petitcolas, "Introduction to information hiding" in *Information Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Northwood, MA: Artec House, Dec. 1999, pp 1-11

[16] A.H. Tewfik, "Digital Watermarking", in IEEE Signal Processing Magazine, vol 17, pp 17-88, September 2000

Appendix

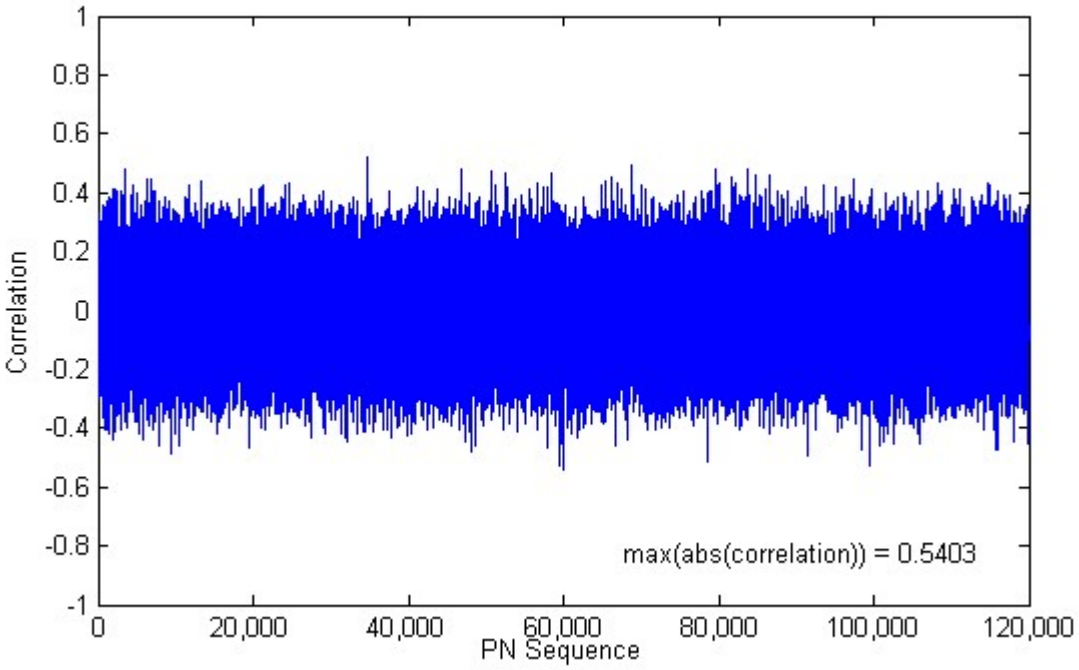

Figure 33- Correlation Between Baseline PN Sequence and Successive Sequences Using MATLAB rand() function

This plot was generated using **pn_period.m** to determine that the MATLAB random number generator would not generate a duplicate PN sequence within a reasonably large period. Clearly however the periodicity of MATLAB's random number generator is well above CDMA requirements.

Figure 34 – 35-digit "key" used as initial state of MATLAB random number generator

MATLAB code and full-size images from each of the tested techniques is available at: http://www.vu.union.edu/~shoemakc/watermarking/