High Bit rate Wavelet Domain Digital Watermarking of Images and Compression Tolerance

Ashoka Jayawardena, Bob Murison and Patrick Lenders

School of Mathematical and Computer Science
University of New England
Armidale, N.S.W

E-mail { ashoka, rmurison,pat}@mcs. une. edu. au

Abstract

The increased commercial activity in internet and media industry demands protection of media such as images, video and audio against unauthorised processing and use. Watermarking is a technique to hide information in media such that the hidden information (watermark) is invisible. This hidden information can be a small sequence of bits resulting in low bit rate watermarking. In low bit rate watermarking, each information bit is represented by an invisible broadband signal when added to the image. The hidden information may be in the form of images in which case large number of bits must be embedded. Such watermarks are known as high bit rate watermarks and are usually binary images.

Unlike low bit rate watermarking where each information bit is transformed to a broadband signal by generating a suitable pseudo-random sequence, high bit rate watermarking demands some other means of embedding and detection due to large number of information bits to be embedded. We use a binary feature based watermarking technique on wavelet domain. Our work is inspired by the work in [2].

Our motivation for this research is twofold. Firstly, we embed the watermark in wavelet domain rather than DCT domain motivated by the fact that the wavelet transform is used in jpeg2000 standardisation process. Secondly, high bit rate watermarks tend to get destroyed, even before average lossless compression ratios between 20% to 32%. We wanted to get the watermark breakup point up to at least within 20% to 32%.

Keywords Watermarking, image compression, image authentication.

1 Introduction

Watermarking aims at achieving copyright protection. This can be done by visibly or invisibly adding information to the image [2, 10, 4, 9, 11, 8, 6, 5] or documents [7] to be protected. The functionality provided by the copyright protection scheme varies depending on the application. The copyright can be a message which carry information regarding the ownership or an image which represents the ownership. The copyright images are usually known as stamps which visually modify the underlying image or the document while not totally destroying the visibility of the underlying image or the document. High bit rate watermarks are usually stamps which are embedded invisibly into the images.

In order to embed a binary watermark image into the original image, we need to select a suitable watermark and then place it's information in the image so that it does not corrupt the image, and is detectable only with the knowledge of the encryption keys and/or the original image.

Most natural images provides some capacity to embed additional information without causing noticeable perceptual differences. The amount of such information which can be embedded depends on the image. We expect such embedded information to survive legitimate image processing activities such as compression.

After recent success of image compression standards such as JPEG, we believe it is worthwhile to design the watermarking algorithms specific to the compression standards such as JPEG. We expect this will give us more capacity for information embedding thus suits for high bit rate watermarking.

We target the future JPEG200 standard which is based on wavelet transform. Due to the lack of information of final JPEG2000 compression algorithms, we choose SPIHT [1] as the compression algorithm due to its superior performance and simplicity. We expect the ideas implemented in this paper can be easily modified for JPEG2000. We have designed the embedding scheme to survive significant compression ratio under SPIHT compression, in particular we used the bit-plane ordering of SPIHT and embedded binary watermark variables onto bit-planes.

Proceedings of the Fourth Australasian Document Computing Symposium, Coffs Harbour, Australia, December 3, 1999.

2 Binary Feature Based Embedding Process

The watermark image is represented as

$$W = \{w(i,j) \text{ E } \{0,1\}, 0 < i < 2^{k_1}, 0 < j < 2^{k_2}\}$$

and image to be watermarked as

$$X = \{x(i,j), 0 \leq i \leq 2^{l_1}, 0 \leq j \, 2^{l_2}\}.$$

We used grey scale images with 8-bit pixels thus intensity ranging from 0 to 255.

Wavelet transform enables images to be transformed into multiresolution images which enables embedded progressive image coding [1] . In line with this property of the wavelet transform, we transform the watermark image to a multiresolution representation, $\psi_W(W)$,so that we can embed the watermark in a progressive manner. This multiresolution representation of the binary images is discussed in section 6.

For improved security we can encrypt the original watermark image and use this new encrypted image for watermark embedding. A simple technique which can be used is random permutation of watermark pixels. Since we apply a multiresolution transform on to the watermark image, we encrypt the subbands separately so that we do not lose the gains of the multiresolution transform. We call this new binary image as the encrypted watermark image which is represented by $\psi^*_W(W)$.

Any watermark embedding scheme must alter a selected portion or a set of pixels of the original image. We can allocate only small number of bits of the original image to each binary watermark image pixel. In high bit rate watermark embedding we cannot embed a particular image bit to the statistics such as adding a pseudo-random sequence of the given set of pixels due to its smaller size and statistics does not make sense when the size is small. Thus we look for binary features of$X$ so that we can embed the watermark bits by altering these binary features. The particular feature required is that if the binary value is flipped, ie $1 \rightarrow 0, 0 \rightarrow 1$, then there is no perceptual distortion of the image. This feature extraction process is denoted by $F_e(X)$which take the original image and returns the extracted feature image $f_e(i,j) = F_e(\psi_X(X))$. The way these features are found is explained in section 5.

Now the embedding process is to mix $F_e(\psi_X(X))$ and $\psi^*_W(W)$ so that with the knowledge of $F_e(\psi_X(X))$or $\psi^*_W(W)$ or nothing we are able to detect the encrypted watermarked image. We will denote the embedding process as $E(F_e(\psi_X(X)), \psi^*_W(W))$which takes the extracted feature image and the encrypted watermark image and returns the feature image to be updated, $f_u(i,j)$.Details of this embed operator

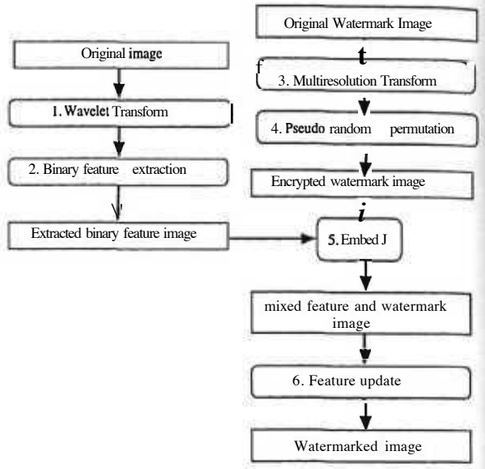

Figure 1: The high bit rate watermark embedding process

is discussed in section 4. Using this mixed feature image we will construct the watermarked image, $x_w(i,j)$. We denote feature update process as $F_u(X, E(F_e(\psi_X(X)), \psi^*_W(W)))$which takes the original image and the mixed feature image and returns the watermarked image. This feature based high bit rate watermark embedding process is described in Fig. 1.

3 The Detection Process

The detection process is comparatively simpler and is depicted in Fig. 2. Using the same parameters of feature extraction of embedding process, the feature image must be detected from the suspected image. This extracted feature image must be submitted to the detection operator which we will discuss in the next section. The detected feature image will be transformed to suspected watermarked image using the encryption parameters or the permutation matrix used in the embedding process.

This binary image which is in multiresolution representation, is inverse transformed back into the original image using the synthesis binary filters discussed in the section 6. This suspected watermark image will be compared with the original watermark image manually or automatically using a similarity measure.

4 Embed and Detect functions

Now we look for arguments which will reveal properties on Embed(E) and Detect(D) functions. We assume they are bitwise operators. The possible inputs to the detector are encrypted watermark image $p(i,j)$ or extracted feature image $f_e(i,j)$or nothing from the original image as the possible key

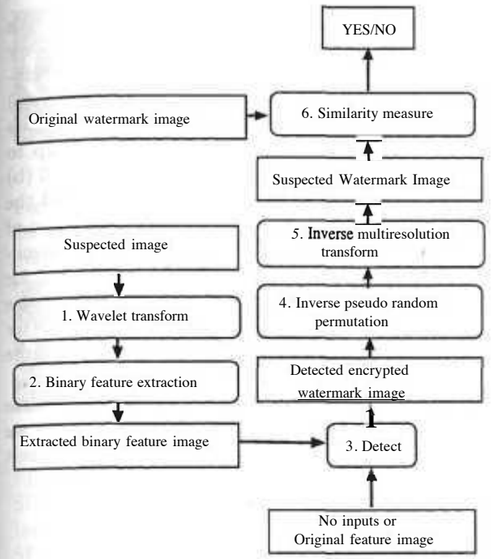

Figure 2: The high bit rate watermark detection process

information together with extracted feature image of the suspected image. If watermarked image is fetched into the detector we get the following two equations,

$$f_u(i,j) = f_e(i,j) \; E \; p(i,j)$$

$$p(i,j) = (\text{either } p(i,j) \text{ or } f_e(i,j) \text{or nothing})$$

$$D \quad f_u(i,j)$$

From the equation only $f_e(i,j)$makes sense and hence we define the detection operator as,

$$p(i,j) = f_e(i,j) \quad D \quad f_u(i,j).$$

The following theorem explains possible choices for the detector and embedder.

Theorem 1 *The only consistent choices for the operators E and D are given by the following table.*

	E	D
1	XOR	XOR
2	$\bar{P}$	$\bar{V}$
3	V	P
4	NOT XOR	NOT XOR

Proof: These cases can be proven using the following table.

ft	P					1	2	3	4
0	0	0	X	1	X	0	1	0	1
0	1	0	X	1	X	1	0	1	0
1	0	X	1	X	0	1	1	0	0
1	1	X	1	X	0	0	0	1	1
Validity		N	N	N	N	Y	Y	Y	Y

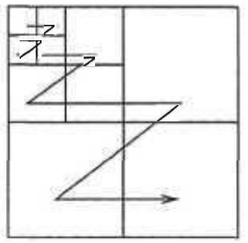

Figure 3: Scan order of coefficients

Only last four columns are the possibilities for identifying the E and D functions. The columns must be consistent in the sense of functions, i.e one-to-one mapping. We can identify these consistent columns by looking to identify f_u from f_e and p, and p from f_u and either f_e or p.

The second and third possibilities correspond to the detector with no inputs except the extracted feature image from the suspected image. Our next question is which is the best choice in terms of protection it provides. If we change the feature bits randomly, irrespective of which embedding operator we use, its going to effect the encrypted watermark image. Thus our criteria for the best choice is simplicity, and protection must be achieved from other parts of the embedding process such as feature extraction and encryption of watermark signal. Also we expect the watermark embedding process is image dependent otherwise an attacker will find it easy to identify the watermark if we embed the same watermark in different images. The choice 1 or 4 is suitable for cases where the feature extraction parameters are image independent as in [2]. Choice 2 or 3 is suitable when feature extraction parameters are image dependent as in our paper.

5 Bit Plane Embedding

Due to compression and perceptual significance all coefficients are not suitable for watermark embedding. We need to embed the watermark bits progressively such that embedded information is tolarent to compression. Our embedding algorithm is designed for bit-plane oriented image coding, in particular for SPIHT image coding algorithms. We embed the watermark bit to the wavelet coefficient by simply making the i^{th} bit plane value equal to the watermark bit while guaranteeing the error within a given bound (distortion step). The higher significant bit planes are coded earlier and hence we use a simple sorting algorithm to identify the coefficients and the corresponding bit planes in the most significant order.

Algorithm 1

- *Scan the subband in the order depicted in figure 3. Denote these coefficients* $c_0, ..., c_N$.

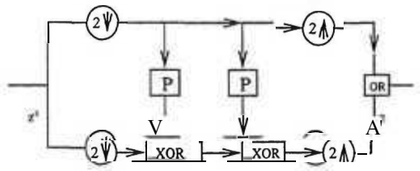

Figure 4: 2-channel Binary Signal Filter Bank

- *Let* $c_{max} = max\{c_0, ..., c_N\}$ *and* $p = 0$.
- *For* $i = \lfloor log_2 c_{max} \rfloor$ *to* $i = \lceil log_2 q \rceil$
  For $j = p$ *to* N
  If $c_j + 1$ *flips* i^{th} *bit plane*
  Swap c_p *and* c_j .
  Increment p.

6 Multiscale Transform of the Binary watermark Image

We have chosen the coefficients such that the chosen bitplane values for the watermark bits survive compression. In order to make use of this property we transform the watermark image to a multiresolution representation using the filter bank as shown in figure 4. Successive application of the filter bank to low pass subband yield the multiresolution representation. We used separable filters. Other such binary wavelet filters [3] can also be used.

The filter bank follows the lifting approach [12]. In the analysis side, original signal is separated into even and odd components by the down-sampling operators. The odd values are predicted from the even values and the prediction error is calculated using the XOR operator.

Again, in the synthesis side, odd values are predicted from the even values and original odd values are recovered from the XOR operation. The resulting odd and even values are interleaved using the up-sampling operator to construct the original signal. Notice, the perfect reconstruction of the filter bank is guaranteed since,

$$(o \ XOR \ e) \ XOR \ e) = (o \ XOR \ (e \ XOR \ e))$$

$$= o \ XOR \ false = o$$

where o is an odd value and e is a predicted odd value. Multiresolution representation of our watermark image is shown in 5 (b).

7 Results and Discussion

We have used the lena image in figure 9 and bike image in figure 10 used in JPEG2000 standardisation process for watermarking purposes. The watermarked lena image is given in figure 11 and the watermarked bike image in figure 12 for the distortion step 8 and decomposition levels 5. We have measured the compression performance of the watermarking algorithm under SPIHT compression. We have achieved subjective detection performance up to 9.4:1 compression ratio for lena and 11:1 for the bike image without zeroing of wavelet coefficients as given in figure 6 (a) and in figure 7 (a) respectively. With zeroing of wavelet coefficients we have achieved higher compression ratio up to 23:1 for lena and 32:1 for the bike as in figure 6 (b) and in figure 7 (b) respectively. We also tested the subjective performance with partial construction of the detected watermark. The partial reconstruction of the watermark at compression ratio 32:1 for lena is given in figure 8. The better performance of the bike image is due to its relatively better smoothness which results in the survival of more least significant bits in the compression process.

What is evident in our results is that to achieve watermark survival at high compression ratios we need either to design compression algorithm to suit the watermarking algorithm or vice versa.

References

[1] Said Amir and Pearlman William. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Tech.*, Volume 6, pages 243-250, June 1996.

[2] Hsu Chiou-Ting and Wu Ja-Ling. Hidden digital watermarks in images. *IEEE Trans. on Image Processing*, Volume 8, Number 1, pages 58–68, January 1999.

[3] Swanson M. D. and Tewfik A. H. A binary wavelet decomposition of binary images. *IEEE Trans. on Image Processing*, Volume 5, Number 12, December 1996.

[4] Swanson Mitchell D., Zhu Bin and Tewfik Ahmed H. Robust data hiding for images. In *IEEE Digital Signal Processing Workshop (DSP 96)*, Loen, Norway, September 1996.

[5] Kundur Deepa and Hatzinakos Dimitrios. A robust digital image watermarking method using wavelet-based fusion. In *Proc. IEEE Int. Conf. on Image Processing*, Santa Barbara, California, October 1997.

[6] Voyatzis G., Nikolaidis N. and Pitas I. Digital watermarking: An overview. In *Proc. of EUSIPCO'98*, Rodes, Greece, September 1998.

[7] Low S. H., Maxemchuk N. F., Brassil J. T. and O'Gorman L. Document marking and identification using both line and word shifting. In *Infocom'95*, Boston, Massachusetts, April 1995.

[8] Pitas Ioannis. A method for watermark casting on digital images. *IEEE Trans. on Circuits and Systems for Video Tech.*, Volume 8, Number 6, pages 775-780, October 1998.

[9] Cox Ingemar J., Kilian Joe, Leighton Tom and Shamoon Talal. Secure spread spectrum watermarking for multimedia. Technical Report 95-10, NEC Research Institute, Princeton,NJ, 1995.

[10] Smith Joshua R. and Comiskey Barrett 0. Modulation and information hiding in images. In *Proceedings of the First Information Hiding Workshop*, Isaac Newton Institute, Cambridge, U.K, may 1996.

[11] Wolfgang raymond B., Podilchuk Christine I. and Delp Edward J. Perceptual watermarks for digital images and video. In *Proceedings of IEEE*, July 1999.

[12] Sweldens W. The lifting scheme: A custom design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, Volume 3, Number 2, 1996.

a. b.

Figure 5: (a) The original watermark image (b) One level multiresolution transform of the watermark image

a. b.

Figure 6: (a) The detected watermark image after SPIHT compression ratio of 9.4:1 with no smoothing. (b) The detected watermark image after SPIHT compression ratio of 23:1 with smoothing for lena image.

a. b.

Figure 7: (a) The detected watermark image after SPIHT compression ratio of 11:1 with no smoothing. (b) The detected watermark image after SPIHT compression ratio of 32:1 with smoothing for bike image.

Figure 8: The detected watermark image with partial reconstruction after SPIHT compression ratio of 32:1 with smoothing for lena image.

Figure 9: The original lena image

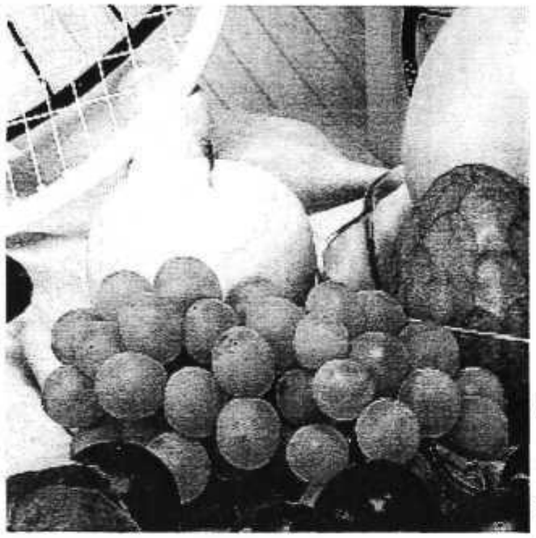

Figure 10: The original bike image

Figure 12: The watermarked bike image

Figure 11: The watermarked lena image

The use of argumentation to assist in the generation of legal documents

John Yearwood
School of Information Technology and Mathematical Sciences
University of Ballarat, Ballarat, Victoria, Australia
j.yearwood@ballarat.edu.au

Andrew Stranieri
Donald Berman Laboratory for Information Technology and Law, Dept of Computer Science and Computer Engineering
La Trobe University, Bundoora, Victoria, Australia
stranier@cs.latrobe.edu.au

Chaula Anjaria
School of Information Technology and Mathematical Sciences
University of Ballarat, Ballarat, Victoria, Australia
c.anjara@ballarat.edu.au

Abstract

Many text documents in the legal domain are created in order to express the reasoning steps a decision maker followed in reaching conclusions. For example, refugee law determinations are documents that express the reasoning steps a member of the Refugee Review Tribunal in Australia followed in order to infer conclusions regarding the status of an applicant. Although, it is reasonable to expect that a mapping between the reasoning steps used by a decision maker and the structure of the document produced would clearly be apparent, a number of authors have discovered that such a mapping is by no means obvious. In order to develop legal knowledge based systems that generate documents from their own reasoning steps, discourse analysis is invoked to bridge the gap and perform the mapping. In this paper, we articulate a heuristic that we use to generate a plausible document structure without the use of discourse analysis. Without discourse analysis, the heuristic cannot ***contribute*** *to our understanding of the process employed by decision makers to convert reasoning to text. Nevertheless, the heuristic can mimic the process. The heuristic has been trialed with a small sample of refugee law determinations by extracting the reasoning steps from each determination and applying the heuristic to reproduce each document's structure.*

Keywords Document generation, argumentation, refugee law.

1 Introduction

In many applications of human reasoning, conclusions ultimately reached and the reasoning steps employed to reach conclusions are expressed in written natural language. For example, the inferences that members of the Refugee Review Tribunal (RRT) make in assessing claims for asylum seekers to remain in Australia as refugees are natural language documents called determinations that vary from 6 pages to many tens of pages in length and only loosely conform to a pre-defined structure. However, although determinations reflect reasoning, each one is written in a style that is not obviously consistent with a representation of refugee reasoning that we have developed over the last two years in close collaboration with RRT members.

The disparity between a natural language document and a representation of reasoning that a document expresses has been noticed by a number of authors using different knowledge representation schemes so is likely to be an artifact of communication styles rather than a peculiarity of any one knowledge representation scheme. Dick [2], in translating legal decisions that spanned hundreds of years into conceptual graph frames initially attempted to do so by creating graphs directly from text components. This proved to be too difficult because of the disparity between the document text and a conceptual graph representation of the reasoning that the document expressed. She proceeded by reading each judgement in its entirety, then formulated a set of conceptual graphs from her understanding and not from the text.

Proceedings of the 4th Australasian Document Computing Symposium,
Coffs Harbour, Australia,
December 3, 1999.