

國立交通大學

資訊管理研究所

碩士論文

以均勻 Deadzone 為基礎之小波樹量化的

數位影像浮水印

Uniform Deadzone Based Wavelet Tree quantization for
Digital Image Watermarking

研究生：林承龍

指導教授：蔡銘箴博士

中華民國九十四年六月


以均勻 Deadzone 為基礎之小波樹量化的數位影像浮水印

學生：林承龍

指導教授：蔡銘箴

國立交通大學資訊管理研究所

摘要



資訊的數位化使得數位內容保護版權保護受到重視，數位浮水印技術的發展提供一種數位權利保護方法。數位浮水印須同時具備透明度、安全性、明確性、強韌性、容量及不需來源比對等的要求，能夠承受來自外在的幾何攻擊、壓縮攻擊等惡意破壞。本研究以小波轉換理論為基礎，以均勻量化方法將數位浮水印嵌於由中頻係數組成之小波樹中。而小波樹之量化，則是在不影響圖形品質之下決定之。此外，本研究提出小波濾波器組選擇原則，適用的 filter bank 將對於解浮水印有正面的幫助。而本研究提出之均勻 Deadzone Scalar Quantization 的方法，使以小波樹為基礎之浮水印演算法有更高的強韌性，在數位智財權的保護上提供更有效的技術。

關鍵字：

數位影像浮水印、小波轉換、小波樹量化、deadzone scalar quantization

Uniform Deadzone Based Wavelet Tree Quantization for Digital Image Watermarking

Student: Chen-Long Lin

Advisor: Min-Jen Tsai

Institute of Information Management
National Chiao-Tung University

ABSTRACT

The digitization of information makes digital content copyright protection attracts much attention. The development of digital watermarking technique provides a way for digital rights protection. A good watermarking scheme must have the characteristics of transparency, safety, unambiguous, robustness, capacity and blindness simultaneously. Furthermore, digital watermark must resistant to malicious geometric attacks and compression. This research is based on wavelet transformation theory, in which an uniform quantization method is used to embed watermark in Super Trees which chosen from wavelet middle-band coefficients. The quantity of the quantization is dynamically decided by image quality. This research focuses on Deadzone Scalar Quantization which provides the robustness for watermark extraction and provides a effect technique on digital rights protection.

Keywords:

digital image watermarking, wavelet packet, wavelet tree quantization, deadzone scalar quantization.

誌謝

承蒙蔡銘箴老師的細心指導，感謝老師在這兩年間培養學生報告的技巧、獨立思考的能力與創新的研究精神，並在生活上給予關心與幫助，在此謹向吾師致上衷心的感謝與敬意。本論文的完成，同時也感謝口試老師林源倍教授、杭學鳴教授在論文上的細心評閱與指正，使得本論文在整體上更為充實完整。林源倍教授及王士豪學長的論文對我提供了一些新的想法，並且非常感謝老師與學長，給予我在程式驗證之協助。

碩士兩年的時間很快就過去了，在段時間裡也感謝實驗室畢業的學長曾立信傳下來的小波轉換程式，及以前常一起打麻將、聊天的學姊千毓、筱盈、viviann、秋雅。此外與博宇、冠輝不論在課業研究、多媒體操作、打壘球、聊天講笑話及討論總是能合作無間使人心情愉快。也感謝實驗室王振生學長、學弟聖閔、國鼎、紹榮、政良、錡樂及昌興平時的支持。

最後將此論文獻給我最敬愛的父母與家人，感謝您們在我求學時期全心全意給予我的支持與鼓勵，若不是您們長久以來的支持，不可能有我今天的小小成果。也將此論文獻給所有關心我的朋友，並敬上最真誠的敬意與謝意。



林承龍

2005年7月15日

謹於交通大學資訊管理研究所

目錄

	頁次
第 1 章 緒論	1
1.1 研究背景.....	1
1.2 研究動機與目的.....	1
1.3 研究重點.....	2
1.4 論文架構.....	2
第二章 文獻探討	4
2.1 資訊隱藏.....	4
2.2 數位浮水印.....	6
2.2.1 數位浮水印之定義及用途.....	6
2.2.2 浮水印設計之考慮因素.....	7
2.2.3 浮水印之種類.....	8
2.2.4 浮水印嵌入方式.....	10
2.3 小波轉換.....	12
2.3.1 小波轉換簡介.....	12
2.3.3 濾波器組理論：.....	12
2.3.4 小波頻率域特性：.....	14
2.4 Correlation – based watermark detection	15
第三章 非均勻量化Super Tree演算法	18
3.1 非均勻量化Super Tree浮水印嵌入流程	18
3.1.1 浮水印的產生.....	18
3.1.2 小波轉換：.....	19
3.1.3 Super Tree的選擇：.....	20
3.1.4 Super Tree Bitplane.....	21

3.1.5 位元平面量化.....	22
3.1.6 浮水印的嵌入.....	23
3.2 非均勻量化Super Tree浮水印取出流程.....	24
3.2.1 Re-quantization and watermark extraction.....	24
3.2.2 浮水印之驗證：.....	25
3.3 非均量化Super Tree浮水印技術之缺點.....	27
3.3.1 Case 1：(當 $q'_n > q_n$ 時發生).....	27
3.3.2 Case 2：(當 $q'_n = q_{\max}$ 時發生).....	30
第四章 Deadzone Scalar Quantization Watermarking Algorithm.....	34
4.1 浮水印之嵌入方法.....	34
4.1.1 Minimum quantization step 之設計.....	34
4.2 誤差對於解浮水印過程之影響.....	36
4.3 Filter Bank之選擇.....	39
4.3.1 Filter Bank 對Super Tree係數Magnitude大小之影響。.....	39
4.3.2 Filter Bank對 $T_{e_i}(j)$ 大小之影響。.....	41
4.4 Uniform Deadzone Quantization Algorithm for Watermark Extraction.....	43
4.4.1 Decoding實作原理.....	43
4.4.2 浮水印位元取出流程.....	44
4.5 Normalized Correlation與False positive probability分析.....	48
4.6 完整浮水印嵌入與取出之演算法.....	51
4.6.1 浮水印嵌入演算法.....	51
4.6.2 浮水印取出演算法.....	51
第五章、研究成果.....	53
5.1 JPEG Compression.....	55
5.2 SPIHT compression.....	58

5.3 Median Filter Attack	60
5.4 Pixel Shifting Attack	63
5.5 Bit-plane Remove Attack	66
5.6 Multiple Watermark Attacks	68
5.7 Rotation and Scaling	70
5.8 Filter convolution	71
5.9 比較結果整理	72
六、結論與展望	73
參考論文	75
附錄(一) Filter Banks	79
附錄(二) 著作	81



圖目錄

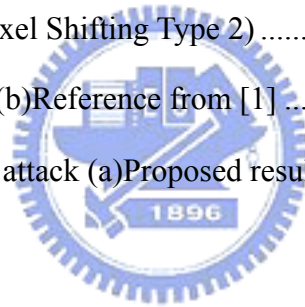
	頁次
圖 1、LSB嵌入示意圖	10
圖 2、利用Transform domain 嵌入數位浮水印示意圖	11
圖 3、一維小波轉換之分解與合成示意圖 (a)訊號分解 (b)訊號合成	13
圖 4、基本的浮水印偵測流程	15
圖 5、門檻值由期望值判率決定	16
圖 6、非均勻量化Super Tree浮水印嵌入流程	18
圖 7、小波轉換示意圖 (a)將host image進行小波分解，共分解成四個level，每個level包含了LL,LH HL與HH四個子頻道 (b)經(a)分解後，實際上每個子頻道於圖形中相對應之位置	19
圖 8、Wavelet Tree、Super Tree之定義 (a)Wavelet tree的選擇 (b)A Wavelet Tree (c)A Super Tree	21
圖 9、Super Tree轉換為二位平面位元之範例	22
圖 10、 q_n 決定一棵Super Tree量化的量。	23
圖 11、非均勻量化Super Tree浮水印取出流程	24
圖 12、decoding找到的 q'_n 大於encoding量化使用 q_n 之示意圖。	27
圖 13、Case 1 無法正確判斷浮水印位元之範例 (a) p_i in encoding (b) p'_i in decoding (when $q'_n = q_n$) (c)Re-quantization完成 in decoding	29
圖 14、圖 13 中實際產生判定之誤差	30
圖 15、(a)如果 T_{2i-1} 或 T_{2i} 的 M_{2i-1} 或 M_{2i} 有任一個小於我們所設定之reference error，則 q_n 便會達到 q_{\max} (b)對 T_{2i-1} 量化後的結果	31
圖 16、接續圖 15，在decoding中實際判定結果	32
圖 17、以 100 組seeds為測試，上方A線條表示 $q'_n = q_{\max}$ 發生次數，中間B線條表示在 $q'_n = q_{\max}$ 之下，浮水印位元發生誤判次數。	33
圖 18、本文於encoding採用的位元平面量化方法	35
圖 19、 $Te_i(j)$ 誤差之產生範例	36
圖 20、Quantization and rounding errors in encoding	37

圖 21、以 $f_1(m)$ 檢測五組Filter Banks之magnitude大小.....	40
圖 22、以 $f_2(e)$ 檢測四組Filter Banks產生的 $T_{e_i}(j)$ 大小.....	41
圖 23、Proposed watermark extraction process.....	43
圖 24、Deadzone Scalar quantization 示意圖.....	45
圖 25、一個對於 (T'_{2i-1}, T_{2i}) 以deadzone scalar quantization之範例。.....	47
圖 26、實驗所使用的三張影像.....	53
圖 27、原始Lena影像與以JPEG壓縮對影像攻擊過影像之比較 (a)原始影像 (b)JPEG壓縮過之影像。.....	55
圖 28、三張影像對於JPEG攻擊之承受程度 (a)Lena (b)Goldhill (c)Peppers.....	57
圖 29、原始Lena影像與SPIHT壓影像之對照(a)原始影像(b)SPIHT壓縮後影像.....	58
圖 30、Median Filter 運作原理.....	60
圖 31、原始Lena影像與median filter處理過影像之對照 (a)原始影像 (b)以 6×6 median filter處理過之影像.....	61
圖 32、原始Peppers影像與type 1 shifting影像之對照 (a)原始影像 (b)以type1 shift 9 個pixels之影像.....	63
圖 33、原始Peppers影像與type 2 shifting影像之對照 (a)原始影像 (b)以type2 shift 9 個pixels之影像.....	64
圖 34、原始Lena影像與移除空間域中每個像素 4 個LSB影像之對照 (a)原始影 像 (b)移除空間域中 4 個LSB對影像破壞結果.....	67
圖 35、原始Goldhill影像與額外加入 4 個浮水印之影像之對照 (a)原始影像 (b) 加入 4 個浮水印之破壞結果.....	68

表目錄

頁次

表格 1、使用Normalized Correlation時，不同的浮水印特性，可以有不同的false positive probability估算方法。	17
表格 2、不同的 ρ_r 與 N_w 得到的False positive probability	50
表格 3、浮水印抗攻擊實驗列表.....	54
表格 4、SPIHT compression (a)Proposed results (b)Reference from [1]	59
表格 5、Median filter attack (a)Proposed results (b) Reference from[1].....	62
表格 6、Pixel shifting attack (a)Proposed results(Pixel Shifting Type 1) (b) Reference from[1] (Pixel Shifting Type 1) (c) Proposed results(Pixel Shifting Type 2) (d) Reference from[1] (Pixel Shifting Type 2)	65
表格 7、(a)Proposed results (b)Reference from [1]	67
表格 8、Multiple watermark attack (a)Proposed results (b)Reference from [1]	69



第 1 章 緒論

1.1 研究背景

數位化技術的廣泛應用，引發了一次技術革命。目前的科技已能將任何傳統媒介轉換為數位電子媒介，而電腦的儲存量幾無限制，既省錢又極省空間，因此，電腦能很容易匯集大量數位資訊一起處理。有鑑於資訊科技的進步與通訊網路的普及，數位化的資訊內容將成為人類擷取資訊的主要來源，對於數位內容的保護也愈來愈重要。

數位化及網路化的科技發展，使得著作者、版權擁有者與網路服務提供者之義務複雜化，為了因應數位化所帶來的數位所有權問題，我國著作權法為保護著作人權利、平衡使用者及社會公共利益，於民國 92 年 7 月 9 日公佈新著作權法，針對傳統法律對於數位內容之不足處加以增加、修訂，以協助數位內容相關產業之發展。

1.2 研究動機與目的

對於數位內容的保護，除了對數位內容以密碼加密，防止數位內容在傳輸的過程中，被未授權之第三者非法取得之外，一個合法的使用者，仍然有可能將所獲取的數位內容進行未授權的複製行為，然後再轉移到他處使用。然而如果於數位內容中加入數位浮水印，一旦發現數位內容有版權爭議時，便可利用數位浮水印當作版權侵犯之證據。

雖然數位影像浮水印技術尚不能解決所有智財權的問題。但大多數的研究者均同意，它仍然具備強大的潛力吸引著作權所有者，提供可信賴的智財權保護機

制。本研究基於上述之研究動機，希望能夠以小波轉換理論為基礎，利用小波轉換後的 wavelet tree 係數值之統計特性，於已嵌入浮水印的小波樹中，找出所嵌入之浮水印位元，並期待能夠將之應用於數位影像著作權利之保護之上。

1.3 研究重點

一個良好的數位浮水印必須具備有，強健性 (Robustness)、不可見性 (Invisible)、明確性 (Unambiguous)、不可被統計性 (Statistically undetectable)、驗證不需原始數位影像 (Blindness) 及安全性 (Security) 等特性。本研究期許建立一個符合上述特性之浮水印演算法。

本文方法建立在[1]的 Wavelet Tree quantization algorithm 之上，經由浮水印 decoder 之精確度分析、filter bank 選擇及 Deadzone Scalar quantization 之應用，以增加每個浮水印位元解出之機率，同時兼顧浮水印的安全性、強韌性

1.4 論文架構

本論文共分為六個部份，來探討如何基於小波樹量化方法，來實做出一個具有一定強韌性之數位浮水印，論文的各章節架構簡述如下：

第一章、緒論：

本章對於數位影像技術發展之現況事實做陳述。並分析一個好的數位影像浮水印所應具有的條件，對於整個研究的方向做一個概述。

第二章、文獻探討：

本章內容為述說資訊隱藏簡介、數位浮水印之定義用途及種類，並討論

小波轉換及 Correlation based watermark detection，希望能從中找到浮水印設計之方向、方法及浮水印驗證之可靠性。

第三章、非均勻量化 Super Tree 演算法：

詳細說明參考論文[1]之浮水印嵌入、取出流程及原理，並在 3.3 節提出可能導致誤判之原因及實例，並從中討論可能的改善方法，以找到可行的改善方法。

第四章、Deadzone Scalar Quantization Watermarking Algorithm：

提出本論文浮水印嵌入、取出演算法，並選擇適用本方法之 Filter Bank，同時達到保有精確度及強韌性之浮水印。

第五章、研究成果：

針對實驗影像，以不同的頻率域、空間域攻擊方法進行浮水印強韌性測試，並與[1]結果比較。



第六章、結論與展望：

提出本文貢獻及未來研究之方向。

第二章 文獻探討

2.1 資訊隱藏

資訊隱藏現在多以Information Hiding、Data Hiding來稱呼，然而在早期的資訊隱藏則是使用”Steganography”來代表。Steganography其實可以拆成兩個字，分別是Steganos與graph。在希臘文”Steganos”代表遮蔽、隱藏之意，”graph”代表寫作、寫到某處之意，所以將Steganos與graph合起來，表示將所寫的字隱藏至某處，也就是資訊隱藏的意思[2]。希臘歷史學家希羅多德 (Herodotus, 484~425BC) 在他的著作曾提到關於資訊隱藏的例子，例如將奴隸的頭髮剃光，然後在頭皮上刺上所要傳遞的秘密訊息，待其頭髮又長出後才將其送出。在這個過程中奴隸的頭皮上存放的秘密訊息便可被安全地送出，外界無法得知此訊息。

在 Trithemius 的 1499 年所寫的 Steganographia 這本書源自於希臘字 $\sigma\tau\epsilon\rho\alpha\nu\delta\varsigma$ ， $\gamma\rho\alpha\varphi-\epsilon\iota\nu$ 意指”covered writing”，也就是藏避所寫的資訊[3]。在書中介紹了一種類似的方法，就是在一堆無意義的字(word)中，如

Parmesiel Oshurmi Delmuson Thafloin Peano Charustrea Melany Lyamunto .

在每兩個字中萃取每兩個字元來，透露出以下的拉丁文訊息：

Sum tali cautela ut .

在二十世紀，Steganography 一直到第二次世界大戰才又受到廣泛的注意，美國在所有的郵件與電報通道上過濾所有往來的訊息，以防止、偵測間諜的秘密通訊。這些審查員所採用的檢查技術暗示了早期這些隱藏技術設計者 (steganographer)所展現的發明才能。美國稽查局 (The Office of Censorship) 儘可能地禁止所有可能用來傳遞訊息的通道，從西洋棋、紡織圖案到小孩子的塗鴨等

等都在禁止之列。同樣地 Crossword、Newspaper clippings 也被禁止。郵票因為其面值可以被用來解譯成訊息，所以被限制成一共同型式。

Covert Channels 的研究其實是近代資訊隱藏技術研究的開始，其中以 Lampson 為最早[4]。Lampson 定義所謂秘密通道(Covert Channels)，指的是透過一個私密的通道來傳遞秘密的資訊。在 Cover Channels 的一些應用上，例如軍方在建立軍事通訊所使用的特殊頻道，或是一些特別的資訊交換方式，都能夠稱為 Cover Channels，在這樣的機制底下傳遞機密訊息不易被外界所發現。

另外根據 1985 在 U. S. Department of Defense Publication 中，標題為 “Trusted Computer System Evaluation” 的文章[5]，對於 Cover Channel 的定義，如下：

“... any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy.” 秘密通道能夠確保資訊的隱密及安全性。

由於數位科技的進步及資料的數位化等影響，在近代的資訊隱藏研究中數位資訊隱藏技術成為熱門的研究題材。其中以數位浮水印及數位指紋最為興盛。數位浮水印可將一些智慧財產權的訊息，例如原作者，擁有者，出版處，連絡公司地址等等隱藏在數位媒體產品上。有關數位浮水印的研究將在 2.2 節做更詳細的介紹。數位指紋則可將每一個商品給一個不同的商品編號，以便日後若有非法使用的控訴，可將由一些仲裁機制，將非法拷貝者，或傳播者找出。

2.2 數位浮水印

2.2.1 數位浮水印之定義及用途

數位影像浮水印技術是利用資料隱藏(data hiding)的技術，將一些具有代表性的版權資訊嵌入數位影像中，並可利用特定的方法將版權資訊取出分析，以證明版權之所屬。例如將作者的資訊、版權聲明、產品序號等資訊嵌入於聲音(audio)、影像(image)、視訊(video)等不同數位媒體中。浮水印之應用依據不同的需求，可以分成下列幾項[6]：

1. 版權保護(Copy right protection)

此為數位浮水印最常被使用的用途。之所以能保護數位影像版權，是由於浮水印必須具備有相當的強韌性(Robustness)，可以抵抗各種影像處理攻擊、防止偽造攻擊。並且浮水印是獨一無二的，能明確辨認出版權擁有者(Unambiguous)。



2. 認證與完整性確認(Authentication and Integrity verification)

一般而言強韌性較低的脆弱浮水印(fragile watermark)[7]或半易碎型浮水印(semi-fragile watermark)常用於影像竄改偵測或影像完整性之應用。例如軍事或醫學影像必須防止任意的竄改，並保證影像的正確性。雖然密碼學同樣能達到認證及資料完整性確認，不過可認證的數位浮水印因為具有與資料不可分離之優點，故在資料處理上更為容易。

3. 追蹤來源(Tracking / Traitor Tracing)

為了追蹤數位內容於網路上非法散佈、使用、拷貝、轉移等問題，在數位內容出售前，賣方會事先在每一個數位內容中各別加入一組獨一無二的數位指紋

(Fingerprinting)。Fingerprinting 能確認數位內容的買方，具有不可否認性，如果日後發生了非法散佈產品的情形，業者可以根據 Fingerprinting 來查出非法散佈的來源。

2.2.2 浮水印設計之考慮因素

根據論文[8][9]中整理，浮水印設計之考慮因素有以下幾點：

1. 透明度 (Transparency)

浮水印加入影像後，能夠不影響到影像本身品質為佳，如果圖形品質因此被影響了，便可能失去影像本身的美觀與商業價值。而原始影像與加入浮水印之影像品質，一般使用 PSNR 來判定。而 Human Visual System(HVS)亦能用於評判加入浮水印後圖形，對於人類視覺感觀影響程度[10][11]。

2. 安全性(Security)

為了不讓惡意攻擊者輕易猜出浮水印嵌入方法及順序，通常浮水印的演算法中都會加入 Secret key。Decoder 必須使用與 encoder 相同的 key 才能順利解出浮水印，這也使攻擊者不容易猜測浮水印嵌入方式。

3. 明確性(Unambiguous)

嵌入影像中的浮水印，應該能明確辨認版權的所有者或著作者，以保護數位影像創作者本身的權利。

4. 強韌性(Robustness)

浮水印嵌入演算法必須符合強韌性，嵌入浮水印之影像要能夠承受外在的攻擊(包括各種空間域攻擊、壓縮攻擊及頻率域攻擊...等等)。並且由受過攻擊的待測影像中順利取出浮水印以證明版權。

5. 容量(Capacity)

浮水印演算法須能加入不同浮水印長度，當加入的浮水印長度愈長時，浮水印驗證之精度便能提高。因此好的浮水印演算法必須盡可能容納更多浮水印資訊。不過一般來說浮水印嵌入長度愈長時可能不能滿足 Transparency 的要求。如果浮水印嵌入長度愈長，同時要求相同的 Transparency，則很可能使得浮水印 Robustness 不足。因此浮水印之設計必須同時考慮 Capacity、Transparency 與 Robustness 這三個重要因素。



6. 是否需要來源影像之比對(Blindness)

於浮水印取出演算法中，是否需要使用原圖進行比對。一張影像在網路上流通，如果要證明版權時沒有原圖就無法證明版權，這是非常不方便的；同時 encoder 要傳送大量資料(指的是原圖)給 decoder，相當花費資源且安全性低，因此近期的浮水印研究都是以不需要原圖的技術為主。

2.2.3 浮水印之種類

依可視性可以分為：

1. 可視浮水印(Visible watermark)

浮水印資料加入影像後，可以察覺浮水印的存在。雖然可以明確辨識版權資

訊，不過數位內容可能因此失去美觀與商業價值。

2. 不可視浮水印(In-visible watermark)

使用資訊隱藏技術，數位版權資料加入影像後，無法由肉眼察覺浮水印的存在者。這個方法不會破壞數位內容的美觀，不過版權的辨識須先解析出浮水印才能完成。

另外，常見的數位浮水印有下面三種形式呈現：

1. 圖形標誌(Logo、Mark)

即嵌入一個可以代表版權圖用者之圖示。當圖形標誌為可視浮水印時，可以憑肉眼就能辨認版權所屬。當圖形標誌為不可視浮水印時，由驗證程度所解出的浮水印與原浮水印進行比較，如果錯誤率達到一定程度以下，則可以宣稱版權的所有[12]-[16]。



2. 數字序列(Number sequence)

這類浮水印的產生一般而言是依據特定的統計分配而得來的 (如 Normal distribution)，將浮水印視為 noise 加入影像頻率域中以供版權認證。當浮水印由待測影像中解出後，解出的數字序列與原始數字序列之比對同樣是依據統計學上的一些特性加以完成[17]。

3. 文字訊息(Text message)

直接於數位內容中嵌入一段有意義之文字，如公司名稱、住址...等，它可以是可視的，或不可視浮水印。這一類的浮水印具有明確的辨認版權效果。不可視的文字訊息浮水印對於外在攻擊比較不具有強韌性，因為一旦受到攻擊，則文字

內容便出差錯，所以尚需加上一些偵錯機制來預防錯誤之發生。一般市面上數位浮水印產品大多數都提供文字訊息嵌入。

2.2.4 浮水印嵌入方式

空間域浮水印

此類浮水印是以直接修改影像的像素(Pixel)做為嵌入的方法，它可以是將版權訊息直接顯示於圖形上的可視浮水印，也可以是不可視浮水印。例如[18]-[22]中，皆提出以 LSB(Least Significant Bit)做為浮水印之嵌入區域，因為在空間域中每一個像素的 LSB 即是影像中比較不重要的位元，在其中嵌入資訊對於影像品質的破壞程度比較不高，不過如果嵌入 LSB 之位元數太少則浮水印容易遭到外在破壞。如圖 1 所示，一個值為 51 之像素，修改其 LSB 後數值產生變化，同時也等於達到浮水印嵌入之效果。



圖 1、LSB 嵌入示意圖

頻率域浮水印

目前的數位浮水印技術大多使用頻率域的嵌入方式居多，如圖 2 所示。一般而言，原始影像經過 transform domain 的轉換，由空間域轉換至頻率域，再以嵌入演算法把浮水印藏入頻率域係數中，接下來由 inverse transform 轉換回頻率域，達到浮水印之嵌入。當我們欲檢驗一張待測影像是否含有浮水印時同樣經由領域轉換至頻率域，再取出浮水印以檢測取出之浮水印與原浮水印之相關性，如

圖 2(b)。常用的轉換方法有離散餘弦轉換(Discrete Cosine Transform)、離散傅立葉轉換(Discrete Fourier Transform)以及離散小波轉換(Discrete Wavelet Transform)。這一類的浮水印嵌入過程是比較花費時間的，不過也因為嵌於頻率中，一般在空間域中的影像破壞對於這類浮水印影像比較不大。一般而言在影像頻率域的中頻區域是比較適合浮水印嵌入的，2.3.4 節中我們將進一步討論。

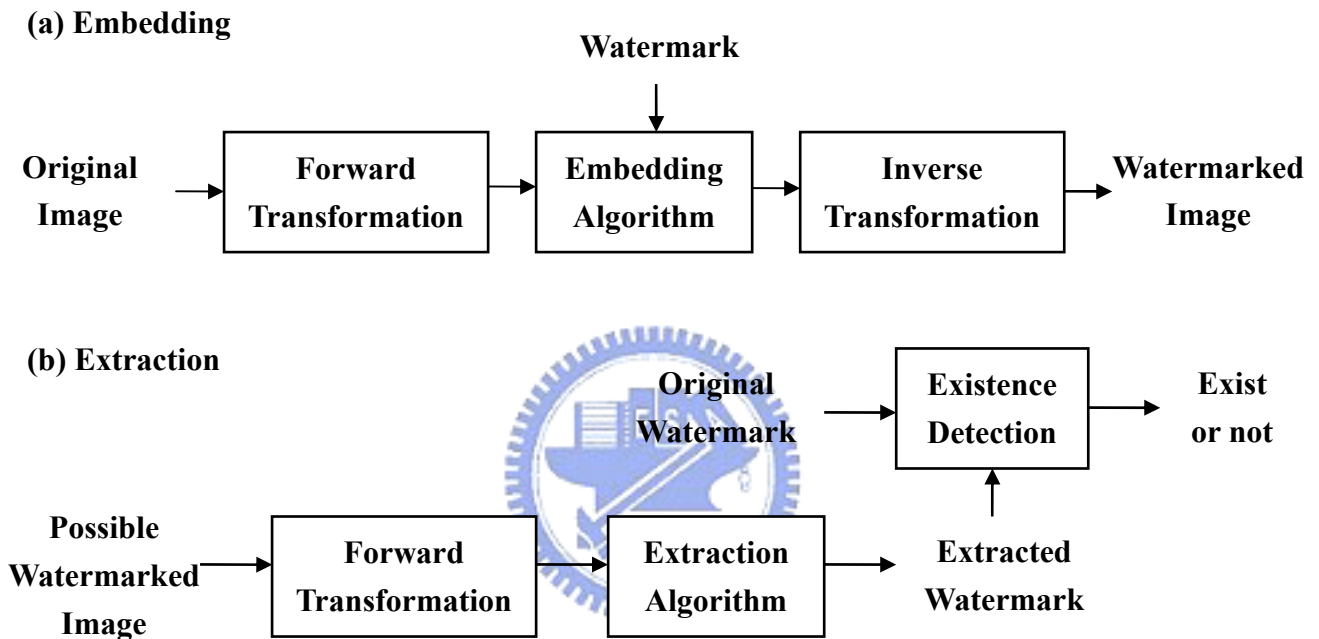


圖 2、利用 Transform domain 嵌入數位浮水印示意圖

2.3 小波轉換

2.3.1 小波轉換簡介

小波轉換(Wavelet Transformation)源起於 Joseph Fourier 的熱力學公式。傅利葉方程式在十九世紀初期由 Joseph Fourier (1768-1830)所提出，它是現代信號分析的基礎，同時在十九到二十世紀的數學研究領域中，也佔有重要的地位。小波轉換是近幾年來發展出來的數學理論，它是傅利葉轉換的延伸。

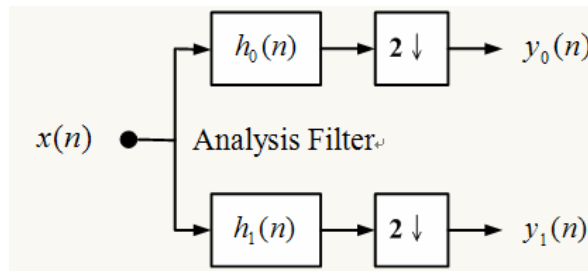
小波轉換方法的提出可追溯到 1910 年 Haar 提出小波概念，將訊號處理的方法往前跨進一大步[23]。其後 1984 年，法國地球物理學 J. Morlet 在分析地震波的局部性質時，發現傳統的傅利葉轉換，難以達到其要求，因此引進小波概念於信號分析中，對信號進行分解。1988 年 Daubechies 建構了具有正交性 (Orthonormal) 及緊支集 (Compactly Supported)，如此小波轉換的系統理論得到了初步建立[24][25]。

2.3.3 濾波器組理論：

小波轉換使用將輸入訊號通過濾波器組的方式達到訊號的分解與合成。濾波器乃是信號處理系統代稱，濾波器的功能就是選擇性地處理信號。而經由濾波器可以將訊號做高頻、低頻的分解，此種將信號分解的過程我們一般稱為分析 (Analysis)。另外一方面，就是藉由濾波器的作用，將所輸入信號的每個頻率成分合成而得到一個輸出信號，此種程序稱為合成 (Synthesis)。一個訊號通過 Analysis Filter 的分解，這個分解後的結果經過 Synthesis Filter 重組產生了另一個新的訊號，假如此新的訊號與原始訊號相同，那麼我們稱這樣的 Analysis Filter 與 Synthesis Filter 能達到訊號的完美重建 (Perfect Reconstruction)，而我們稱這樣

的 Analysis Filter 與 Synthesis Filter 為一組濾波器組(Filter Bank)。

(a). 訊號分解



(b). 訊號合成

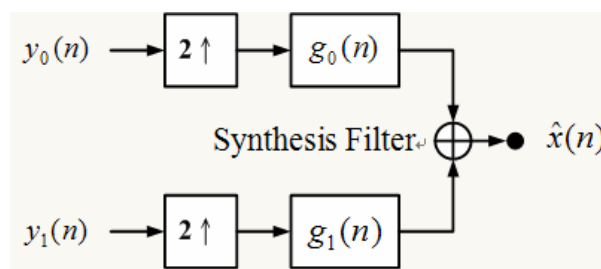


圖 3、一維小波轉換之分解與合成示意圖 (a)訊號分解 (b)訊號合成

上圖 3 (a)中，原始訊號經 $x(n)$ 由 h_0 與 h_1 將低頻及高頻訊號分解出來，其中 h_0 為低通濾波器(low pass filter)，能讓低頻訊號通過； h_1 為高通濾波器(high pass filter)，高頻訊號能通過。由於最後得到的低頻訊號 $y_0(n)$ 與高頻訊號 $y_1(n)$ 之總合必須與原訊號 $x(n)$ 大小相同，因此通過 h_0 與 h_1 之訊號必須進行 down-sample，如此一來使得 $y_0(n)$ 及 $y_1(n)$ 的大小剛好是 $x(n)$ 的一半。圖 3(b)中，當我們利用合成濾波器 $g_0(n)$ 、 $g_1(n)$ 對低頻、高頻訊號進行合成前，必須先提高取樣 up-sample，才能使訊號的合成順利進行。而能夠達成完美重建的濾波器組，必須滿足下面公式：

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0$$

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2$$

2.3.4 小波頻率域特性：

在多重解析小波轉換中，一個特定的訊號被分解成特定的頻段，這些頻段即被稱為子頻道(Sub-band)。這些子頻道分別屬於三個頻區，分別有不同之特性：

低頻區特性：

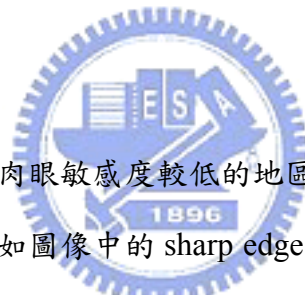
一般而言，人類肉眼的視覺感觀對低頻敏感度較高，如果我們在低頻區加入一些資訊，則轉換回空間域後，影像整個便會被改變。也因此可以知道低頻區是最重要的區域，它的係數實際上代表影像整體的特性、整體的資訊。在空間域的像素中，比較平滑的部份(像素間數值差異不大)。

高頻區特性：

而高頻區則是對於人類肉眼敏感度較低的地區，也就是在空間域中像素與像素之間的變化大的部份，例如圖像中的 sharp edge、黑白相間的部份。一般而言高頻區的資料如果遭受損失或增加時，轉換回空間域後，並不會讓影像本身產生強烈破壞。換句話說，如果影像在頻率域屬於高頻部份受到破壞，則它相對應空間域的數值分佈也很容易受改變。

中頻區特性：

它是介於低頻與高頻之間的區域，一般而言如果在此區域加入額外資訊，並不像低頻區一樣容易對影像本身特性產生影響，同時也不像高頻一樣，不容易保存高頻區域資訊。因此這是一般的數位浮水印技術均選擇中頻區域嵌入浮水印的原因。



2.4 Correlation – based watermark detection

在浮水印系統中，版權的驗證是根據能不能於一張待測影像中測得浮水印的存在(Existence Measurement)而決定的。所謂的 correlation-based watermark detection 就是以原浮水印 W 與待驗證浮水印 W' 進行相關係數運算，最後依這個運算的結果 ρ 與一定之門檻值(Threshold) ρ_T 進行比較，來證明版權、所有權資訊的存在。如果在驗證過程中 $\rho \geq \rho_T$ 表示 W 與 W' 是相同的，則我們可以宣告版權的存在，反之如果 $\rho < \rho_T$ 則表示 W 與 W' 是兩個不相同的浮水印。如圖 4 即是一個典型的 correlation-based watermark detection 流程[26]，以下依序為整個流程做解說。

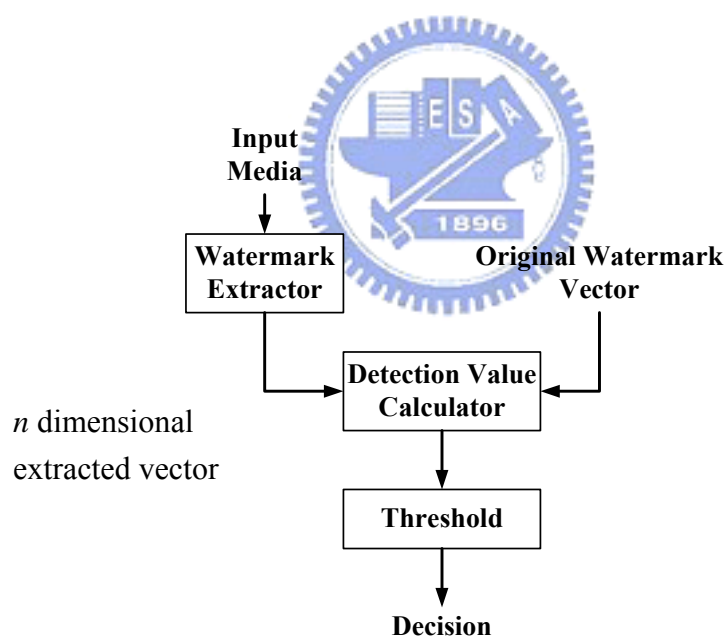


圖 4、基本的浮水印偵測流程

Input Media 一般指的是待測之數位內容，如 video、audio、image....等。Input media 經過 Watermark Extractor 將待測浮水印解出。在[26]中對於 correlation-based watermark detection 提到了幾個重要的前提必須要進行討論。其一，原始浮水印 W 它是一個長度為 n 的向量(single watermark vector)，而且它是一個已知的，事先定義好的向量(Constant, Predefined)。其二，待測浮水印 W' 亦是一個長度為 n 的

向量，不過它的每一個浮水印位元 w'_i 均是無法預測、無法事先猜測的。接下來才將已知向量 W 與待測向量 W' 進行 correlation value 之計算，最後再由 Decision 決定浮水印版權的存在。這其中隱含的意義是在 correlation-based watermark detection 中，watermark extractor 與 detection value calculator 是沒有關係的，因為每一個 w'_i 均是無法預測，detection value calculator 單純是針對兩個 vector 做比較。

Decision 程度決定浮水印的存在(Existence)與否，所以有一個問題是如何選擇適當的 Threshold ρ_T 作為浮水印判別之依據，而將 ρ_T 做為判定之鑑別時誤判的可能性如何。False Positive Probability P_f 指的是在一張未嵌入浮水印之影像中，偵測出浮水印的機率。一個最常使用的方法是先設定我們期望的 False Positive Probability，再由 P_f 推導出在此誤判機率之下的門檻值 ρ_T 。如圖 5 所示：



圖 5、門檻值由期望值判率決定

Detection Value Calculator、False Positive Probability 與 Threshold 之間的關係整理如下所示：

1. 不同的 Detection Value Calculator，在相同的 False Positive Probability 之下，會有不同的 Threshold。
2. 相同的 Detection Value Calculator，因為 False positive probability 推導之 model 不同或逼近法的不同，在相同的 False Positive Probability 之下，會有不同的 Threshold。以 Normalized Correlation 為例，Approximate Gaussian method、Fisher Z-statistic method 與 method in [26]，都是提供計算使用 Normalized

Correlation 當作 Detection Value Calculator，不過因為逼進法使用的 model 之差異，當 false positive probability 固定時，門檻值亦不相同。

3. 相同的 Detection Value Calculator，但假設浮水印的特性不相同，亦影響 False positive probability 估計法。表格 1 整理了三篇以 Normalized Correlation 當 Detection Value Calculator 的參考論文，因為浮水印假設條件不相同，所以其推導之 model 亦不同。

參考論文	浮水印之特性	ρ_T
[26]	Identical, zero-mean, independent, Gaussian distributions.	$\frac{I_{n-2}(T_\alpha)}{2I_{n-2}(\pi/2)}$
[27]	1, -1 sequence	$\sum_{n=\lceil N_w \times (\rho_T + 1) / 2 \rceil}^{N_w} \binom{N_w}{n} 0.5^{N_w}$
[28]	Real number, normalized	$\frac{\int_T^1 (m-1)V_{m-1}(\sqrt{1-x^2})^{m-3} dx}{mV_m}$

表格 1、使用 Normalized Correlation 時，不同的浮水印特性，可以有不同的 false positive probability 估算方法。

第三章 非均勻量化 Super Tree 演算法

本章中，3.1 小節介紹的非均勻量化 Super Tree 浮水印嵌入流程、3.2 節的非均勻量化 Super Tree 浮水印取出流程是由[1]所提出來的，然而在 3.3 節介紹的非均勻量化浮水印技術之缺點中，我們將提出[1]中可能導致解浮水印不精確的問題，並加以討論。

3.1 非均勻量化 Super Tree 浮水印嵌入流程

整個嵌入流程是基於小波轉換後對於 SuperTree 量化的嵌入方法。主要之流程如圖 6 所示，本節將對浮水印嵌入的步驟進行討論。

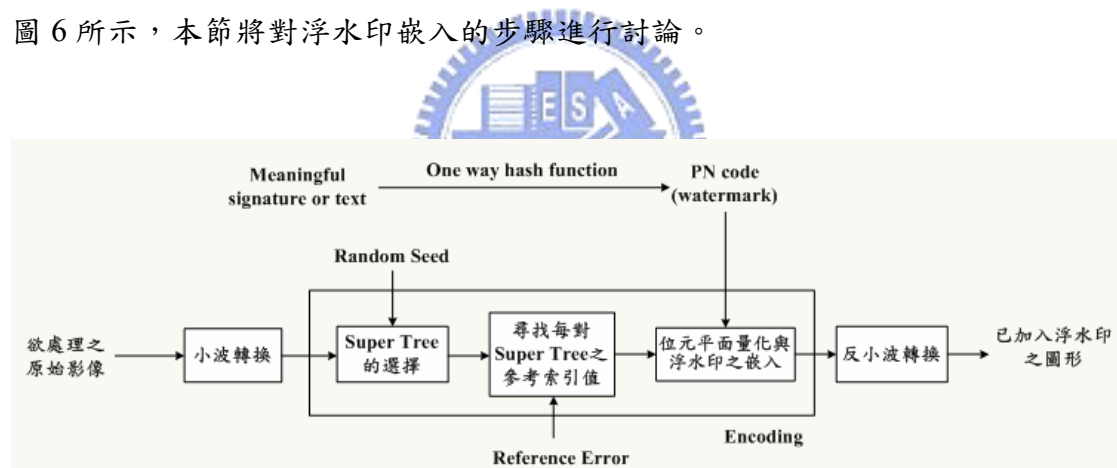


圖 6、非均勻量化 Super Tree 浮水印嵌入流程

3.1.1 浮水印的產生

一般的浮水印可以是圖形化的 Logo，也可以是一串數字序列或文字訊息，而在本文所使用的浮水印為一串 ± 1 的 PN 數字序列(Pseudo Noise)，然而這一串數字序列之產生可以由一個 Logo 或文字訊息，經過一個 one way hash function 轉換至 ± 1 的數字序列。然而本論文實際在模擬浮水印的產生，是採用一個亂數種子(seed) s 來進行模擬，實際產生使用的演算法如下：

```

Srand(s);
For (i=0 ; i<watermark_length ; i++) {
    If (rand() % 2 == 1) Watermark[i] = 1
    Else Watermark[i] = -1
}

```

3.1.2 小波轉換：

小波轉換能將原始影像由空間域(Spatial domain)轉換至頻率域(Frequency domain)，而小波轉換的 Subband coding 能夠讓我們選擇最適當的子頻道進行浮水印嵌入。由文獻探討中可以知道，如果將版權資訊加在低頻地區則將使整體影像受到破壞(即使加入的量不多)，而版權資訊加入高頻的子頻道則使得浮水印容易因為外在攻擊而遭到移除，使浮水印不具有強韌性。因此最好的方法是選擇中頻的子頻道嵌入。在此之前，我們將浮水印進行四層小波轉換，如下圖所示，圖 7(a)是將 Host image 以小波轉換進行四層分解之示意圖，圖 7(b)則是圖 7(a)所有子頻道所在之相對位置。

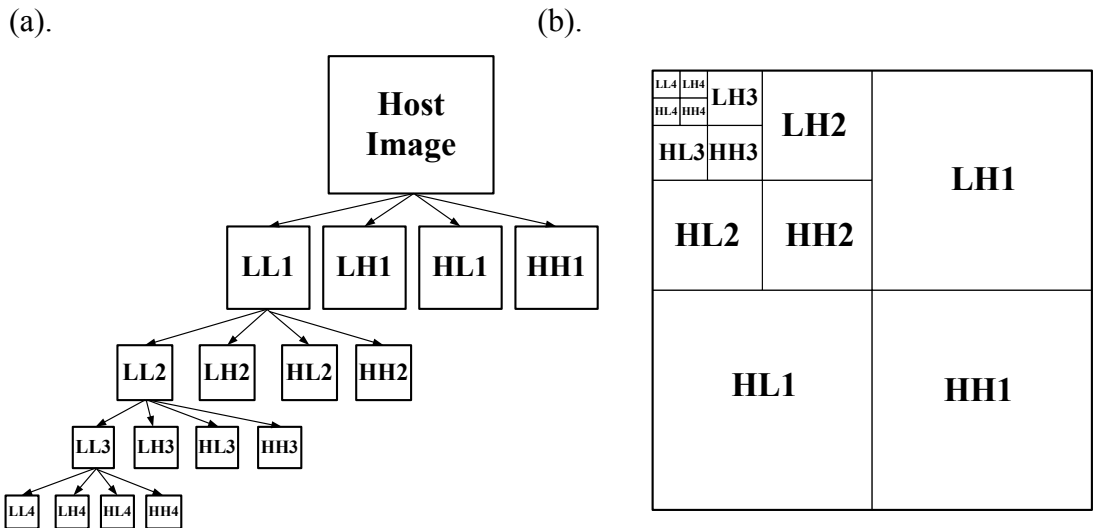


圖 7、小波轉換示意圖 (a)將 host image 進行小波分解，共分解成四個 level，每個 level 包含了 LL,LH HL 與 HH 四個子頻道 (b)經(a)分解後，實際上每個子頻道於圖形中相對應之位置

3.1.3 Super Tree 的選擇：

浮水印的嵌入，是以量化 Super Tree 的方式進行，因此首先我們對於 wavelet tree、super tree 及 super tree pair 等相關的專有名詞做定義。再討論如何進行 Super Tree 的選擇。

Wavelet Tree：

在圖 8(a)中，我們選擇小波頻率域中的 Level 2、Level 3 及 Level 4 進行研究，wavelet tree 選擇的方法，是先由 Level 4 的 band 1、2、3 中任取一個係數出來。此係數會有 4 個 children 位於 Level 3 中，位於 Level 3 的這 4 個係數分別各有 4 個 children 於 Level 2 中。Wavelet tree 即是將 Level 4 的一個係數、Level 3 的四個係數及 Level 2 的 16 個係數集合起來的 21 個小波頻率域中頻係數，如圖 8(b)所示。一張圖形的小波樹個數其實就等於 Level 4 中 band 1、2、3 的總係數個數。因此可以得知小波樹總個數為：



$$\text{Level 4 的 band 個數} \times \text{每個 band 中係數個數} = 3 \times 32^2 = 3072 \text{ 棵。}$$

Super Tree：

在 3072 棵小波樹中任取兩棵小波樹，形成一組 Super Tree T_i ， $i=1,2,\dots,1536$ ，如圖 8(c)所示。在選取小波樹時，已被選取過的樹將不再被選取，因此 3072 棵最後變成了 1536 棵 Super Trees。從小波樹中選取兩棵小波樹時，同樣地我們使用一個亂數當作種子，因此我們最後得到的 Super Tree 能達到安全的目的，惡意攻擊者無法在有限時間內猜到 Super Tree 的組合以取出浮水印資訊。

Super Tree Pair

一個 Super Tree Pair p_i 是由 1536 棵 Super Tree 中的兩棵樹 (T_{2i-1}, T_{2i}) 所組成的，其中

$i = 1, 2, \dots, 768$ 。Super Tree Pair p_i 的目的，是為決定浮水印第 i 個位元 w_i 的嵌入，也就是 w_i 嵌入於 p_i 中， $i = 1, 2, \dots, 768$ 。然而嵌入的方式是透過對 (T_{2i-1}, T_{2i}) 的其中一棵 tree 做量化來達成。如果 $w_i = -1$ 則我們對 T_{2i-1} 進行 uniform quantization，如果 $w_i = 1$ 則是對於 T_{2i} 進行 uniform quantization。因此我們可以知道浮水印之長度必須小於等於 Super Tree pair 之個數，如此每一個浮水印位元 w_i 才能有相對應之 Super Tree pair p_i 。

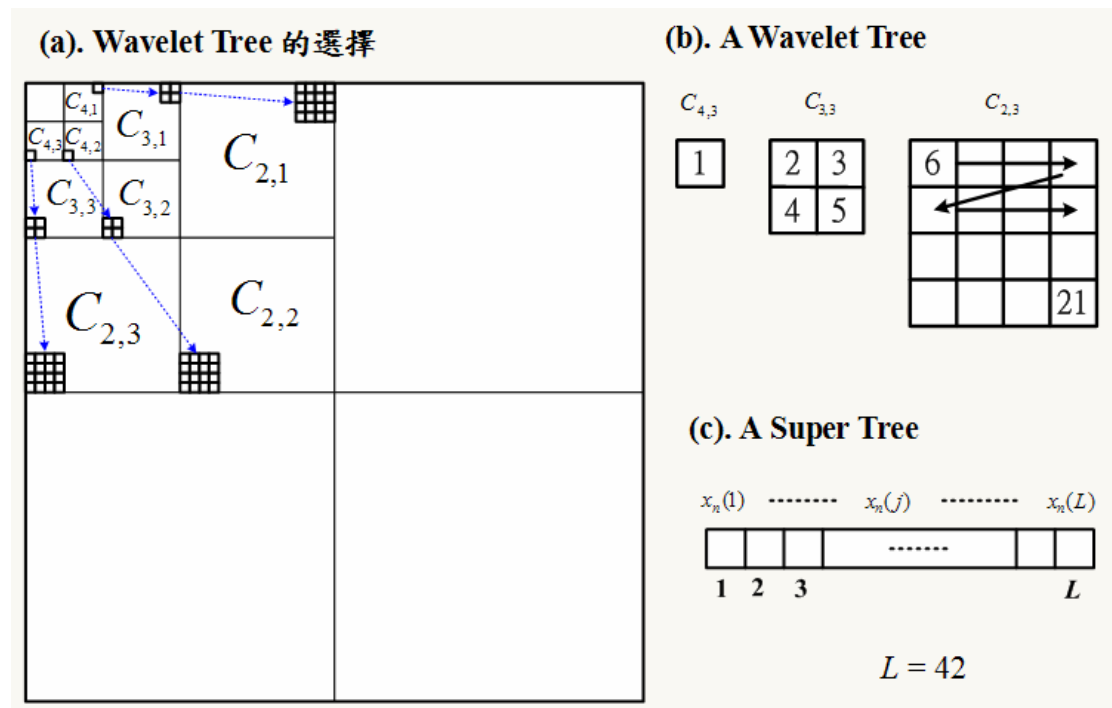


圖 8、Wavelet Tree、Super Tree 之定義 (a)Wavelet tree 的選擇 (b)A Wavelet Tree (c)A Super Tree

3.1.4 Super Tree Bitplane

進行量化前，每一個在 Super Tree T_i 的係數 $x_i(j)$, $j = 1, 2, \dots, L$ 都將以二進位的形式表現，也因此形成一個位元平面(bitplane)。如圖 9 所示，位元平面中的 Least-Significant Bitplane(LSB)為 2^0 位於 bitplane 底部，Most-Significant Bitplane (MSB)定義為 2^p 位在上方，此外同時紀錄了正負位元(Sign bit)。因此在

整個 bitplane 中共有 $L(p+1)$ 個位元。

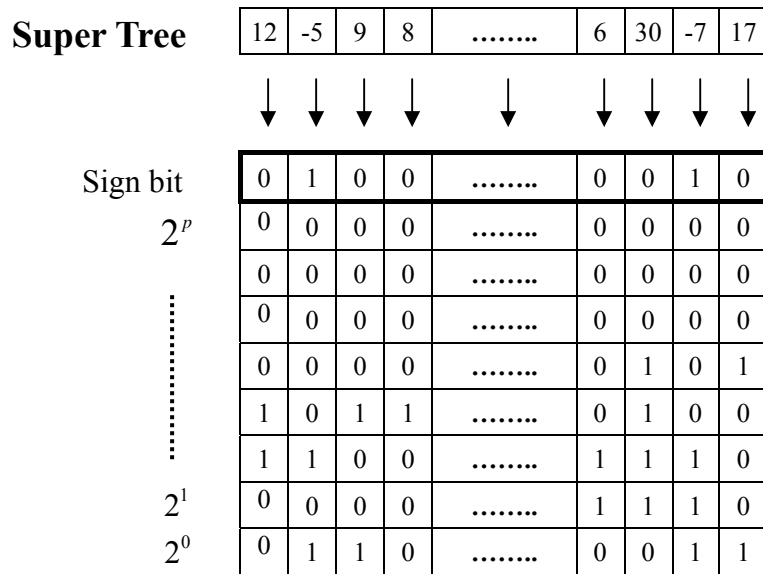


圖 9、Super Tree 轉換為二位平面位元之範例



3.1.5 位元平面量化

浮水印的量化是根據一個量化前與量化後誤差值的累積而來，量化的進行由 bitplane 的右下角開始，先由右至左再由下到上的順序進行誤差累積。而參考索引值 q_n (reference index) 所代表的意義為已量化多少個 bitplane 中的位元。如圖 10 所示， q_n 所表示的是下方斜線部份之 bit 個數。假設 q_n 所在座標為 (a_n, b_n) ，在 q_n 左側的每個 Super Tree 係數之 quantization step 為 2^{a_n} (包含 $j=b_n$ 時)，位於 q_n 右側的每個 Super Tree 係數之 quantization step 為 2^{a_n+1} 。每個 Super Tree $x_n(j)$ 係數在量化後與量化前產生之差值定義為 $e_n(j) = Q[x_n(j)]_{q_n} - x_n(j)$ ，其中

$$Q[x_n(j)] = \begin{cases} \text{round}(x_n(j))_{a_n} & , j \leq b_n \\ \text{round}(x_n(j))_{a_n+1} & , \text{otherwise} \end{cases}$$

而總共量化的量 $\varepsilon_n(q_n) = \sum_{j=1}^L |e_n(j)|$ 。

$\varepsilon_n(q_n)$ 所代表的意義為當量化了 bitplane 中 q_n 個係數後，對於 Super Tree 產生的資料損失量為 $\varepsilon_n(q_n)$ 。 $\varepsilon_n(q_n)$ 愈大(量化的量愈大)對圖形品質影響愈大，因為它代表對 Super Tree 量化的量。

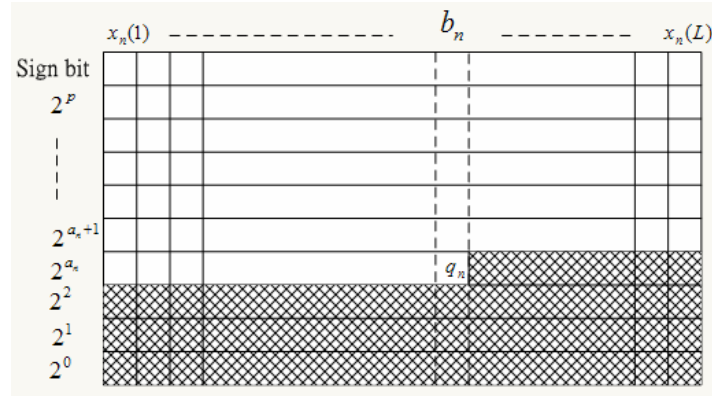


圖 10、 q_n 決定一棵 Super Tree 量化的量。

3.1.6 浮水印的嵌入

進行真正量化前，根據一個參考誤差 ε (reference error)，來大略估計未來要量化的量。 ε 愈大量化的量可能愈大，對圖形的品質會造成影響；但 ε 太小時會使強韌性不足。我們試圖找到一個 q_n ，同時滿足 $\varepsilon_{2^{i-1}}(q_n) \geq \varepsilon$ 且 $\varepsilon_{2^i}(q_n) \geq \varepsilon$ 且 $q_n \leq q_{\max}$ ，這個 q_n 便決定 $p_i(T_{2^{i-1}}, T_{2^i})$ 量化的量。浮水印位元 w_i 之嵌入是根據 w_i 的值對 $p_i(T_{2^{i-1}}, T_{2^i})$ 中某一棵 tree 量化而達成。如果 $w_i = -1$ 則對 $T_{2^{i-1}}$ 以 q_n 量化，反之對 T_{2^i} 進行量化。

3.2 非均勻量化 Super Tree 浮水印取出流程

對於一張待測影像，我們想要驗證其中是否含有特定的版權證明或其他資訊時，則必需利用浮水印萃取演算法來達成，流程如圖 11 所示。一開始同樣地將欲檢驗之影像經小波分解至頻率域，接下來以 encoding 使用的 key 尋找出正確的 Super Tree Pair $p'_i (i = 1, 2, \dots, 768)$ 組合。由於每個浮水印 w_i 以量化方式藏於 p'_i 中，所以解浮水印演算法之重點在於如何從 $p'_i(T'_{2i-1}, T'_{2i})$ 中找出何者被量化過，以推測浮水印位元可能的值。

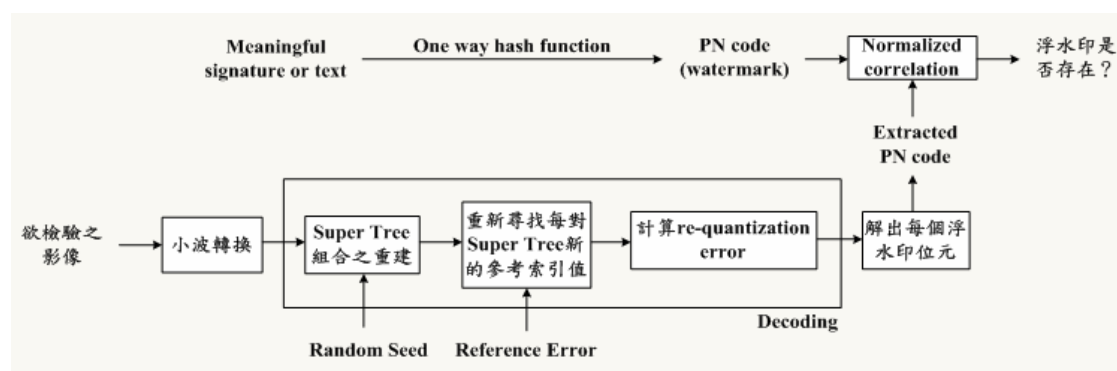


圖 11、非均勻量化 Super Tree 浮水印取出流程

3.2.1 Re-quantization and watermark extraction

在解浮水印時同樣地，我們的目的是找到一個最小的 q'_n ，使找到的 q'_n 同時滿足 $\varepsilon'_{2i-1}(q'_n) \geq \varepsilon$ 或 $\varepsilon'_{2i}(q'_n) \geq \varepsilon$ ，並且 $q'_n \leq q_{\max}$ 。接下來根據 q'_n 分別找出 (T'_{2i-1}, T'_{2i}) 在 decoding 的 re-quantization error e'_{2i-1} 與 e'_{2i} 。完整的 re-quantization error 定義如下：

$$e'_l(j) = x'_l(j) - Q[x'_l(j)]_{q'_n}$$

，其中 $l = 2i-1, 2i$ ， $x'_l(j)$ 表示在第 l 個 Super Tree 中的第 j 個係數值。

$Q[x'_l(j)]_{q'_n}$ 表示依據 q'_n 對 $x'_l(j)$ 重新量化後的值。假設 q'_n 所在座標為 (a'_n, b'_n) ，在 q'_n

左側的每個 Super Tree 係數之 re-quantization step Δ'_l 為 2^{a_n} (包含 $j=b'_n$ 時)，位於 q_n 右側的每個 Super Tree 係數之 re-quantization step Δ'_l 為 2^{a_n+1} 。

當 (T'_{2i-1}, T'_{2i}) 中的每個係數誤差值 $e'_{2i-1}(j)$ 與 $e'_{2i}(j)$ 被找到後，[1] 定義了 cdf of the magnitude of normalized re-quantization errors，定義如下：

$$f(y) = \text{prob} \left[\left| \frac{e'_l(j)}{\Delta_l(j)} \right| < y \right]$$

$f(y)$ 代表的意義是，Super Tree l 的 re-quantization error magnitude 大小 $|e'_l(j)|$ 與 quantization step $\Delta_l(j)$ 之比例，高於門檻值 y 的機率。假設 T'_{2i-1} 滿足 $f(y)$ 的個數共有 N_{2i-1} 個，而 T'_{2i} 中滿足 $f(y)$ 的個數共有 N_{2i} 個。浮水印 w'_i 依下面判斷公式取出：

$$w'_i = \begin{cases} -1, & \text{if } N_{2i-1} > N_{2i} \\ 1, & \text{otherwise.} \end{cases}$$

當所有浮水印位元 w'_i 由 p_i 中取出後，解出之浮水印 W' 與原始浮水印 W 進行版權之判定。

3.2.2 浮水印之驗證：

參考論文 [1] 中浮水印的驗證方法採用 correlation-based watermark detection，針對由演算法中解出之浮水印 W' 與原始浮水印 W 進行相關性檢驗，檢驗方法則採用 Normalized Correlation 做為 Correlation Value Calculator。

Normalized Correlation 完整的定義如下：

$$\rho(W, W') = \frac{\sum_{m=1}^{N_w} w_m w'_m}{\sqrt{\sum_{m=1}^{N_w} w_m^2 \sum_{m=1}^{N_w} w'_m^2}}$$

上式中， N_w 為浮水印之長度， w_m 與 w'_m 分別是浮水印 W 與 W' 序列中的第 m 個浮水印位元。經計算後所得到的相關係數 ρ 與我們設定的一個 Correlation Threshold ρ_T 進行比較，若 $\rho \geq \rho_T$ 則我們便可以宣稱浮水印 W 與 W' 是相同的，反之則 W 與 W' 並不相同。然而 ρ_T 之所以能宣稱浮水印的存在，是由於它建立在一定的誤判情形之下，有關如何設定 ρ_T 我們將在 4.5 節做詳細推導及討論。



3.3 非均量化 Super Tree 浮水印技術之缺點

在解浮水印的整個過程中，根據圖 6 可知我們將待測的圖形經過小波轉換，接下來利用與 encode 相同的亂數種子，找到在 encode 中的 super tree 順序，以利 decode 中能正確找出浮水印。而在 decode 找到的 super tree 此時以 T'_l 來表示，其中 l 的範圍由 1 到 1536，並將它們配對成 i 個 super tree pair (T'_{2i-1}, T'_{2i}) , $i=1,2,\dots,768$ 。

在參考論文[1]中提到的解浮水印方法，主要是對於每一個 super tree pair (T'_{2i-1}, T'_{2i}) ，在一定的 reference error 之下，尋找每個 pair 的 re-quantization index q'_n 。找 q'_n 的做法與 encode 相似，預測 T'_{2i-1} 誤差的累積量或 T'_{2i} 誤差的累積量達到 reference error 時停止，進而發現 q'_n 。一旦有了 q'_n ，那麼隨即對 T'_{2i-1} 與 T'_{2i} 進行 requantized error 之統計。理論上來說 requantized error $(\varepsilon_{2i-1}(q'_n), \varepsilon_{2i}(q'_n))$ 一定會有一個的數值大小的分佈是比較小的，如此便可以猜測何者在 encode 中被量化過，因此藏在 (T'_{2i-1}, T'_{2i}) 中的浮水印位元便被找出。不過此一方法有隱含了兩個潛在的問題，使得浮水印位元的尋找出現判定精確度不足，如下說明之：

3.3.1 Case 1 : (當 $q'_n > q_n$ 時發生)

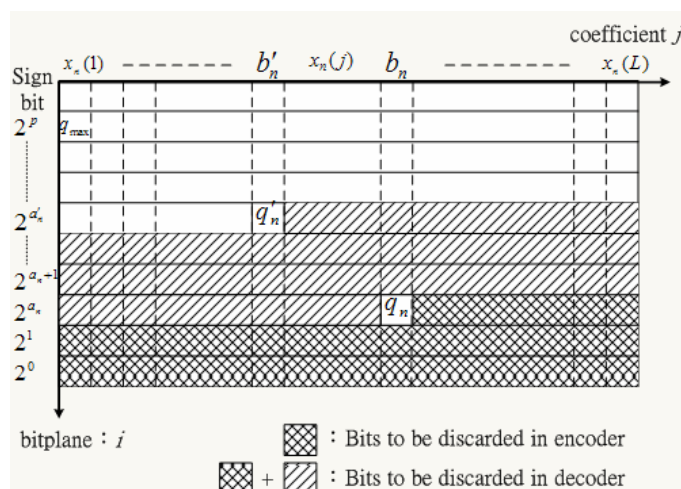


圖 12、decoding 找到的 q'_n 大於 encoding 量化使用 q_n 之示意圖。

一般而言演算法期望 decode 找到的 q'_n 能夠小於等於 q_n 是最理想的情形，此時能完美找出 (T'_{2i-1}, T'_{2i}) 何者被量化過。我們以同樣一棵 Super Tree 在 encoding 與 decoding 中的 quantization 與 re-quantization 所量化的量來看，如圖 12 所示，當 $q'_n > q_n$ 時， q'_n 與 q_n 中間存在的區域便是誤差的來源。

會造成這種情況的原因如下圖所分析：

(a) In encoding

T_{2i-1} (Quantized)					T_{2i} (Un-quantized)				
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0
0	1	1	0	0	1	0	1	0
1	0	1	1	1	0	0	1	0
1	0	0	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0
0	0	0	0	1	0	0	0	0

(b) In decoding – Re-quantization 當 $q'_n = q_n$ 時，兩棵 Tree 累積之誤差量均未達到 reference error

T'_{2i-1}					T'_{2i}				
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0
0	1	1	0	0	1	0	1	0
1	0	1	1	1	0	0	1	0
1	0	0	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c) In decoding $-T'_{2i-1}$ 或 T'_{2i} 累積之誤差達到 reference error，Re-quantization 完成

T'_{2i-1}					T'_{2i}				
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0
0	1	1	0	0	1	0	0	1
1	0	1	1	1	0	0	0	1
1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

圖 13、Case 1 無法正確判斷浮水印位元之範例 (a) p_i in encoding (b) p'_i in decoding (when $q'_n = q_n$) (c) Re-quantization 完成 in decoding

上圖 13(a)我們對 T_{2i-1} 與 T_{2i} 進行量化，假設依據浮水印位元，我們對 T_{2i-1} 進行量化，則 Super Tree T_{2i-1} 中灰色區域部份即被量化為空值。圖 13(b)表示，在 Decoding 時浮水偵測浮水印 w_i ，所以對於 T'_{2i-1} 與 T'_{2i} 進行 re-quantization 累積誤差量，試圖找出何者被曾經被量化過。當 $q'_n = q_n$ 時，發現 T'_{2i-1} 或 T'_{2i} 累積的量均未達到我們所設定的 reference error，此時演算法仍然持續進行誤差量的累積。圖 13(c)表示當 T'_{2i-1} 或 T'_{2i} 其中一棵樹誤差累積量達到 reference error 時，演算法停止 re-quantization，不過此時 $q'_n > q_n$ 。

實際誤判的產生：

圖 14 中間灰色區域，顯示出圖 13(c)中 T'_{2i-1} 的量化區域與圖 13(a)的量化區域的差距，也就是可能產生誤差的 region，我們定義每一個量化誤差之差值 $\varepsilon'_{2i-1}(j) - \varepsilon_{2i-1}(j)$ ， $j = 1, 2, \dots, 42$ 。如果 $\varepsilon'_{2i-1}(j) - \varepsilon_{2i-1}(j)$ 為 0，則表示第 j 位元不會

有誤判情形，但如果 $\varepsilon'_{2i-1}(j) - \varepsilon_{2i-1}(j)$ 不為 0，則表示第 j 位元會對判定之準確度造成影響。

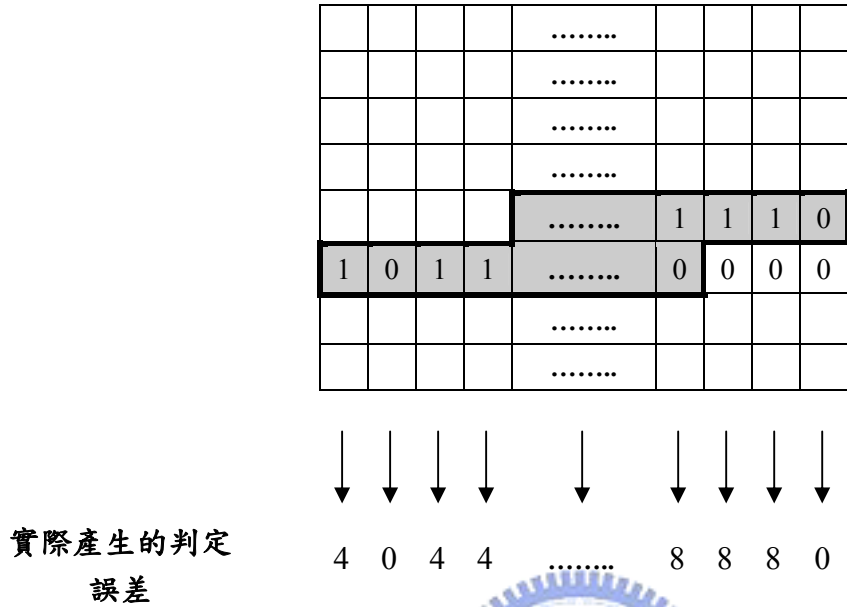


圖 14、圖 13 中實際產生判定之誤差



3.3.2 Case 2 : (當 $q'_n = q_{\max}$ 時發生)

在談論 case 2 的問題之前，我們先對 Magnitude of a Super Tree 做一個簡單的定義，一個 Super Tree 的 magnitude M_i 指的是此 Super Tree T_i 中所有系數絕對

值之合，也就是 $M_i = \sum_{j=1}^L |e_i(j)|$ ，其代表的義意是 T_i 中所隱含之資訊量的大小。Case

2 的發生有兩個主因。其一為 encode 時， (T_{2i-1}, T_{2i}) 中有一棵 Super Tree 的 magnitude 小於我們所訂定之參考誤差 ε ，使得 quantization step 達到最大值(做法與 3.1.5 相同)。其二為 decode 時 (T'_{2i-1}, T'_{2i}) 之 magnitude 均小於參考誤差。圖 15 為一個例子。

(a). In encoding – 在進行量化之前

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	1	0	1	1	1	0

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0
0	1	0	1	1	0	1	1
0	0	1	0	1	1	0	0

(b). In encoding – 進行量化之後

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

 q_n

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0
0	1	0	1	1	0	1	1
0	0	1	0	1	1	0	0

 q_n

圖 15、(a)如果 T_{2i-1} 或 T_{2i} 的 M_{2i-1} 或 M_{2i} 有任一個小於我們所設定之 reference

error，則 q_n 便會達到 q_{max} (b)對 T_{2i-1} 量化後的結果

我們定義在 decoding 時 Super Tree T_l 的 quantization error $e'_l(j) = x'_l(j) - Q[x'_l(j)]_{q'_l}$ ， $\Delta'_l = \Delta'_{max}$ ，其中 $l = 2i-1, 2i$ 。在 decoding 時，如果 T'_{2i-1} 與 T'_{2i} 的 M_{2i-1} 與 M_{2i} 均小於我們設定的 magnitude，則同樣地在 Re-quantization 時量化索引值 q'_l 將達到 q_{max} 。當判別浮水印時參考論文[1]使用 $|e'_l(j)/\Delta'_l(j)| < \varepsilon$ 來判定，很有可能造成 (T'_{2i-1}, T'_{2i}) 符合 $|e'_l(j)/\Delta'_l(j)| < \varepsilon$ 之個數相同，而誤判因而產生。這個結果的發生主要是由於 (T'_{2i-1}, T'_{2i}) 使用的 quantization step Δ'_l 已達到最大值(Maximum quantization step)，且 re-quantization error 均非常小所導致的。圖 16

攻擊後，誤判之情形會更為嚴重。

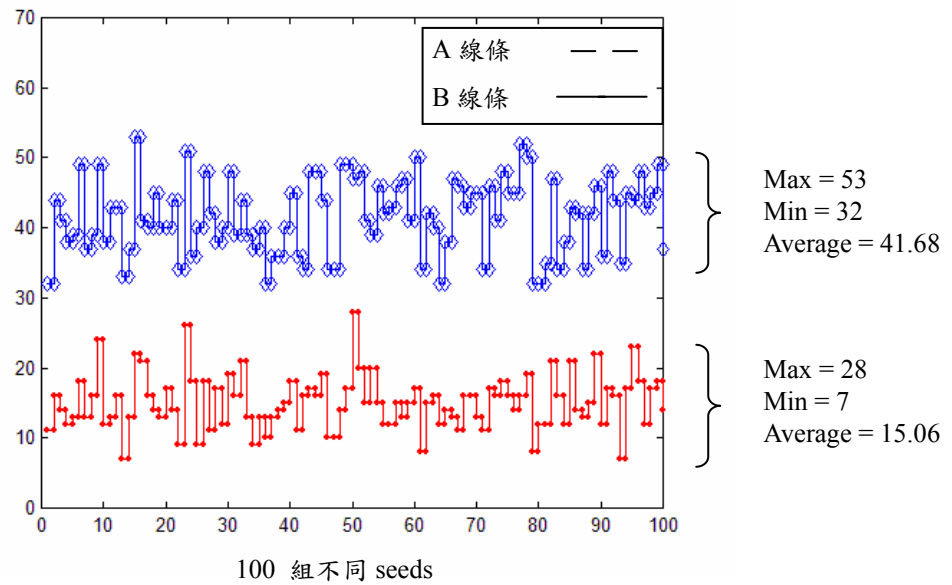


圖 17、以 100 組 seeds 為測試，上方 A 線條表示 $q'_n = q_{\max}$ 發生次數，中間 B 線條表示在 $q'_n = q_{\max}$ 之下，浮水印位元發生誤判次數。



第四章 Deadzone Scalar Quantization

Watermarking Algorithm

本章將提出浮水印演算法之改進方式，在 4.1 節中提出本文使用的浮水印嵌入方法，整個嵌入流程與參考論文[1]的方法相似，而最小量化階層(minimum quantization step)及均勻量化(uniform quantization)的使用將幫助 4.4 節浮水印取出。4.2 及 4.3 節中討論了可能影響浮水印解出之因素，最後 4.5 節則將完整演算法列出。

4.1 浮水印之嵌入方法

本文使用的浮水印嵌入方法，其架構與 3.1 節中類似。嵌入程序開始一張影像經過小波分解後，以一 key 選擇 Super Tree $T_i, i = 1, 2, \dots, 1536$ ，並將它們分配組成 Super Tree Pair $p_i, i = 1, 2, \dots, 768$ 。浮水印使用 ± 1 數字序列，每個浮水印位元 w_i 透過均勻量化 $p_i(T_{2i-1}, T_{2i})$ 中的一棵 Super Tree 來達成。

本文嵌入方法與[1]之不同點在於 Super Tree 的 bitplane 量化。如圖 18 所示，在一個 Super Tree bitplane 中，假設 reference error q_n 所在的縱座標 2^{a_n} ，那麼我們在量化時，便以 2^{a_n+1} 當作量化階層，也就是在 bitplane 中總共量化的位元個數為 $L(a_n + 1)$ 個。

4.1.1 Minimum quantization step 之設計

我們定義一個 Minimum quantization step Δ_{\min} ，如果依據 q_n 找出的量化階層

$2^{a_n+1} < \Delta_{\min}$ 時，則我們將量化階層強制設為 Δ_{\min} ，這樣的作法可以確保每一棵被量化過的 Super Tree，量化階層至少都是 Δ_{\min} ，如圖 18 所示。另外，當 $q_n = q_{\max}$ 時，也就是 Super Tree Pair $p_i(T_{2i-1}, T_{2i})$ 中 Tree 的 Magnitude 均太小時，我們也把量化階層設為 Δ_{\min} ，這樣一來就能解決 3.3.2 節中 Case 2 產生的問題。

每個 Super Tree $x_i(j)$ 係數在量化後與量化前產生之差值定義為

$e_i(j) = Q[x_i(j)]_{q_n} - x_i(j)$ ，其中 $Q[x_i(j)] = \text{round}(x_i(j))_{a_n+1}$ ，而總共量化的量為

$$\varepsilon_i(q_n) = \sum_{j=1}^L |e_i(j)|。$$

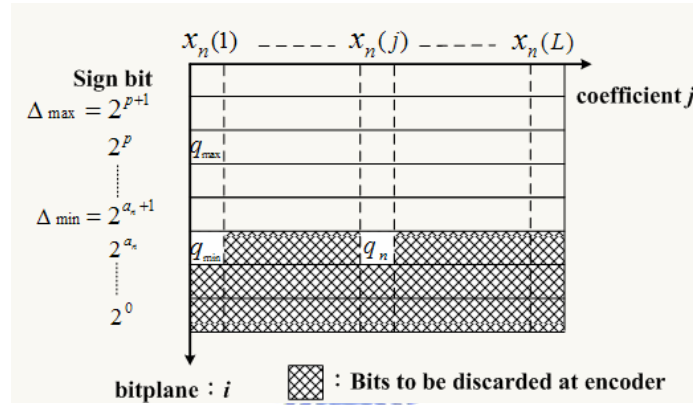


圖 18、本文於 encoding 採用的位元平面量化方法

實際對於 $p_i(T_{2i-1}, T_{2i})$ 量化前，仍然經由參考誤差 ε (reference error)，來大略估計未來要量化的量。我們試圖找到一個 q_n ，同時滿足 ($\varepsilon_{2i-1}(q_n) \geq \varepsilon$ 或 $\varepsilon_{2i}(q_n) \geq \varepsilon$) 且 $q_n \leq q_{\max}$ ，這個 q_n 便決定 $p_i(T_{2i-1}, T_{2i})$ 量化的量。找尋 q_n 的條件式與[1]之不同點在於只要 (T_{2i-1}, T_{2i}) 中任一棵 Super Tree 誤差量的累積 $\varepsilon_i(q_n)$ (其中 $l = 2i - 1, 2i$) 大於 ε 就決定 uniform quantization step size 為 2^{a_n+1} 。如果與[1]同樣使用條件式” $\varepsilon_{2i-1}(q_n) \geq \varepsilon$ 且 $\varepsilon_{2i}(q_n) \geq \varepsilon$ 且 $q_n \leq q_{\max}$ ”當作尋找 quantization step 之條件式，則會使得 uniform quantization step 值太大。最後浮水印位元 w_i 之嵌入是根據 w_i 的值對 $p_i(T_{2i-1}, T_{2i})$ 中某一棵 tree 量化而達成。如果 $w_i = -1$ 則對 T_{2i-1} 以 q_n 量化，

反之對 T_{2i} 進行量化。

4.2 誤差對於解浮水印過程之影響

解浮水印過程產生非預期之誤差：

假設我們在 encoding 時，依浮水印位元 w_i 對 Super Tree pair $p_i(T_{2i-1}, T_{2i})$ 中某一棵 tree 進行均勻量化，則在 decoding 時，我們找到這個在 encoding 中相同順序的 Super Tree Pair $p'_i(T'_{2i-1}, T'_{2i})$ ，在理想的狀況下(沒有任何能量增加或損失的情況下)此兩棵 Super Tree 應該會與 (T_{2i+1}, T_{2i}) 是一模一樣的，也就是 $T_{2i+1} = T'_{2i+1}$ 且 $T_{2i} = T'_{2i}$ 。不過實際上 (T'_{2i+1}, T'_{2i}) 與 (T_{2i+1}, T_{2i}) 並不會相同。

(a) A quantized tree T_i in encoding (b) the same quantized tree T'_i in decoding

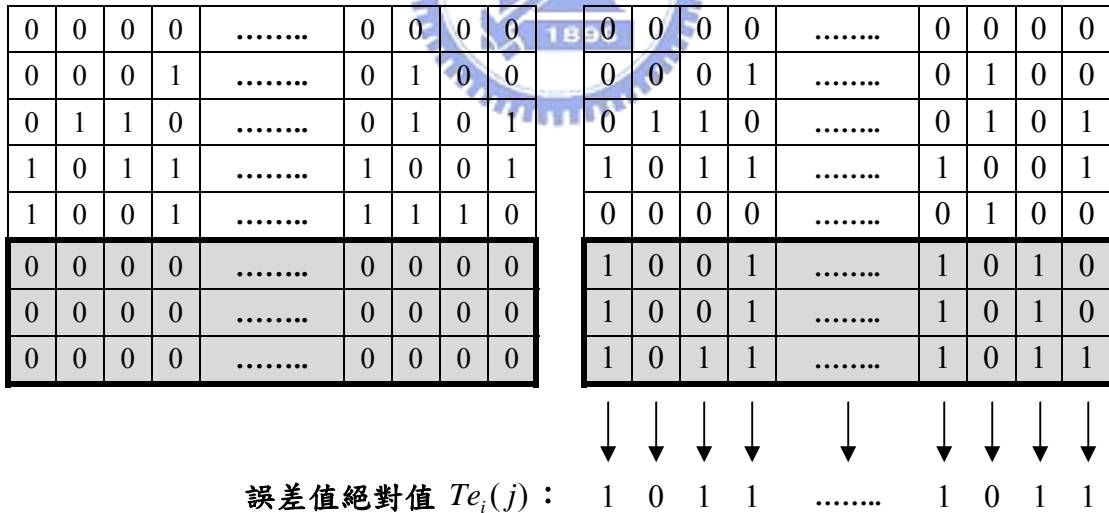


圖 19、 $Te_i(j)$ 誤差之產生範例

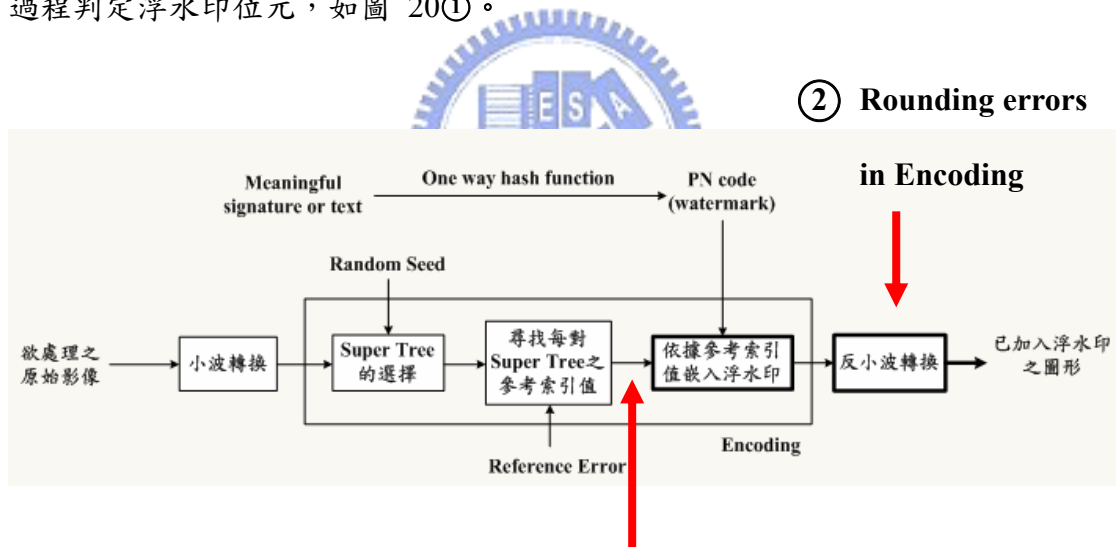
以圖 19 的例子來說，原本在 encoding 中 T_i 被均勻量化，理論上在 decoding 時 T'_i 會與 T_i 一模一樣，但實際上卻不然。圖 19 (b) 為 T'_i 可能的數值分佈，最明顯的差異就是圖 19 (b) 中間灰色區域。假設我們定義 T_i 與 T'_i 中每個係數差之

絕對值為 $Te_i(j) = |x_i(j) - x'_i(j)|$, $i = 1, 2, \dots, 1536$, $j = 1, 2, \dots, 42$ 。在圖 19 (b)最底下則是我們所得到之 $Te_i(j)$ 。 $Te_i(j)$ 如何產生，以及對於浮水印判定之影響性我們將在下一節中詳細說明之。

解浮水印過程中產生 $Te_i(j)$ 誤差之原因，主要包括以下三點：

1. Quantization in Encoding

在浮水印的嵌入演算法中，一個浮水印位元 w_i 的嵌入方式是根據 w_i 的值來對 (T'_{2i-1}, T'_{2i}) 的其中一棵 Super Tree 進行量化。量化的結果使得其中一棵 Super Tree 的能量產生損失，也就是因為這個能量的損失，使得我們能在 decoding 的過程判定浮水印位元，如圖 20①。



① Quantization in Encoding

圖 20、Quantization and rounding errors in encoding

2. Rounding errors in Encoding

當所有浮水印位元 w_i 都經由量化方式加入 Super Tree pair p_i 後，影像經由反小波轉換至空間域，此時影像中的所有 pixel 的數值仍然是浮點數型態，這是因為所有在小波頻率域的數值都是使用浮點數進行處理。由於空間域中的每個像素均為正整數，因此我們將這些浮點數的 pixel 的數值，經由四捨五入轉換至正整

數，以方便影像之儲存，如圖 20②。在這個步驟中，針對每個浮點數四捨五入的運算對於影像的每個像素之能量有稍許的變化，有些像素之能量增加，有些像素能量減少，這是在 decoding 時重建的每一棵 Super Tree 與 encoding 中量化過 Super Tree 在數值上有稍許不同之地方。

3. 對於加入浮水印之破壞

浮水印加入浮水印後，使用者接下來便使用此影像。當使用者使用影像時，很可能因為網路傳輸資料遺失、影像壓縮等過程引起對於影像的非故意破壞。此外惡意攻擊者可能經由高壓縮比的影像壓縮(如 JPEG、JPEG2000...等)、影像內容之故意竄改、Collusion Attack...等針對影像內容或版權資訊惡意破壞攻擊。然而這些對於影像之非故意或惡意的影像破壞，已增加或減少影像本身之資訊，將導致 decoding 過程中，以固定 Key 建立之 Super Tree 係數產生變化。如果對於影像破壞嚴重，則 Super Tree 係數產生變化強烈，這將使得演算法不容易解出每個浮水印位元。反之如果影像被破壞的情形不嚴重，那麼演算法將容易正確地解出每個位元。本文的 4.4 節演算法的建立將有效地抵抗各種破壞，並提升判定之精準度。

4. Filter Bank 影響

小波轉換中，Filter Bank 扮演一個將訊號分解的角色，它直接影響到 Super Tree 數值之大小與分佈，我們將在 4.3 節詳細討論 Filter Bank 的影響性。

4.3 Filter Bank 之選擇

對於 4.2 節所提到的 $Te_i(j)$ 誤差，還有一個重要的影響因素，就是小波轉換所使用的小波轉換濾波器組(Filter Bank)。對於量化一棵 Super Tree T_i 而言，如果 Encoding 過程中量化的量固定、使用 Rounding errors 於浮點數轉換成整數且加過浮水印之影像不受外在攻擊的話，使用不同小波轉換濾波器組時， $Te_i(j)$ 的大小並不會相同。一個比較好的 filter bank 它能減少 $Te_i(j)$ 誤差量，幫助每個浮水印位元 w'_i 之解出。然而在本節中，我們將提出對於 Uniform deadzone 解浮水印時，選用 Filter bank 之原則，並分別於 4.3.1 節與 4.3.2 節討論。

4.3.1 Filter Bank 對 Super Tree 係數 Magnitude 大小之影響。

在 Encoding 過程中，一張影像經由小波轉換由空間域轉換至頻率域，在這個過程中 Filter Bank 與影像資料的 convolution 幫助達成訊號分解之目的。因為 Filter Bank 的不同，轉換分解後的頻率域係數特性也跟著不同，有的 Filter Bank 產生的頻率域係數 Magnitude 高，有些 Filter Bank 產生比較低的頻率域係數值。因此不同的 Filter Bank 會使 Super Tree 係數的 Magnitude 產生差異。

如果一組 Filter Bank 普遍使 Super Tree Magnitude 偏低，則在量化 Super Tree 時將使得量化的量變少。量化的量少時，實際上浮水印之強韌性便不足。量化的量較大時，浮水印之強度也就愈強。也就是說使 Super Tree Magnitude 偏低之 Filter Bank 不適合於此方法。

因此我們定訂了一個 CDF of Super Tree Magnitude 以檢測 Super Tree Magnitude 大小的方式，評量哪些 Filter Bank 不適合於此方法使用。評估的方法

的使用在 encoding 中，原始影像經過小波轉換，且建立完成 1536 組 Super Tree 時，針對 1536 組 Super Tree Magnitude 進行統計計算。CDF of Super Tree Magnitude 其定義如下：

$$f_1(m) = \text{probability}[M_i \leq m]$$

，其中 $M_i = \sum_{j=1}^{42} |x_i(j)|$ ， $i = 1, 2, \dots, 1536$ ， m 為 Super Tree Magnitude 。

$f_1(m)$ 的目的在統計所有 Super Tree Magnitude M_i 中，小於等於 m 之機率佔了多少。圖 21 的曲線顯示了五組不同的 Filter Bank 產生的 $f_1(m)$ 圖形，而此統計圖是根據 12 張常用的影像實驗結果之平均得來的[31]。可以很清楚地看見 JPEG2000 filter 所分解之 Super Tree 其 magnitude 大致偏低，也因此由實驗結果我們發現此 Filter 不適合用於本方法，否則浮水印強度將不足。從實驗中也能看出 JPEG 2000 為了能讓壓縮的資料量變少，因此它的頻率域係數一般來說都比較少，這是它的一個特點。

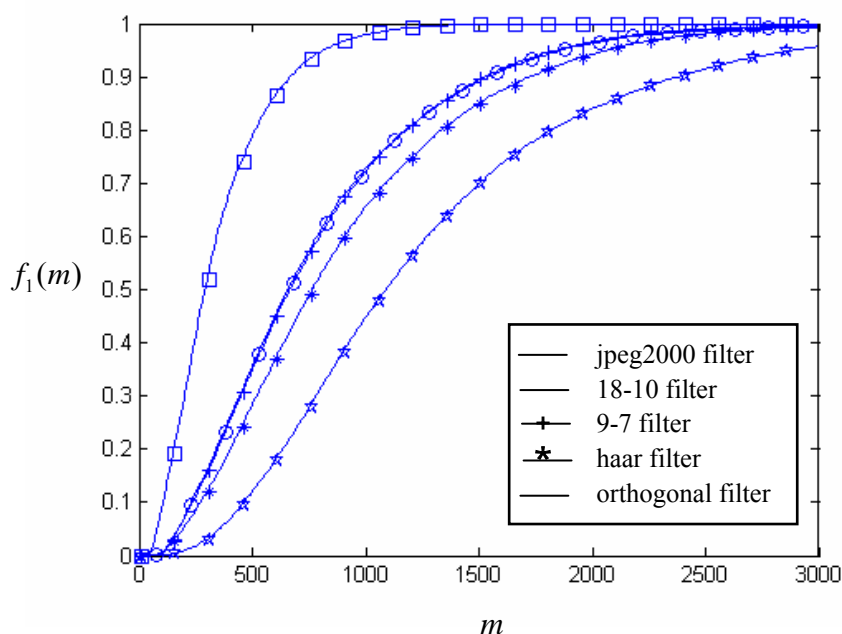


圖 21、以 $f_1(m)$ 檢測五組 Filter Banks 之 magnitude 大小

4.3.2 Filter Bank 對 $Te_i(j)$ 大小之影響。

在 4.2 節中我們討論了 $Te_i(j)$ 誤差值之產生，此外 Filter Bank 亦會影響 $Te_i(j)$ 之大小。同樣地我們定義了一個 CDF of $Te_i(j)$ ，來討論 Filter Bank 對於 $Te_i(j)$ 的影響，其完整定義如下：

$$f_2(e) = \text{probability}[Te_i(j) \leq e]$$

上式中 $Te_i(j) = |x_i(j) - x'_i(j)|$ ， $i = 1, 2, \dots, 1536$ ， $j = 1, 2, \dots, 42$ ， e 為指定之誤差大小。因為 $Te_i(j)$ 對於 quantized tree 影響比較大，所以在圖 22 的實驗中，我們針對所有被量化過的 Super Tree，利用 CDF of $f_2(e)$ 討論 Filter Bank 對 $Te_i(j)$ 誤差之影響，實驗使用之圖形同樣來自[31]中的 12 張常用影像。

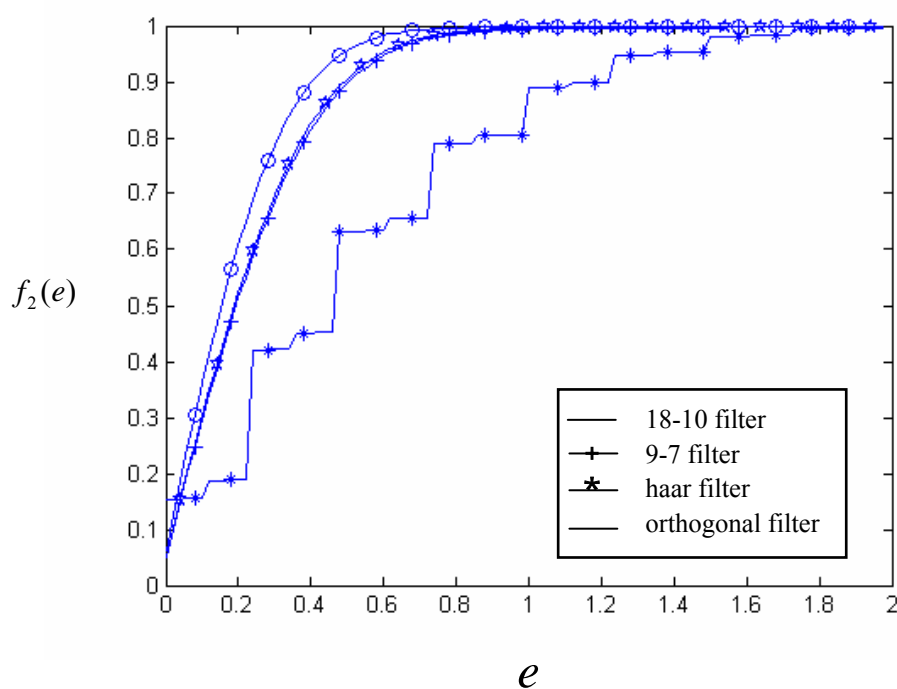


圖 22、以 $f_2(e)$ 檢測四組 Filter Banks 產生的 $Te_i(j)$ 大小

從實驗結果我們可以發現 18-10 filter 對於 quantized Super Tree 之 $Te_i(j)$ 的影響較小，因為它所產生的 $Te_i(j)$ 較小，所以在接下來第五章的實驗中，我們均使用 18-10 filter 當作實驗之用。關於本節測試的五組 filter 係數在附錄(一)中將列舉出來。



4.4 Uniform Deadzone Quantization Algorithm for

Watermark Extraction

驗證一張待測影像是否含有特定的版權證明或其他資訊時，本論文提出的演算法完整流程如圖 23 所示。一開始同樣地將欲檢驗之影像經小波分解至頻率域，接下來以 encoding 使用的 key 尋找出正確的 Super Tree Pair $p'_i (i = 1, 2, \dots, 768)$ 組合。由於每個浮水印 w_i 以量化方式藏於 p'_i 中，所以解浮水印演算法之重點在於如何從 $p'_i(T'_{2i-1}, T'_{2i})$ 中找出何者被量化過，以推測浮水印位元可能的值。

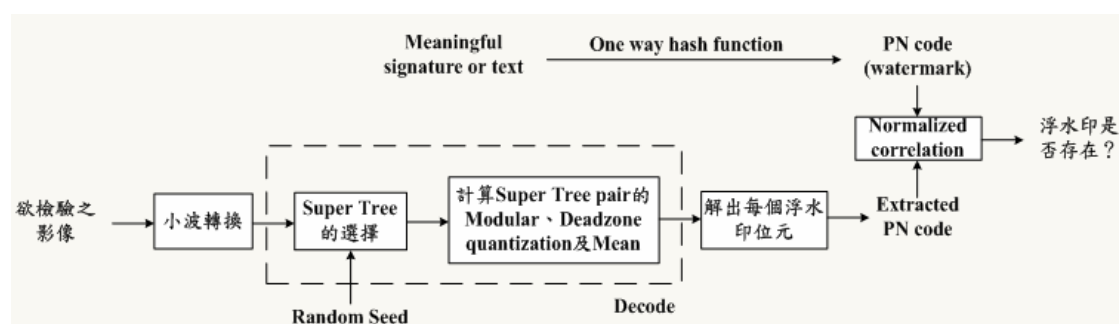


圖 23、Proposed watermark extraction process

4.4.1 Decoding 實作原理

在 Encoding 過程中，我們根據 reference error ε 於 $p_i (T_{2i-1}, T_{2i})$ 找到 quantization index q_i ，假設 q_i 所在的座標為 (a_n, b_n) ，則我們利用 2^{a_n+1} 對於 $p_i (T_{2i-1}, T_{2i})$ 其中一棵 Tree 均勻量化。也就是去除 Super Tree 位元平面 2^0 至 2^{a_n} 中的資訊。然而如果根據 q_i 找到的 $2^{a_n+1} < \Delta_{\min}$ ，則我們便將 q_i 的量化階層強制設為 Δ_{\min} ，雖然這樣會讓 $p_i (T_{2i-1}, T_{2i})$ 所量化的量比其他 Super Tree Pair 大，設置 Δ_{\min} 的好處在於我們可以確保所有 Super Tree 被量化的 quantization step 一定高於 Δ_{\min} 以增加 Robustness。

在 decoding 中，解浮水印演算法的構想是：

在 encoding 中每一對 Super Tree Pair p_i 的 (T_{2i-1}, T_{2i}) 量化階層最少都是 Δ_{\min} ，因此 decoding 可以根據 T'_{2i-1} 與 T'_{2i} 的 bitplane 中 2^0 至 2^{a_n} 的區域之資訊量大小 $x'_{2i-1}(j)_m$ 與 $x'_{2i}(j)_m$ 來做為浮水印位元之判定。

在 $x'_{2i-1}(j)_m$ 與 $x'_{2i}(j)_m$ 中非常接近 0 或非常接近 Δ_{\min} 的數值，我們認為它們很可能在 encoder 中被量化過，因為 $|\Delta_{\min} - x'_{2i-1}(j)_m|$ 或 $|\Delta_{\min} - x'_{2i}(j)_m|$ 之微小的誤差是因為 4.2 節、4.3 節提到的因素而產生的。所以我們必須制定一套機制，將 $|\Delta_{\min} - x'_{2i-1}(j)_m|$ 或 $|\Delta_{\min} - x'_{2i}(j)_m|$ 數值微小之數值變為 0，本文以 deadzone quantization 方法對 $x'_{2i-1}(j)_m$ 與 $x'_{2i}(j)_m$ 量化，變成 $x'_{2i-1}(j)_{mq}$ 與 $x'_{2i}(j)_{mq}$ 。在 $x'_{2i-1}(j)_{mq}$ 與 $x'_{2i}(j)_{mq}$ 中沒有接近 0 或接近 Δ_{\min} 的數值，所以剩餘在 $x'_{2i-1}(j)_{mq}$ 與 $x'_{2i}(j)_{mq}$ 中的數值便成為我們可以判定浮水印之資訊。我們將經過 deadzone quantization 後的 Super Tree Pair 表示為 (Tmq_{2i-1}, Tmq_{2i}) ，如果 $Mean(Tmq_{2i-1}) > Mean(Tmq_{2i})$ 則表示 Tmq_{2i-1} 中資訊量高於 Tmq_{2i} ，浮水印位元 w'_i 可能是嵌在 T'_{2i} ， $w'_i = 1$ ；反之浮水印位元 w'_i 可能是嵌在 T'_{2i-1} ， $w'_i = -1$ 。

4.4.2 浮水印位元取出流程

1. Modularization

目的：從 (T'_{2i-1}, T'_{2i}) 位元平面中取出低於 Δ_{\min} 部份之平面

方法：

我們將 T'_{2i-1} 與 T'_{2i} bitplane 中， 2^0 至 2^{a_n} 部份以 Δ_{\min} 進行 modularization 方式取出使得，

$$x'_{2i-1}(j)_m = x'_{2i-1}(j) \text{ MOD } \Delta_{\min}$$

$$x'_{2i}(j)_m = x'_{2i}(j) \text{ MOD } \Delta_{\min}$$

其中 $x'_{2i-1}(j)_m$ 與 $x'_{2i}(j)_m$ 分別是 $x'_{2i-1}(j)$ 與 $x'_{2i}(j)$ 經過 Δ_{\min} modularization 之結果， $j = 1, 2, \dots, 42$ 。

2. Deadzone scalar Quantization

目的：以 Deadzone Scalar Quantization 處理 $(x'_{2i-1}(j)_m, x'_{2i}(j)_m)$ 中數值，接近 0 或接近 Δ_{\min} 設為 0。

方法：我們定義 $x'_l(j)_{mq}$ 為 $x'_l(j)_m$ 經過 Deadzone Scalar Quantization 後的值，

$$\text{令 } Z_l(j) = \text{pow}2((\text{int})[\log(x'_l(j)_m)/\log 2] + 1)$$

則

$$x'_l(j)_{mq} = \begin{cases} 0 & , \text{ if } x'_l(j)_m = 0 \\ Z_l(j) & , \text{ if } x'_l(j)_m \geq Z_l(j) - Z_l(j)/4 \\ Z_l(j) - Z_l(j)/2 & , \text{ otherwise} \end{cases}$$

其中， $l = 2i - 1, 2i$ ， $j = 1, 2, \dots, L$

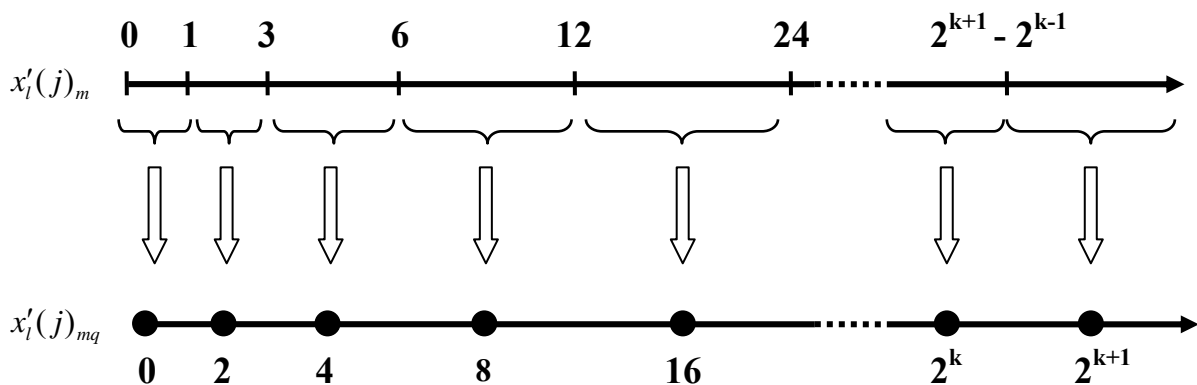


圖 24、Deadzone Scalar quantization 示意圖

如圖 24，Deadzone Scalar Quantization 就是依 $x'_l(j)_m$ 值大小的不同，而有不同的 quantization step size。最後，如果 $x'_l(j)_{mq} = \Delta_{\min}$ 時，則將 $x'_l(j)_{mq}$ 以 Δ_{\min} modular 為 0。

3. Mean of deadzone quantization tree for Watermark bit extraction

目的：計算以 deadzone 量化過樹的平均值 $Mean(Tmq_{2i-1})$ 、 $Mean(Tmq_{2i})$ 以資訊量做為判斷取出浮水印位元 w'_i 。

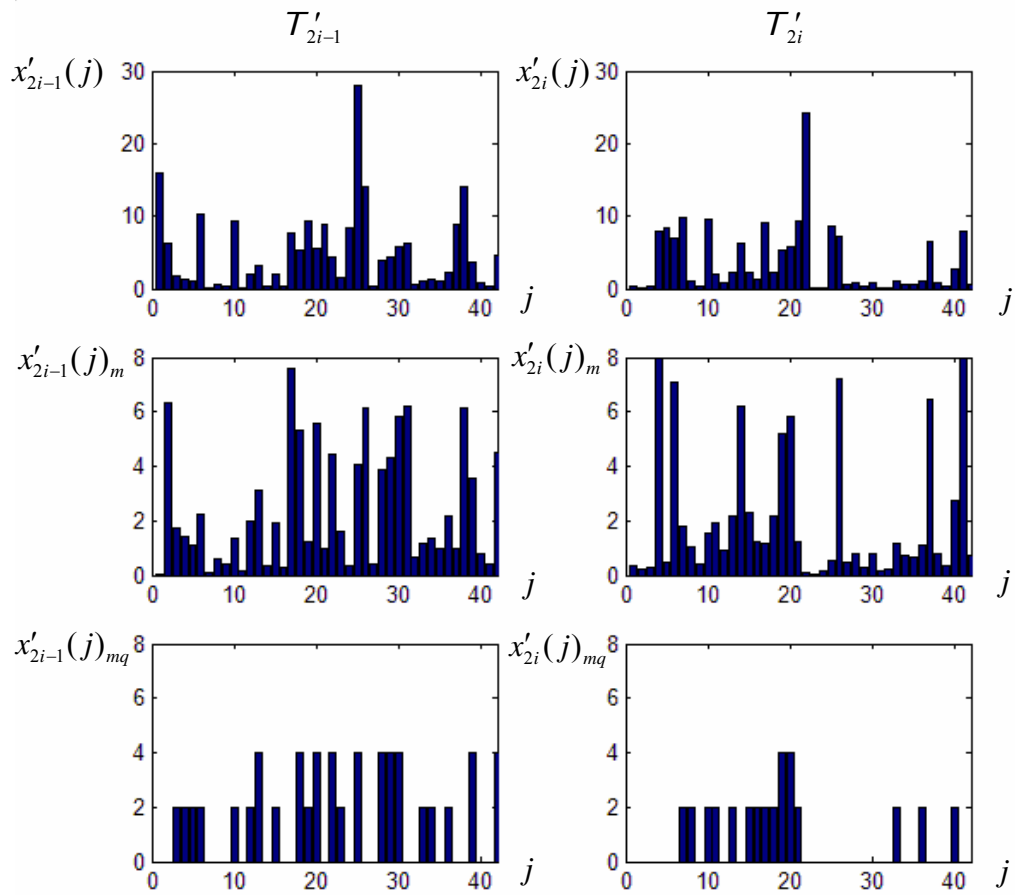
方法：

我們定義一棵 deadzone 量化過樹的平均值 $Mean(Tmq_l)$ 如下：

$$Mean(Tmq_l) = \frac{\sum_{j=1}^L x'_l(j)_{mq}}{L}, \text{ 其中 } L = 42, l = 2i - 1, 2i$$

如果 $Mean(Tmq_{2i-1}) > Mean(Tmq_{2i})$ 則表示 Tmq_{2i-1} 中資訊量高於 Tmq_{2i} ，浮水印位元 w'_i 可能是嵌在 T'_{2i} ， $w'_i = 1$ ；反之浮水印位元 w'_i 可能是嵌在 T'_{2i-1} ， $w'_i = -1$ 。

圖 25 是一個簡單的解浮水印演算法運作範例，圖 25(a)與圖 25(b)分別為兩棵待測 Super Tree (T'_{2i-1}, T'_{2i})。圖 25(c)與圖 25(d)分別為 (T'_{2i-1}, T'_{2i}) 經過 modularization 後之值 ($x'_{2i-1}(j)_m, x'_{2i}(j)_m$)， $j = 1, 2, \dots, 42$ (假設 $\Delta_{\min} = 8$)，則所有 modularization 後之係數系均介於 0 與 8 之間。圖 25(e)與圖 25(f)分別為係數 ($x'_{2i-1}(j)_m, x'_{2i}(j)_m$) 經過 deadzone scalar quantization 後的值 ($x'_{2i-1}(j)_{mq}, x'_{2i}(j)_{mq}$)。最後實驗根據 deadzone quantized tree (Tmq_{2i-1}, Tmq_{2i}) 之平均值 $Mean(Tmq_{2i-1})$ 、 $Mean(Tmq_{2i})$ 來判定浮水印位元 w'_i 。最後的結果 $Mean(Tmq_{2i-1}) > Mean(Tmq_{2i})$ ，所以判定 $w'_i = 1$ 。



結果 \implies $Mean(Tmq_{2i-1}) = 1.523810$ $Mean(Tmq_{2i}) = 0.809524$



子圖順序

(a)	(b)
(c)	(d)
(e)	(f)

圖 25、一個對於 (T'_{2i-1}, T'_{2i}) 以 deadzone scalar quantization 之範例。

4.5 Normalized Correlation 與 False positive probability 分析

當所有的浮水印位元均經過演算法解出以後，為了判別解出的浮水印與原浮水印是否相同，我們使用 Normalized Correlation Coefficient 做為相關性的判斷[21] 比較與相關性門檻值 ρ_T 之大小，以宣稱浮水印的存在。Normalized Correlation Coefficient 之定義如下：

$$\rho(W, W') = \frac{\sum_{m=1}^{N_w} w_m w'_m}{\sqrt{\sum_{m=1}^{N_w} w_m^2 \sum_{m=1}^{N_w} w'_m{}^2}}$$

由於原始浮水印 W 與欲檢查之浮水印 W' 都屬於 PN code，那麼接下來我們將討論門檻值 ρ_T 的大小，對於此二浮水印之判別的正確性達到最低的誤判率[21] 們定義 P_f 為 false positive probability，其代表的含意為在一個未加入浮水印的圖形中，檢測出浮水印之機率，那麼 $P_f = \{\rho(W, W') \geq \rho_T \mid non_watermark\}$ 。從 $\rho(W, W')$ 原式中，我們可以知道

分母部份：

$$\sqrt{\sum_{m=1}^{N_w} w_m^2 \sum_{m=1}^{N_w} w'_m{}^2} = \sqrt{\sum_{m=1}^{N_w} w_m^2} \times \sqrt{\sum_{m=1}^{N_w} w'_m{}^2} = \sqrt{N_w} \times \sqrt{N_w} = N_w$$

分子部份：

此兩個浮水印位元的第 m 個位元之相乘值為 $w_m w'_m$ ，如果第 m 個位元之 $w_m w'_m = 1$ 代表兩個浮水印位元之值相同，如果 $w_m w'_m = -1$ 表示浮水印位元值相異，則我們令 $w_m w'_m = K(m)$ ，則可知 $K(m) \in \{1, -1\}$ ，原式可改寫成：

$$\rho(W, W') = \frac{\sum_{m=1}^{N_w} K(m)}{N_w}$$

因

此

$$P_f = \left\{ \rho(W, W') \geq \rho_T \mid non_watermark \right\} = \left\{ \frac{\sum_{m=1}^{N_w} K(m)}{N_w} > \rho_T \mid non_watermark \right\}$$

$$= \left\{ \sum_{m=1}^{N_w} K(m) \geq \rho_T N_w \mid non_watermark \right\}$$

也就是說在 non_watermark 圖形中，測得只要滿足大於等於 $\rho_T N_w$ 的所有 $K(m)$ 之可能性，就是 false positive 的發生機率。

$$\sum_{m=1}^{N_w} K(m) \in \{-N_w, -N_w + 2, -N_w + 4, \dots, -N_w + 2n, \dots, N_w - 4, N_w - 2, N_w\}$$

我們可以用使用通式 $-N_w + 2n$ 表示 $\sum_{m=1}^{N_w} K(m)$ ，其中 $n = 0, 1, 2, \dots, N_w$ 。所有

$\sum_{m=1}^{N_w} K(m) \geq \rho_T N_w$ 的可能性，亦即從 $-N_w + 2n$ 之所有滿足條件式的機率加總：

$$P_f = \sum_{n=\lceil N_w \times (\rho_T + 1) / 2 \rceil}^{N_w} P \left\{ \sum_{m=1}^{N_w} k(m) = -N_w + 2n \mid non_watermark \right\}$$

上式中加總的 lower bound 為 $\sum_{m=1}^{N_w} K(m) = \rho_T N_w$ ，因此從 $-N_w + 2n = \rho_T N_w$ 可

以推導出 $n = \lceil N_w \times (\rho_T + 1) / 2 \rceil$ 為機率加總之 lower bound。然而

$$P \left\{ \sum_{m=1}^{N_w} k(m) = -N_w + 2n \mid non_watermark \right\} = \binom{N_w}{n} P_E^{N_w - n} \cdot (1 - P_E)^n, \quad P_E \text{ 表示}$$

$K(m) = -1$ 的機率亦即 $w_m \neq w'_m$ 之機率，在此 $P_E = 0.5$ 。

$$P_f = \sum_{n=\lceil N_w \times (\rho_T + 1) / 2 \rceil}^{N_w} \binom{N_w}{n} P_E^{N_w - n} \cdot (1 - P_E)^n = \sum_{n=\lceil N_w \times (\rho_T + 1) / 2 \rceil}^{N_w} \binom{N_w}{n} 0.5^{N_w}$$

由上式可以看出，兩個 PN code 的 False positive probability 跟浮水印長度及門檻值有極大的關係，在選定門檻值與浮水印長度的同時，為了建立與[24]的比較基礎，所以在本文的實驗，使用長度為 512 的浮水印並設定 $\rho_T = 0.23$ ，在

$P_f = 1.03 \times 10^{-7}$ 底下進行比較。較完整的 ρ_T 與 N_w 關係如表二所示。

$N_w \backslash \rho_T$	0.15	0.2	0.23	0.25
768	1.61×10^{-5}	1.5×10^{-8}	1.13×10^{-10}	2.14×10^{-12}
512	4.5×10^{-4}	3.78×10^{-6}	1.03×10^{-7}	8.45×10^{-9}

表格 2 、不同的 ρ_T 與 N_w 得到的 False positive probability



4.6 完整浮水印嵌入與取出之演算法

4.6.1 浮水印嵌入演算法

- Step 1：將能夠證明版權的 logo、使用者與擁有者之描述或圖形相關資訊，經過一個 one way hash 後得 ± 1 的數字序列(PN code)，以此 PN code 為浮水印。
- Step 2：以小波轉換計算出頻率域係數值，並以一個虛擬亂數的方式，將兩個小波樹結合成為一個 super tree T_k ，其中 $k = 1, \dots, 2N_w$ 。設定 $i = 1$ 。
- Step 3：如果 $M(T_{2^{i-1}}) < \varepsilon$ 或 $M(T_{2^i}) < \varepsilon$ ，則 $\Delta_i = \Delta_{\min}$ 並且移至 Step 6，否則移至 Step 4。
- Step 4：設定 $q_i = 1$ ， $\varepsilon_{2^{i-1}}(1) = 0$ 以及 $\varepsilon_{2^i}(1) = 0$
- Step 5：while ((($\varepsilon_{2^{i-1}}(q_i) < \varepsilon$) or ($\varepsilon_{2^i}(q_i) \geq \varepsilon$)) and $q_i < q_{\max}$) 計算 $\varepsilon_{2^{i-1}}(q_i)$ 與 $\varepsilon_{2^i}(q_i)$ 並設定 $q_i = q_i + 1$ 。
- Step 6：如果 $2^{q_i+1} \leq \Delta_{\min}$ 則 $\Delta_i = \Delta_{\min}$ ，否則 $\Delta_i = 2^{q_i+1}$ (如圖 18 所顯示)
- Step 7：如果 $w_i = -1$ 則對 $T_{2^{i-1}}$ 量化，否則對 T_{2^i} 做量化。設定 $i = i + 1$ 。
- Step 8：如果 $i < N_w$ ，則回到步驟二。
- Step 9：將已修改過小波係數，經由反小波轉換得到空間域，得到一張已嵌入浮水印的影像。

4.6.2 浮水印取出演算法

- Step 1：同嵌入過程中的 Step 1，取得原浮水印 PN code。
- Step 2：計算欲解浮水印影像中，每個像素的位元數 $b(\text{bits/pixel})$ 。以小波轉換計算出頻率域係數值，並以一個虛擬亂數的方式，將兩個小波樹結合成為一個 super tree T_k ，其中 $k = 1, \dots, 2N_w$ 。設定 $i = 1$ 。
- Step 3：設定 $\Delta'_i = \Delta'_{\min}$ 。
- Step 4：對每個在 $T'_{2^{i-1}}$ 中的係數 $x'_{2^{i-1}}(j)$ 以及在 T'_{2^i} 中的係數 $x'_{2^i}(j)$ 以 Δ'_i 進行

Modularization ($j = 1, \dots, L$) :

$$x'_{2^{i-1}}(j)_m = x'_{2^{i-1}}(j) \text{ MOD } \Delta'_i \text{ 以及}$$

$$x'_{2^i}(j)_m = x'_{2^i}(j) \text{ MOD } \Delta'_i$$

Step 5 : 對 $x'_{2^{i-1}}(j)_m$ 及 $x'_{2^i}(j)_m$ 進行 deadzone scalar quantization (其中 $j = 1, 2, \dots, L$)

後, 得到 modularization 過的樹, 其係數的表示法為 $x'_{2^{i-1}}(j)_{mq}$ 及 $x'_{2^i}(j)_{mq}$,

其演算過程如下 :

$$\text{令 } Z_l(j) = \text{pow}2((\text{int})[\log(x'_l(j)_m) / \log 2] + 1), l = 2n - 1, 2n$$

$$\text{IF } x'_l(j)_m < 1.0, \text{ then } x'_l(j)_{mq} = 0$$

$$\text{Else if } x'_l(j)_m \geq Z_l(j) - Z_l(j) / 4, \text{ then } x'_l(j)_{mq} = Z_l(j)$$

$$\text{Else } x'_l(j)_{mq} = Z_l(j) - Z_l(j) / 2$$

$$\text{IF } x'_l(j)_{mq} = \Delta'_i, \text{ set } x'_l(j)_{mq} = 0$$

Step 6 : 如果 $\text{Mean}(Tmq_{2^{i-1}}) < \text{Mean}(Tmq_{2^i})$ 則 $w'_i = -1$, 反之 $w'_i = 1$ 。

Step 7 : 設定 $i = i + 1$, 如果 $i < N_w$, 則回到 Step 2。

Step 8 : 計算 normalized correlation coefficient ρ 。

Step 9 : 如果 ρ 值大於門檻值 ρ_T , 則表示浮水印存在; 否則此驗證之圖形中不存在浮水印。

第五章、研究成果

本章利用前面章節所提出的方法，將影像加入浮水印後，實際對影像進行各種攻擊，以證明本論文所題出的方法具有強韌性，並且能保證已嵌入之數位版權資訊能成功地被偵測出來，同時探討本研究能夠承受各項攻擊最大的程度為何。本章節亦列舉參考論文[1]項實驗結果進行比較。

(a).



(b).



(c).

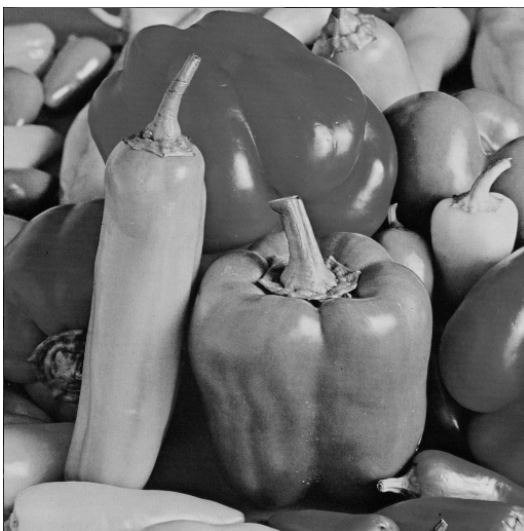


圖 26、實驗所使用的三張影像

(a)Lena

(b)Goldhill

(c)Peppers

實驗中所用的圖形分別是知名的 Lena、Goldhill 及 Peppers 三張圖，而嵌入浮水印之後三張圖形之品質，將它們分別維持在超過 38.2(dB)、38.7(dB)及

39.8(dB)，這樣使得本論文實驗結果與[1]果在進行比較時，能有一個合理的比較基礎，同時對人類視覺感觀上也不會跟原圖形差距太大。

在實驗使用的參數設定方面，本論文使用的浮水印長度為 512 的±1 數字序列，最小量化階層(Minimum quantization step)設定為 16，最大的量化索引值設定為 336，最後以 Normalized Correlation 做為浮水印 Existence 的偵測計算，門檻值 $\rho=0.23$ ，也就是在 False positive probability 為 1.03×10^{-7} 之下進行原浮水印與待測浮水印之比較。

實驗的測試內容可以分成三個部份，於下表中歸納列出：

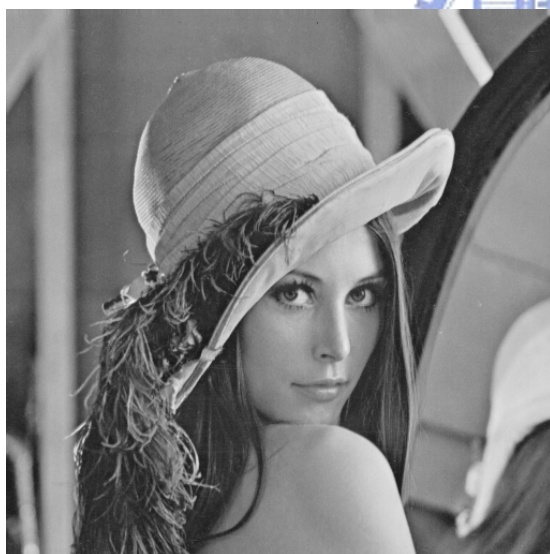
測試種類	測試實驗
抗壓縮能力	JPEG compression、SPIHT compression
空間域各種破壞	Median filter、Bitplane remove、Pixel shifting、 Filter convolusion、Rotation and scaling
多重浮水印嵌入	Multiple-watermarking attack

表格 3、浮水印抗攻擊實驗列表

5.1 JPEG Compression

JPEG 由國際標準組織(International Organization for Standardization, 簡稱 ISO) 和國際電話電報諮詢委員會(International Telegraph and Telephone Consultative Committee, 簡稱 CCITT) 所建立的一個數位影像壓縮標準，用於靜態影像壓縮方面。JPEG 的編碼方法的概念，是將影像切割成以 Block 為單位的區塊，再對這些 Block 利用數位餘弦轉換法(Discrete Cosine Transform, 簡稱 DCT) 轉換至頻率域後，將影像資料中位於高頻，較不重要的部份去除，僅保留低頻重要的資訊，達到高壓縮率的目的。雖然 JPEG 壓縮是屬於失真壓縮，也就是處理後的影像會有失真的現象，不過 JPEG 的失真比例可以利用 compression ratio 參數來加以控制的。

(a).



原始影像

(b)



以 JPEG 壓縮對影像攻擊結果
(quality factor = 30)
correlation = 0.28

圖 27、原始 Lena 影像與以 JPEG 壓縮對影像攻擊過影像之比較 (a)原始影像
(b)JPEG 壓縮過之影像。

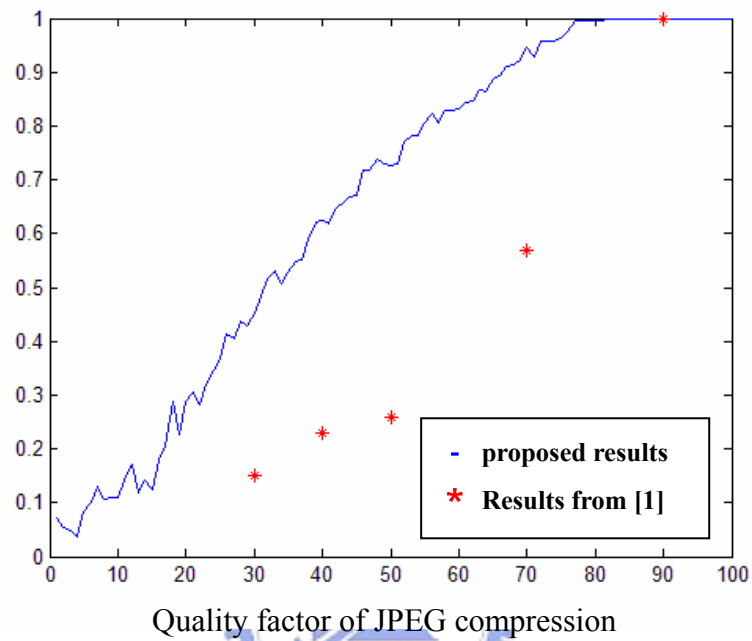
圖 27 是一個比較的例子，其中圖(a)為原始影像，圖(b)為經過 JPEG 壓縮 (quality factor = 30)後結果，影像大小減少，不過卻犧牲了圖形品質，不過我們

仍能從此壓縮過的影像測得浮水印的存在($\rho=0.28$)。

實驗結果：

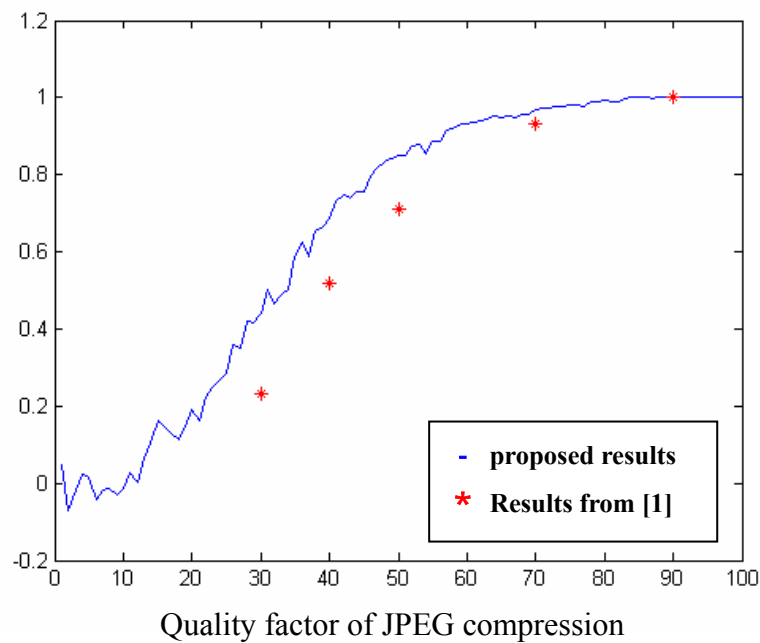
(a).Lena

Correlation value



(b).Gold hill

Correlation value



(c). Peppers

Correlation value

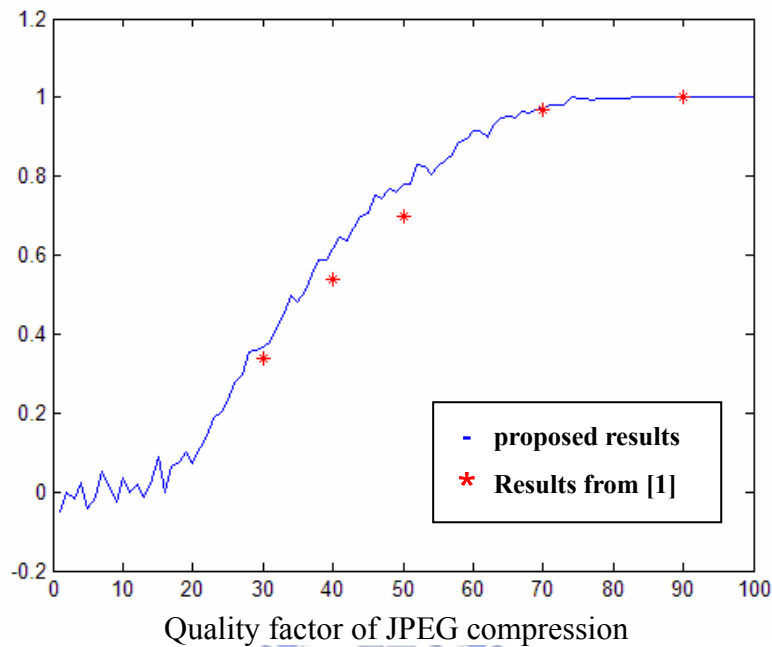


圖 28、三張影像對於 JPEG 攻擊之承受程度 (a)Lena (b)Goldhill (c)Peppers

在圖 28 的實驗結果中，圖(a)、(b)與(c)分別是以 Lena、Godhill 及 Peppers 為實驗對象，設定 JPEG 的 quality factor 由 1 到 100 進行壓縮攻擊。三張實驗結果圖形中，實線部份為本研究之實驗成果，而以”*”標注的數值為參考論文[1]結果。經由圖形可以明顯看出本文提出的方法對於 Lena 圖形大幅提升了浮水印的強韌性，在壓縮係數為 20 左右的 JPEG 壓縮仍能測得浮水印。而在 Goldhill 的結果中，我們也將抗壓縮能力，由壓縮係數 30[1]提升至 20。最後 Peppers 的實驗結果雖然無大幅改善，不過也增加了浮水印判定之精確度。

5.2 SPIHT compression

SPIHT compression 是一種 wavelet-based 影像壓縮演算法，其全名為 *Set Partitioning in Hierarchical Trees* (SPIHT)。它是在 1996 年由 Said 等學者所提出的，目前成為新一代影像壓縮標準 MPEG-4 與 JPEG-2000 的核心技術。SPIHT 的優點在於壓縮過後的影像仍然具有非常高的畫質(相較其他壓縮方法而言)，並且影像在傳輸時能提供漸近式的影像傳輸(Progressive image transmission)，也就是 Receiver 能快速得到所接收影像的大致內容，能節省影像於網路中傳輸的資料量。此外快速的 encoding 與 decoding speed 以及提供無失真壓縮，也是 SPIHT 重要的特性。

圖 29 是一個簡單的 SPIHT 壓縮前與壓縮後的圖片對照。圖 29(a)為原始影像，(b)為原始影像經過 bitrate 為 0.3 的 SPIHT 壓縮後結果，我們仍然可從圖中偵測出浮水印存在。

(a)



原始 Lena 影像

以 SPIHT 對影像壓縮後結果
(bitrate = 0.3, PSNR=33.33)
correlation = 0.32

圖 29、原始 Lena 影像與 SPIHT 壓影像之對照(a)原始影像(b)SPIHT 壓縮後影像

實驗結果：

表格 4(b)是[1]研究結果，兩個表可供對照用。在實驗結果中，Lena 圖形在 bitrate 為 0.3、0.4 時優於[1]之結果。Goldhill 的結果皆優於[1]結果，並且在 bitrate 為 0.3、0.4 時均能有解測得浮水印。不過在 Peppers 圖形雖然本文提出的方法均能測得浮水印，不過精確度尚無法達到[1]的優良結果。

(a)Proposed results

		0.3	0.4	0.5	0.6	0.7
Lena	ρ	0.32	0.49	0.61	0.71	0.79
	PSNR(dB)	33.76	34.64	35.16	35.52	35.87
Goldhill	ρ	0.23	0.32	0.38	0.51	0.66
	PSNR(dB)	30.80	31.68	32.45	33.07	33.57
Peppers	ρ	0.27	0.46	0.61	0.73	0.80
	PSNR(dB)	33.61	34.41	34.83	35.17	35.49

(b)Reference from [1]

		0.3	0.4	0.5	0.6	0.7
Lena	ρ	0.21	0.41	0.85	0.83	0.85
	PSNR(dB)	33.1	34.3	34.6	35.2	36.7
Goldhill	ρ	-0.06	0.02	0.23	0.27	0.35
	PSNR(dB)	31.7	32.4	32.9	33.2	34.1
Peppers	ρ	0.36	0.66	0.65	0.71	0.85
	PSNR(dB)	33.13	33.6	34.4	34.7	34.9

表格 4、SPIHT compression (a)Proposed results (b)Reference from [1]

5.3 Median Filter Attack

Median Filter 是常見的影像處理技術，它通常用於移除影像當中多的雜訊 (Noise)，不過也因此造成了影像的模糊化。Median filter 是 non-linear filter 的一種，它比一般的 non-linear filter 具有更高的強韌性，因為它能夠保存影像中邊緣的部份(sharp edges)。不過缺點在於運算時非常沒有效率，其問題在於在對於 mask 中的值進行排序時非常花時間。解決排序時間辦法是使用 quicksort 進行排序工作。

Median Filter Attack 實作原理：

Median filter 是一個屬於空間域中的方法，通常我們使用一個 2-D 的 Mask 將一個大小為 $m \times n$ 的影像中所有的點掃描過一次。當 Mask apply 到影像中任一個區域時，便將此區域中的中間值取出，當作新的影像中的一個點。如圖 30 所示，以 3×3 的 mask 對一張原始影像進行處理，最後取出中值放至新產生的影像當中，放置這個中間值的座標為 mask 的中心座標。原始圖形中 mask 中心位置是 (3,3)，最後從 mask 中得到的中值便存在右圖座標 (3,3) 位置。

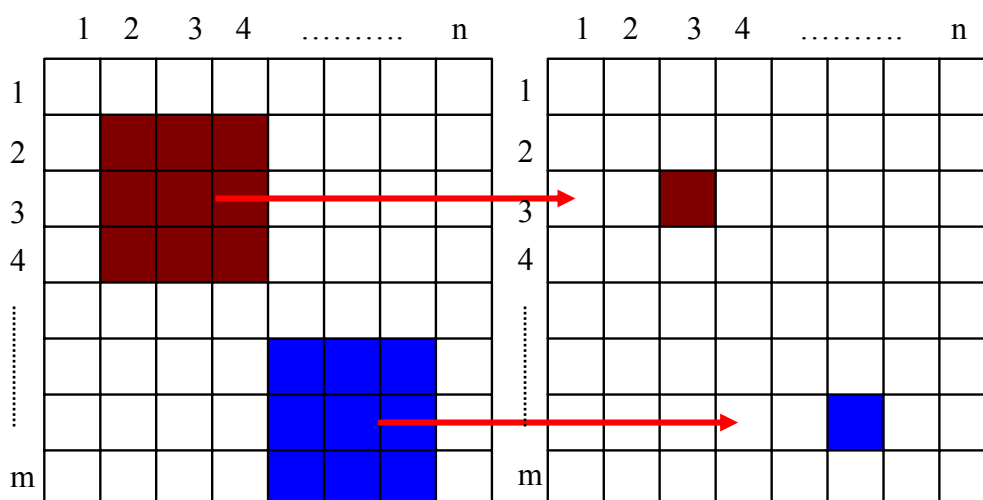


圖 30、Median Filter 運作原理

取中值的方式，是將 mask 中的資訊排序後，取出最中間的值，如下圖的範例所示：

135	152	106	原始資料：135 , 152 , 106 , 83 , 98 , 72 , 56 , 65 , 41
83	98	72	排序資料：41 , 56 , 65 , 72 , 83 , 98 , 106 , 135 , 152
56	65	41	取出中間值： 83

圖 31(a)為原始 Lena 圖形，圖 31(b)是以 6×6 的 mask size 對原始影像破壞後的結果，可以明顯發現圖形的邊緣部份被強調了，但同時也使圖形變模糊。不過仍然可以從(b)圖中偵測出浮水印存在($\rho = 0.28$)。

(a)

(b)



原始影像

以 6*6 median filter 對影像攻擊結果
correlation = 0.28

圖 31、原始 Lena 影像與 median filter 處理過影像之對照 (a)原始影像 (b)以 6×6 median filter 處理過之影像

實驗結果：

在表格 5 中，表格 5(a)是以本文提出之方法，針對 Lena、Goldhill、Peppers 三張圖形，以不同 Mask size 進行實測後的結果。表格 5(b)中的實驗結果來自參

考論文[1]，以供表表格 5(a)做實驗結果對照。由表格 5(a)的結果可以知道本文提出的方法能夠抵抗 mask size 從 2x2 ~ 5x5 的作用(甚至能在 Lena 經過 5x5mask 後圖形中測得浮水印， $\rho=0.28$)，結果優於表(b)(能抵抗 mask size 從 2x2 ~ 4x4)。

(a)Proposed results

Mask size Images	2x2	3x3	4x4	5x5	6x6
Lena	0.48	0.85	0.38	0.43	0.28
Goldhill	0.45	0.81	0.33	0.32	0.20
Peppers	0.62	0.89	0.36	0.41	0.20

(b)Results from reference [1]

Mask size Images	2x2	3x3	4x4	5x5	6x6
Lena	0.38	0.51	0.23	NA	NA
Goldhill	0.35	0.56	0.24	NA	NA
Peppers	0.46	0.71	0.25	NA	NA

(NA – not availible)

表格 5、Median filter attack (a)Proposed results (b) Reference from[1]

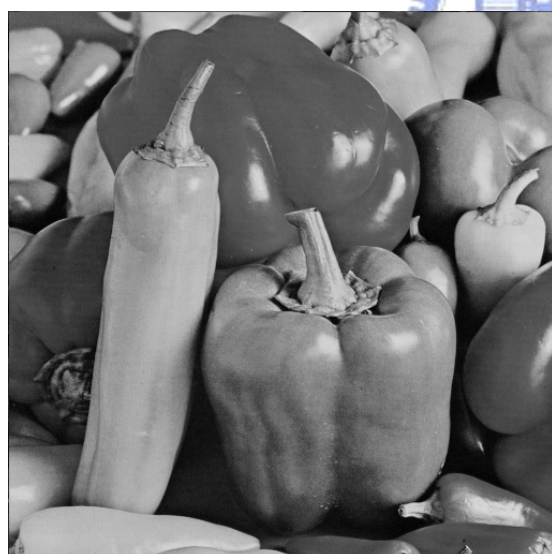
5.4 Pixel Shifting Attack

Pixel Shifting 亦是一種簡單的 spatial domain 攻擊方法，透過對影像中每一個 row 或 column 的旋轉、位移達到目的。位移過後的影像與原始影像雖然對於人的肉眼上的判斷不會有很大差異，不過在兩張圖的頻率域上卻有很大的變化，造成浮水印判定不準確。兩種 pixel Shifting types 將在底下進行介紹：

Pixel Shifting - Type 1 :

對於影像中每個 row 作位移，所有的 pixel 往右位移 n 個位置，而最右方多出來的 n 個 pixel 則補到此 row 的最左邊，也就是形成一個 circular shifting。如圖 32 所示：

(a)



Peppers 原始影像

(b)



Peppers 往右 shifting 9 個 pixels
(使用 Type 1 方法)

圖 32、原始 Peppers 影像與 type 1 shifting 影像之對照 (a)原始影像 (b)以 type1 shift 9 個 pixels 之影像

Pixel Shifting - Type 2

對於影像中每個 column 作位移，所有的 pixel 往下位移 n 個位置，而最下方多出來的 n 個 pixel 則補到此 column 的最上方，同樣是形成一個 circular shifting。

如圖 33 所示：

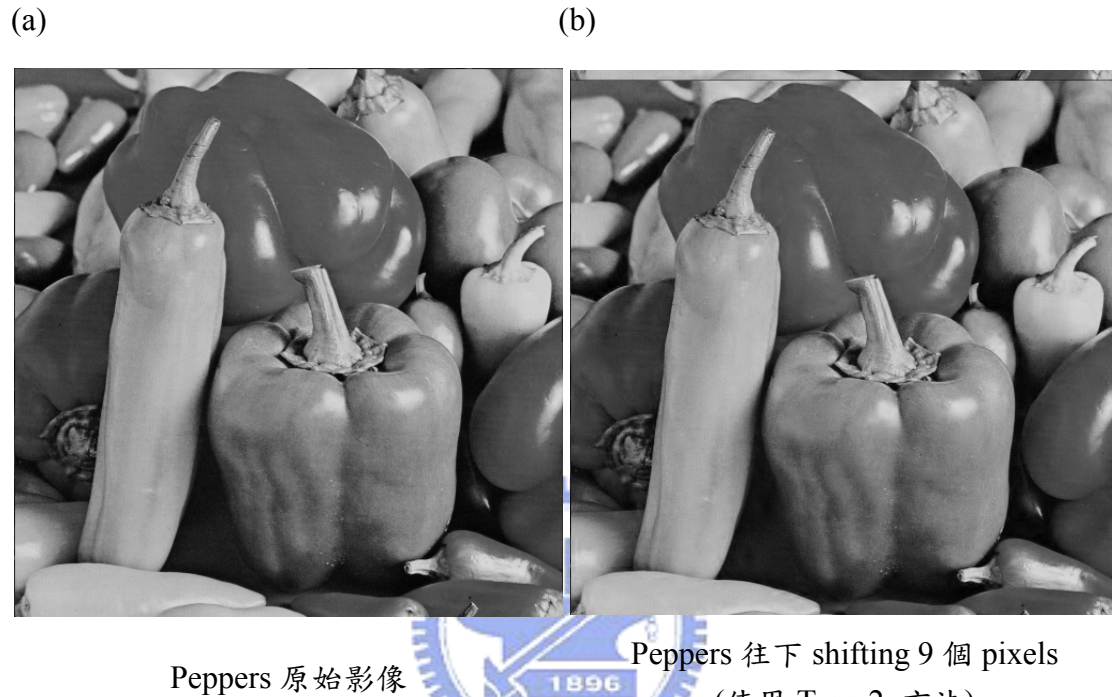


圖 33、原始 Peppers 影像與 type 2 shifting 影像之對照 (a)原始影像 (b)以 type2 shift 9 個 pixels 之影像

實驗結果：

表格 6 中，表(a)與(c)分別為本論文使用 type 1 與 type 2 pixel shifting 之實驗結果，而(b)與(d)則為參考論文中的實驗結果。

(a)Proposed results - Pixel Shifting Type 1

	3	4	5	6	7	8	9	10	11
Lena	0.40	0.91	0.31	0.31	0.32	0.84	0.27	0.23	0.17
Peppers	0.48	0.98	0.42	0.36	0.41	0.90	0.35	0.21	0.23
Goldhill	0.51	0.93	0.46	0.45	0.48	0.89	0.36	0.34	0.26

(b)Results from [1]-Pixel Shifting Type 1

	3	4	5	6	7	8	9	10	11
Lena	NA	NA	0.28	0.34	0.29	0.81	0.26	0.19	NA
Peppers	NA	NA	0.36	0.35	0.41	0.84	0.29	0.21	NA
Goldhill	NA	NA	0.32	0.34	0.29	0.92	0.29	0.26	NA

(c)Proposed results - Pixel Shifting Type 2

	3	4	5	6	7	8	9	10	11
Lena	0.40	0.89	0.36	0.29	0.39	0.82	0.29	0.21	0.21
Peppers	0.50	0.99	0.40	0.32	0.40	0.90	0.26	0.18	0.20
Goldhill	0.52	0.95	0.38	0.34	0.39	0.87	0.31	0.27	0.20

(d)Results from [1]-Pixel Shifting Type 2

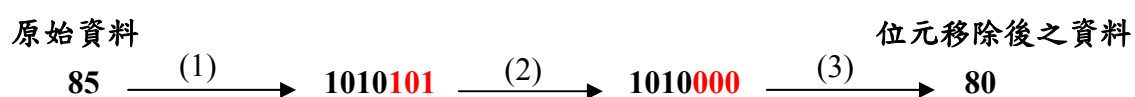
	3	4	5	6	7	8	9	10	11
Lena	NA	NA	0.27	0.33	0.27	0.82	0.25	0.17	NA
Peppers	NA	NA	0.37	0.31	0.43	0.85	0.25	0.20	NA
Goldhill	NA	NA	0.34	0.33	0.31	0.91	0.28	0.27	NA

表格 6、Pixel shifting attack (a)Proposed results(Pixel Shifting Type 1) (b) Reference from[1] (Pixel Shifting Type 1) (c) Proposed results(Pixel Shifting Type 2) (d) Reference from[1] (Pixel Shifting Type 2)

由上面實驗可以看出，本研究所提出的方法針對實驗的 Lena 與 Goldhill 圖形，最多可以有效抵抗 9 個像素位移(包含 type 1 及 type 2 的實驗結果)，而 Peppers 對於 type 1 及 type 2 pixel shifting 也能抵抗 10 個像素位移。這樣的實驗結果可以看出，本文對於 pixel shifting 的抵抗能力與參考論文相同，不過對於判定之精確度卻是有提升的，有效增加了判定結果。

5.5 Bit-plane Remove Attack

移除 spatial domain 的 LSB 是常見且容易實作的影像破壞方法，其做法是對於一張影像中所有像素，移除 k 個 LSB，其中 $k=1,2,\dots,8$ 。當 k 愈小時，影像中被移除的資訊愈少，當 k 增加時，影像中像素值低的部份會漸漸量化為黑色， $k=8$ 時則整個影像便消失(所有資訊都被移除)。下面為一個對於一個 pixel 移除其 3 個 LSB 的例子：



(1) 將原始資料轉換成二進位表示

(2) 移除二進位資料中最低的三個位元

(3) 將位元移除之資料轉換為 10 進位，完成處理

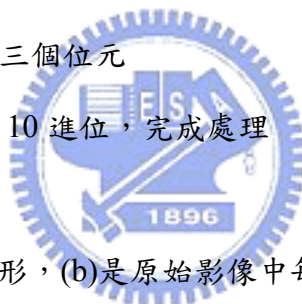


圖 34(a)為原始 Lena 圖形，(b)是原始影像中每個像素受到移除 4 個最低位元破壞後的結果。仍然可以從(b)圖中偵測出浮水印存在($\rho=0.28$)。

(a)



(b)



原始 Lena 影像

移除空間域中 4 個 LSB 的方式，對影像破壞結果 $\rho=0.28$

圖 34、原始 Lena 影像與移除空間域中每個像素 4 個 LSB 影像之對照 (a)原始影像 (b)移除空間域中 4 個 LSB 對影像破壞結果

實驗結果：

(a)Proposed results

Images \ LSBs	1	2	3	4	5
Lena	1	1	0.98	0.83	0.16
Goldhill	1	1	1	0.82	0.20
Peppers	1	1	1	0.70	0.11

(b)Results from [1]

Images \ LSBs	1	2	3	4	5
Lena	1	1	0.99	0.52	0.11
Goldhill	1	1	0.97	0.38	0.14
Peppers	0.99	0.96	0.90	0.64	0.14

表格 7、(a)Proposed results (b)Reference from [1]

表格 7 中，表(a)是由本論文的方法，對於 bits remove attack 之實驗結果。表(b)為參考論文[1]之實驗結果。兩者的結果均能有效抵抗最多移除 4 個最低位元的破壞，不過由數據中也說明了本文提出的方法在對於移除 4 個最低位元的破壞有較佳的效果，對於浮水印判定之精確度能有顯著提升。

5.6 Multiple Watermark Attacks

對於一張已嵌有浮水印的圖形進行 multiple watermarking attack 的目的，主要是想藉此讓原本在影片中的浮水印消失。所以浮水印演算法能抵抗這種抹去浮水印的攻擊方式是很重要的，如此才能確保版權資訊能安全地存在。

多重浮水印攻擊的方法：

在本論文的演算法中，加第一張浮水印是依據 PSNR 來保持影像品質。以 goldhill 為例，我們將加完浮水印圖形的 PSNR 保持高於 38.7dB，在此同時記錄其量化索引值 q_n ，做為接下來其他浮水印嵌入同一張影像所使用。如此一來便可以保證接下來額外嵌入的浮水印強度與原浮水印相同。如圖 35(b)即 Goldhill 額外嵌入 4 個浮水印後的影像，仍然能測得浮水印的存在($\rho = 0.28$)。

(a)



原始 Goldhill 影像

(b)



額外加入 4 個浮水印之結果

correlation = 0.29

圖 35、原始 Goldhill 影像與額外加入 4 個浮水印之影像之對照 (a)原始影像 (b) 加入 4 個浮水印之破壞結果

實驗結果：

表格 8(a)是本論文的方法對多重浮水印攻擊之實驗結果。

表格 8(b)為參考論文[1]之實驗結果。由表(a)、(b)的比較中可以發現本文使用的方法可以允許額外嵌入 4 個浮水印，高於[1]的 3 個額外浮水印。在精確度上，當額外加 1、2 個浮水印時本文的 correlation 低於[1]的結果，不過在額外加入 3、4 個浮水印後，圖形品質較[1]的高(PSNR 較高，表示加入的強度較低)，且解出浮水印之精確度高於[1]。

(a)Proposed results

		1	2	3	4	5
Lena	ρ	0.78	0.56	0.39	0.24	0.14
	PSNR(dB)	34.05	31.55	29.71	28.35	27.14
Goldhill	ρ	0.76	0.54	0.34	0.29	0.15
	PSNR(dB)	34.86	32.37	30.47	29.09	27.94
Peppers	ρ	0.74	0.54	0.34	0.25	0.21
	PSNR(dB)	35.68	32.96	31.01	29.52	28.46

(b)Results from [1]

		1	2	3	4	5
Lena	ρ	0.65	0.41	0.27	0.11	NA
	PSNR(dB)	35.50	32.78	29.35	28.05	NA
Goldhill	ρ	0.79	0.45	0.31	0.18	NA
	PSNR(dB)	35.26	31.50	29.71	28.57	NA
Peppers	ρ	0.80	0.53	0.31	0.22	NA
	PSNR(dB)	34.53	31.99	30.19	28.81	NA

表格 8、Multiple watermark attack (a)Proposed results (b)Reference from [1]

5.7 Rotation and Scaling

本實驗之測試採用 StirMark [32]軟體進行測試，StirMark 是一套 watermark 測試之 benchmark，包含了各種形態的影像破壞。在表格 9(a)與(b)中 Rotation 表示 degree，正數的 degree 為順時鐘方向旋轉角度，負數的 degree 為逆時鐘方向旋轉角度。表格 9(a)與(b)分別是本論文之實驗結果與參考論文[1]結果之比較。

(a)Proposed results

Rotation		0.25	0.5	0.75	1	1.25	1.5	-0.25	-0.5	-0.75	-1	-1.25	-1.5
Lena	ρ	0.46	0.28	0.24	0.25	0.26	0.20	0.46	0.30	0.29	0.19	0.20	0.23
	PSNR(dB)	24.1	22.3	20.9	19.9	19.1	18.4	23.5	21.4	19.9	18.9	18.1	17.5
Goldhill	ρ	0.43	0.24	0.24	0.25	0.23	0.24	0.48	0.38	0.23	0.20	0.25	0.20
	PSNR(dB)	23.5	21.7	20.7	20.0	19.5	19.0	23.3	21.6	20.7	20.0	19.4	19.0
Peppers	ρ	0.50	0.37	0.35	0.26	0.20	0.12	0.49	0.30	0.31	0.29	0.21	0.14
	PSNR(dB)	23.2	21.5	20.1	19.1	18.3	17.6	23.1	21.0	19.6	18.5	17.7	17.0

(b)Reference from [1]

Rotation		0.25	0.5	0.75	1	1.25	1.5	-0.25	-0.5	-0.75	-1	-1.25	-1.5
Lena	ρ	0.37	0.29	0.26	0.24	NA	NA	0.32	0.23	0.24	0.16	NA	NA
Goldhill	ρ	0.33	0.24	0.21	0.15	NA	NA	0.38	0.27	0.25	0.14	NA	NA
Peppers	ρ	0.41	0.30	0.26	0.17	NA	NA	0.39	0.25	0.25	0.16	NA	NA

表格 9、Rotation and scaling attack (a)Proposed results (b)Reference from [1]

5.8 Filter convolution

本實驗是於 spatial domain 中，以不同作用的 filter 對影像進行破壞，其運作原理與 5.3 節 median filter 相似。實驗使用了兩組功能不用之 filter：

Gaussian Filter：空間域的 Gaussian filter 能達到影像模糊化的目的，實驗中使用的 mask 為 $\{1\ 2\ 1, 2\ 4\ 2, 1\ 2\ 1\}$ ，而 scaling factor 為 16。

Sharpening：增強影像的邊緣及黑白相間部份的訊號，所使用的 mask 為 $\{0\ -1\ 0, 1\ 2\ 1, 0\ -1\ 0\}$ 。

實驗結果：

(a)Proposed results

Filter \ Image	Lena	Goldhill	Peppers
Gaussian Filter	0.89	0.91	0.92
Sharpening	0.87	0.63	0.89

(b)Reference from [1]

Filter \ Image	Lena	Goldhill	Peppers
Gaussian Filter	0.64	0.56	0.74
Sharpening	0.46	0.39	0.62

表格 10、Multiple watermark attack (a)Proposed results (b)Reference from [1]

5.9 比較結果整理

在 5.1 節至 5.8 節的實驗結果，將本文的方法與[1]方法結果進行比較，表格 11 中”Yes”代表能夠偵測得浮水印，”No”表示無法偵測出浮水印，檢測 Lena、Peppers 及 Goldhill 三張影像是否同時可以抵抗各種影像破壞。由表格 11 可以看出在 JPEG(QF=25)、Median filter(5×5)、multiple watermark(5 watermarks)及 Rotation and scale (rotate = 1.0)之實驗中，本方法能於測試之三張圖形中找到浮水印，使得浮水印強韌性增加。

Attacks	[1]	Proposed
JPEG(QF = 30)	Yes	Yes
JPEG(QF = 25)	No	Yes
SPIHT(bitrate = 0.3)	Yes	Yes
Median Filter(4×4)	Yes	Yes
Median Filter(5×5)	No	Yes
Pixel shifting(type 1 , 9 pixels)	Yes	Yes
Pixel shifting(type 2 , 9 pixels)	Yes	Yes
Bitplane remove (4 LSBs)	Yes	Yes
Multiple watermark (5 watermarks)	No	Yes
Rotation and scale (rotate = 1.0)	No	Yes
Rotation and scale (rotate = -0.75)	Yes	Yes
Gaussian filter	Yes	Yes
Sharpening	Yes	Yes

表格 11、實驗結果比較表

六、結論與展望

本研究以小波轉換理論為基礎，以均勻量化方法將數位浮水印嵌於小波轉換後中頻係數組成之 Super Tree 中，並在 decoding 中利用 Super Tree Pair $P'_i(T'_{2i-1}, T'_{2i})$ 只有一棵 Tree 被量化的原理反推浮水印。並期許能順利解出已藏在影像中的版權資訊。

本文架構於參考論文[1]之上，於文章裡列舉了[1]中可能導致浮水印判斷之精確度不足之原因(Case 1 及 Case 2)，並提出改善方法。Minimum quantization step Δ_{\min} 的定義，確保 encoding 量化的區域與 decoding 取浮水印的區域將不會有不一致的情形發生，同時 super tree 的 magnitude 過小時則 Δ_{\min} 當作量化階層。於 decoder 中，Uniform deadzone scalar quantization 幫助浮水印每個位元之取得，使得取出浮水印的方法獲得改善，浮水印位元正確解出的機會提高。

此外，不同 filter bank 對於本浮水印演算法有很大的影響，如何選用適用的 filter bank 將對於解浮水印有正面的幫助。文中討論了兩個 filter bank 選擇之原則，使浮水印演算法具備強韌性且降低 $Te_i(j)$ 誤差之影響。

最後由實驗結果可以看出，本文提供了一個更強韌的浮水印技術，能有效承受於頻率域或空間域各種影像攻擊，達到資訊隱藏之目的，對於數位內容所有權保護能提供更高的精確度。

未來研究方向：

- 對於各種性質之 Filter bank 進一步實驗：

實驗中測試的 Filter 並不多，只定訂出選擇 Filter bank 之概略原則，實際上有許多 filter 可以進行討論，例如測試 orthogonal filters、semi-orthogonal filter、Cohen、Daubenchies 及 Feauveau 的 CDF 雙正交 (biorthogonal) 濾波器組合……等等。

同時因為各種不同 filter 對於 Super Tree 係數會產生不同影像，本研究使用的選擇 Filter bank 之概略原則並未對 Super Tree 係數有任何處理，因此未來可以試著對於 normalization，以試圖找出 Optimal solution。

- 實際建構一個線上 DRM 系統

數位浮水印儘能證明數位內容之版權，如果欲進一步實現數位內容保護，仍然得配合一套有效的數位智財權管理機制，並結合密碼學，以期許完成數位內容版權保護之目的。

參考論文

- [1]. S. H. Wang and Y. P. Lin, “Wavelet tree quantization for copyright protection watermarking”, *IEEE Trans. On Image Processing*, vol. 13, no. 2, pp. 154-165, Feb, 2004.
- [2]. <http://www.petitcolas.net/fabien/steganography/history.html>
- [3]. http://debut.cis.nctu.edu.tw/Epages/Research/e_stega.htm
- [4]. B. W. Lampson “A Note on the Confinement Problem”, *comm. ACM*, vol.16, NO.10, October 1973 , 613-615.
- [5]. U.S. Department of Defense. Trusted Computer System Evaluation “The Orange Book”. Publication DoD 5200.28-STD. Washington: GPO 1985
- [6]. Katzenbeisser, Stefan; Petitcolas, Fabien A.P., “Information hiding techniques for steganography and digital watermarking”, Boston : Artech House, 2000.
- [7]. Wu, Min; Liu Bede, “Watermarking for image authentication”, Image Processing, 1998. ICIP98. Proceedings. 1998 International Conference, Volume:2 , 4-7 Oct. 1998, p.437-441 vol.2.
- [8]. Kutter, M. ; Petitcolas, F. A. P. , “A fair benchmark for image watermarking systems”, Electronic Image '99. Security and Watermarking of Multimedia Contens, vol. 3657, Sans Jose, CA, USA, 25~27 January 1999. The International Society for Optical Engineering.
- [9]. Voloshynovskiy, S.; Pereira, S.; Pun, T.; Eggers, J.J.; “Attacks on digital watermarks: classification, estimation based attacks, and benchmarks”, Communications Magazine, IEEE, Volme: 39, Issue:8, Aug.2001, p.118-126.
- [10].I. Hontsch, L. J. Karam, and R. J. Safranek, “A perceptually tuned embedded zerotree image coder,” in Proc. IEEE ICIP, vol. 1, 1997, pp. 41-44.

- [11]. A. B. Watson, G. Y. Yang, A. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," IEEE Trans. Image Processing, vol. 6, pp. 1164-1175, Aug. 1997.
- [12]. Yiwei Wang, John F. Doherty and Robert E. Van Dyck, "A Wavelet-Based Watermarking Algorithm for Ownership Verification of Digital Images", IEEE Transactions on Image Processing, Vol. 11, No. 2, Feb 2002, pp. 77-87.
- [13]. 游光堯(1999), 「使用小波轉換的數位浮水印對數位影像資訊所有權確認之研究」, 國立交通大學資訊管理研究所碩士論文。
- [14]. 曾立信(2003), 「小波封包轉換的浮水印對數位影像所有權之確認」, 國立交通大學資訊管理研究所碩士論文。
- [15]. M.J. Tsai, K.Y. Yu, and Y.Z. Chen (2000), "Joint Wavelet and Spatial Transformation for Digital Watermarking" IEEE Transactions on Consumer Electronics, Volume: 46 Issue: 1, Feb. 2000., pp. 237.
- [16]. M.J. Tsai, K.Y. Yu, and Y.Z. Chen (2000), "Wavelet packet and adaptive spatial transformation of watermark for digital image authentication" Image Processing, 2000. Proceedings. 2000 International Conference, Volume: 1, 2000., pp. 450-453.
- [17]. I. J. Cox, J. Killian, T. Leighton, and T. Shanon (1997), "Secure Spread Spectrum Watermarking for Multimedia", IEEE Trans. Image Process, Vol.6, No.12, 1997, pp.1673-1678.
- [18]. R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne (1994), "A Digital Watermark", IEEE ICIP94, Vol.2, 1994.
- [19]. R. Wolfgang and E. Delp (1996), "A Watermark for Digital Images", IEEE ICIP96, Vol.3, Sep 1996.
- [20]. I. Pitas (1996), "A Method for Signature Casting on Digital Images", IEEE ICIP96, Vol.3, Sep 1996.

- [21].G. Voyatzis, and I. Pitas (1998), “Chaotic watermarks for embedding in the spatial digital image domain”, IEEE ICIP98, Vol.2, Oct 1998.
- [22].S. Walton (1995), “Image authentication for slippery new age”, Dr. Dobb’s J., pp.18-26 and 82-87, Apr 1995.
- [23].單維彰 (1999), 「凌波初步」, 民 88, 台北：全華科技圖書股份有限公司。
- [24].<http://www.cgan.com/science/publish/desktop/wavelet.htm>
- [25].Peter J. Burt and Edward H. Adelson, "The Laplacian Pyramid as a Compact Image Code," IEEE Transactions on Communication 31, no. 4 (1983) 532-540.
- [26].M. L. Miller, and J. A. Bloom. “Computing the Probability of False Watermark Detection”, in Proceedings of the Third International Workshop on Information Hiding, pp. 146-158, 1999.
- [27].D. Kundur and D. Hazinakos, “Digital watermarking, using multiresolution wavelet decomposition”, in Proc. IEEE ICASSP, vol.5, 1998, pp.2969-2972.
- [28].Yiwei Wang, John F. Doherty and Robert E. Van Dyck, “A Wavelet-Based Watermarking Algorithm for Ownership Verification of Digital Images”, IEEE Transactions on Image Processing, Vol. 11, No. 2, Feb 2002, pp. 77-87.
- [29].Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn “Information Hiding – a Survey”, proceedings of the IEEE, vol. 87, NO. 7 , July 1999.
- [30].S. Carver, N. Memon, B. L. Yeo, and M.M. Yeung, “Resolving rightful ownerships with invisible watermarking techniques : Limitation, attacks, and applications”, IEEE J. Select. Areas Commun., vol.16, pp.573-586, May 1998.
- [31].USC SIPI – The USC-SIPI Image Database [Online]. Available:
<http://sipi.suc.edu/services/database/Database.html>
- [32].http://www.petitcolas.net/fabien/software/StirMarkBenchmark_4_0_129.zip



附錄(一) Filter Banks

1. JPEG 2000 Filter Bank

H_0	H_1	G_0	G_1
0.026748757411	0	0	0.026748757411
-0.01686411844	0.091271763114	-0.091271763114	0.01686411844
-0.07822326653	-0.05754352623	-0.05754352623	-0.07822326653
0.266864118443	-0.59127176311	0.59127176311	-0.266864118443
0.602949018236	1.115087052457	1.115087052457	0.602949018236
0.266864118443	-0.59127176311	0.59127176311	-0.266864118443
-0.07822326653	-0.05754352623	-0.05754352623	-0.07822326653
-0.01686411844	0.091271763114	-0.091271763114	0.01686411844
0.026748757411	0	0	0.026748757411

2. 9-7 Filter

H_0	H_1	G_0	G_1
3.782845551e-02	0	0	-3.782845551e-02
-2.384946502e-02	-6.453888263e-02	-6.453888263e-02	-2.384946502e-02
-1.106244044e-01	4.068941761e-02	-4.068941761e-02	1.106244044e-01
3.774028556e-01	4.180922732e-01	4.180922732e-01	3.774028556e-01
8.526986790e-01	-7.884856164e-01	7.884856164e-01	-8.526986790e-01
3.774028556e-01	4.180922732e-01	4.180922732e-01	3.774028556e-01
-1.106244044e-01	4.068941761e-02	-4.068941761e-02	1.106244044e-01
-2.384946502e-02	-6.453888263e-02	-6.453888263e-02	-2.384946502e-02
3.782845551e-02	0	0	-3.782845551e-02

3. Orthogonal Filter

H_0	H_1	G_0	G_1
0.034148	-0.819964	0.819964	0.034148
-0.005255	0.126182	0.126182	0.005255
0.546777	0.107602	-0.107602	0.546777
-0.107602	0.546777	0.546777	0.107602
0.126182	0.005255	-0.005255	0.126182
0.819964	0.034148	0.034148	-0.819964

4. Haar Filter

H_0	H_1	G_0	G_1
0.7071067811865	0.7071067811865	0.7071067811865	-0.7071067811865
0.7071067811865	-0.7071067811865	0.7071067811865	0.7071067811865

5. 18-10 Filter

H_0	H_1	G_0	G_1
9.544158682e-04	0	0	9.544158682e-04
-2.727196297e-06	0	0	-2.727196297e-06
-9.452462998e-03	0	0	-9.452462998e-03
-2.528037294e-03	0	0	-2.528037294e-03
3.083373439e-02	2.885256501e-02	2.885256501e-02	3.083373439e-02
-1.376513483e-02	-8.244478228e-05	8.244478228e-05	-1.376513483e-02
-8.566118833e-02	-1.575264469e-01	-1.575264469e-01	-8.566118833e-02
1.633685406e-01	-7.679048885e-02	7.679048885e-02	1.633685406e-01
6.233596410e-01	7.589077295e-01	7.589077295e-01	6.233596410e-01
6.233596410e-01	-7.589077295e-01	7.589077295e-01	6.233596410e-01
1.633685406e-01	7.679048885e-02	7.679048885e-02	1.633685406e-01
-8.566118833e-02	1.575264469e-01	-1.575264469e-01	-8.566118833e-02
-1.376513483e-02	8.244478228e-05	8.244478228e-05	-1.376513483e-02
3.083373439e-02	-2.885256501e-02	2.885256501e-02	3.083373439e-02
-2.528037294e-03	0	0	-2.528037294e-03
-9.452462998e-03	0	0	-9.452462998e-03
-2.727196297e-06	0	0	-2.727196297e-06
9.544158682e-04	0	0	9.544158682e-04

附錄(二) 著作

- [1] 林承龍, “Uniform Wavelet Tree Quantization for Image Watermarking”, 第十六屆國際資訊管理學術研討會, 輔仁大學, p.167, May 2005.



Uniform Wavelet Tree Quantization for Image Watermarking

林承龍 交通大學資訊管理研究所

barrybounds.iim92g@nctu.edu.tw

摘要：

本研究提出一個使用均勻量化小波樹數(Uniform Wavelet Tree Quantization)之演算法於數位影像浮水印技術。本方法基於數位影像的小波頻率域，依浮水印位元數以均勻量化的方式，於的 Super Tree 中加入浮水印。浮水印的取出則利用 Super Tree pair(T'_{2n-1}, T'_{2n})中，只有一棵樹被量化過的統計特性，推論並解出浮水印位元。其中均勻量化(uniform quantizer)、最小量化階層(Minimum Quantization Step)與 Refinement 浮水印判斷程序的設計與使用能幫助在數位浮水印於萃取過程中，有效地減少浮水印位元誤判的次數。相較於非均勻量化之浮水印演算法(Non-uniform Quantization) 浮水印之強韌性大幅改進了。實驗結果可以看出此方法能有效承受於頻率域或空間域之各種影像攻擊，達到資訊隱藏之目的，對於數位內容所有權保護能提供更高的精確度。

關鍵字：數位浮水印、小波轉換、量化、uniform quantization

1. 簡介

1.1 數位智財權與數位浮水印技術

近年來，由於電腦科技發展的日新月異，各類和電腦相關的產品—不管是硬體方面的週邊設備或是軟體方面的各種軟體—都隨之變得越來越便宜，也越來越簡單易用。在這樣的一個趨勢之下，電腦網際網路的盛行，使得十分大量的資訊得以在網路上流通以及傳播。但由於數位資訊實際上只是許多 0 與 1 的組合，人們可以很輕易的在不失真的情形下將其完全複製。

在如此一個開放的網路環境中，若沒有配合相關的保護機制，那麼任何在網路上的資料都有可能輕易地被人們在未經正當授權的情形下，任意的複製以及傳播。也因此便有了所謂「數位著作權管理系統」的出現(DRMS, Digital Rights Management System)。數位著作權管理系統可自動化管理並在開放網路下發佈經交易的多媒體文件，並且考慮到能夠連結網路環境、協同合作以保護多媒體資料之智慧財產權的整體服務。

雖然數位影像浮水印技術尚不能解決所有智財權的問題。但大多數的研究者

均同意，它仍然具備強大的潛力吸引著作權所有者，提供可信賴的智財權保護機制。本研究基於上述之研究動機，希望能夠以小波轉換理論為基礎，利用小波轉換後的 wavelet tree 係數值之統計特性，於已嵌入浮水印的小波樹中，找出所嵌入之浮水印位元，並期待能夠將之應用於數位影像著作權利之保護之上。

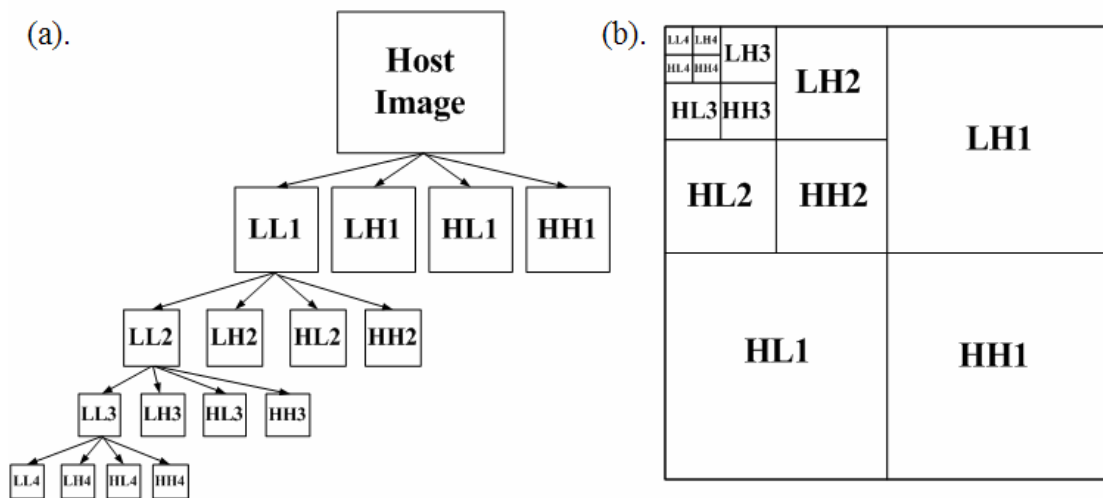
1.2 小波轉換

小波轉換使用將訊號通過分解濾波器組合(analysis filters)的方式來達到將訊號為分解為高頻訊號以及低頻訊號的目的。而分解濾波器組合中，又可細分為一低通濾波器 (low pass filter, 通常以 h_0 稱之) 以及一高通濾波器(high pass filter, 通常以 h_1 稱之)所組成。小波轉換於重建時則使用另外一組合成濾波器組合(synthesis filters), 而合成濾波器組合中，也同樣是由一組低通濾波器(g_0)和一組高通濾波器(g_1)所組成。但是以上這些濾波器組合並不是胡亂選取的，若要讓分解後的訊號能夠完美地重建(perfect reconstruction)成為原來的訊號，則濾波器組合必需要符合下列的條件：

$$h_0(-z)g_0(-z) + h_1(-z)g_1(-z) = 0$$

$$h_0(z)g_0(z) + h_1(z)g_1(z) = 2$$

我們可以使用一樹狀結構來記錄這些次頻道間的分解關係(如圖一(a))，並且我們通常會將其組合成類似圖一(b)的影像以便作為展示以及記錄之用。



圖一、小波轉換示意圖 (a). 將 host image 進行小波分解，共分解成四個 level，每個 level 包含了 LL, LH HL 與 HH 四個子頻道。 (b). 經(a)分解後，實際上每個子頻道於圖形中相對應之位置。

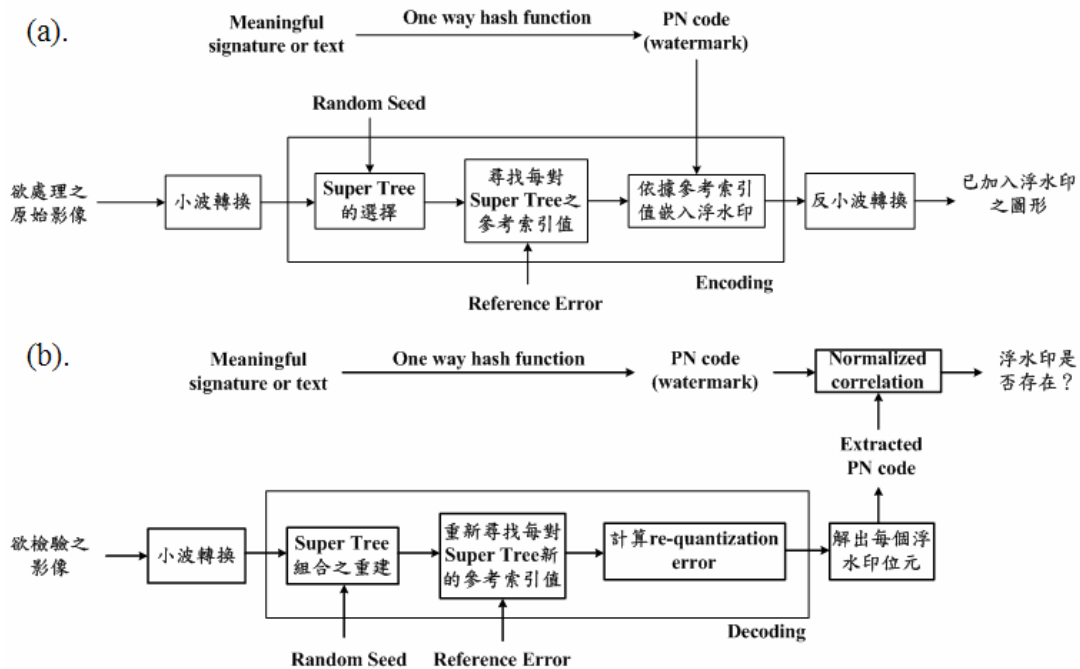
1.3 文獻探討

1997 年，Cox 等人提出了”Secure spread spectrum watermarking for multimedia”，將數字序列浮水印加入訊號的 DCT(離散餘旋轉換)展頻通道中，開啟了以數位浮水印實現資訊隱藏之先河，由於頻率域的嵌入法具有較高強韌性及安全性，因此接下來學者研究則著重於離散餘旋轉換、傅立業轉換及小波轉換技術。由 Wang 與 Lin 提出的小波轉換非均勻量化技術[1,2]中，顯現了量化 Super

Tree 方法對於高度提升了浮水印隱藏強度，並根據 Super Tree 經量化後統計特性之差異，將浮水印解出。本文之均勻量化小波樹演算法基於[1,2]的方法，分析了非均勻量化技術潛在的缺點，並提出能減少浮水印位元誤判之方法，改進浮水印強度，進而增加浮水印於數位版權保護之可靠性。

2. 小波樹量化與浮水印技術

本文所使用的浮水印技術是基於對小波轉換後的係數做處理，如圖二所示。在圖二(a)中首先一開始必需要欲處理之原始影像經過小波分解，其係數再經過 encoding 的過程做浮水印之嵌入動作。整個 encoding 的包括了 Super Tree Selection、參考索引(reference index)使用的尋找以及均勻量化嵌入浮水印三大步驟，這個部份將在 3.1 節中詳細說明之。比較值得注意的是本論文採用的浮水印，是一串 ± 1 的數字序列，我們稱它為 PN code[1]。此 PN code 可以由能夠證明版權的 logo、使用者與擁有者之描述或圖形相關資訊，經過一個 one way hash 後得到的，在 encoding 的第三步驟中加入影像中。當所有資訊加入影像後，再以小波重組使圖形轉換回空間域中，達到浮水印嵌入目的。



圖二、浮水印嵌入、取出流程 (a).加入浮水印。(b).浮水印萃取。

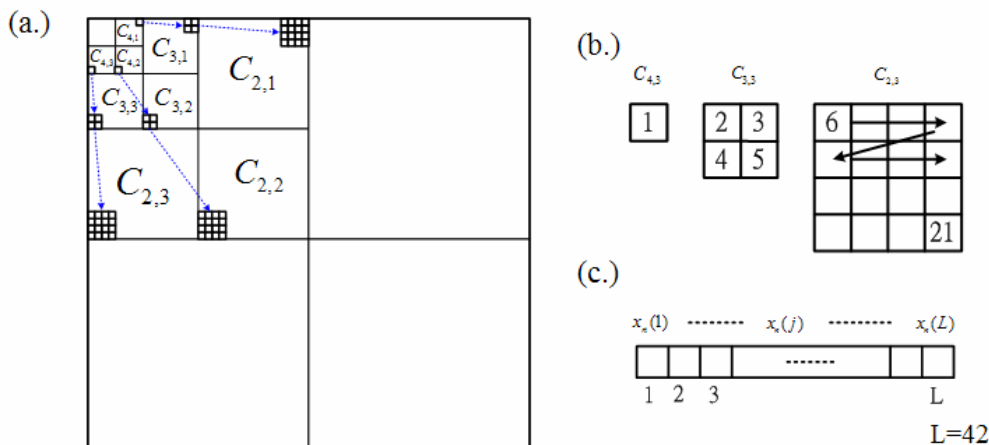
對於一張影像，我們想要驗證其中是否含有特定的版權證明或其他資訊時，則必需利用浮水印萃取方法來達成。此部份將在 3.2 節說明演算法的運作。當浮水印解出後，必須與原始的 PN code 進行比對，才能達到版權驗證的效果，而 Normalized Correlation 可做為此一比對之方法。一旦 $\rho(W, W')$ 大於一定的 Threshold，那麼就可以證明 W 與 W' 是相同的，版權因此而確認。

3. 演算法之設計

3.1 浮水印嵌入方法之設計

3.1.1 小波樹與 Super Tree

嵌入浮水印的第一步是將圖形經小波做四層分解，如圖三(a)所示。我們以 $C_{i,j}$ 來表示經過分解後的子頻道，其中 i 為子頻道所在的Level， j 為每個Level中的編號。例如 $C_{3,2}$ 為Level 3 裡面的第二個子頻道。接下來我們定義 **小波樹** 是由一個Level4 的係數、四個Level3 的係數及十六個Level2 的係數所組成的，其中Level3 的四個係數為Level4 的child，Level2 的十六個係數為四個Level3 的係數Child。每棵小波樹中包含有 21 個係數值，如圖三(b)所示，我們將此 21 個係數依照順序串起來。在一張 512×512 的圖形中小波樹的個數為 $3 \times 32^2 = 3072$ ，也就是相當於Level4 的三個子頻道之係數總個數(Level4 子頻道的邊長為 32)。



圖三 (a).四層小波分解。 (b). 小波樹組成之係數 (c). 一棵 Super Tree

接下來我們定義一個**Super Tree T** 為任意選取的兩個小波樹組成的。在選擇小波樹時，我們使用一個random seed幫助選擇如圖二(a)。在decoding由於必須還原在encoding的Super Tree，因此此seed必須記錄起來，以供decoding 過程的使用。一個super tree T 中共含有 42 個係數($L=42$ 於圖三(c))，且總共有 $3 \times 32^2 / 2 = 1536$ 棵Super Tree於此影像中。

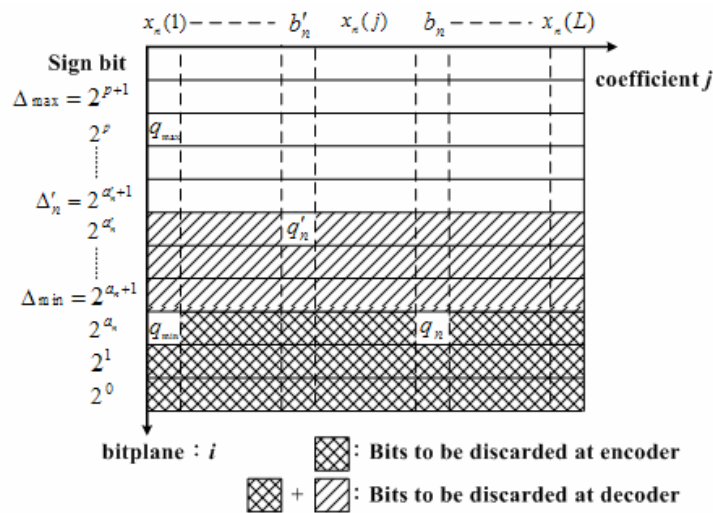
3.1.2 參考索引值之尋找

令浮水印長度為 N_w (小於 768)，在浮水印嵌入之前，我們選取一組 Super Tree pair (T_{2n-1}, T_{2n}) ，用來嵌入第 n 個浮水印位元。假設浮水印為-1 則我們對 T_{2n-1} 量化，如果浮水印位元為 1，則將對 T_{2n} 量化，量化方法將來 3.1.3 節討論。在做量化之前，我們必須先預測 Super Tree 量化的程度。如果我們想讓浮水印更強韌那

麼勢必增加量化的量，參考索引值(Reference Index)就是為了找到一個量化階層(Quantization Step)所設計的，有了量化階層才能在 3.1.2 節進行量化。

Super Tree 最多有 1536 棵，首先定義第 n 棵 Super Tree 中第 j 個係數值定義為 $x_n(j)$ ，其中 $1 \leq j \leq L$ 且 $1 \leq n \leq 2N_w$ 。於圖四中以位元平面表示法(bit plane representation)將一個整數以二元型態程現，而在平面中的座標則以 (a_n, b_n) 來表示。最低有效位元(LSB)之位元平面為 2^0 ，最高有效位元(MSB)之位元平面定義為 2^p 。另外，符號位元(sign bit)位於最高有效位元平面之上方。假定給予一定的參考誤差(reference error)的量，那麼我們隨即根據此參考誤差，尋找參考索引值，以確定量化階層。

首先將 Super Tree pair (T_{2n-1}, T_{2n}) 均以位元平面表示，接下來由平面的左下角開始，由右至左由下到上，進行能量的累積，當達到 T_{2n-1} 中能量的累積達到參考誤差值而且 T_{2n} 中能量的累積亦達到參考誤差值時，此時 q_n 就是我們要找的參考索引。



圖四、Super Tree 均勻量化，於 Encoding 及 Decoding 中。

3.1.3 均勻量化與浮水印嵌入

找到此一量化索引值 q_n ，開始進行量化。為確保量化每一個 Super Tree pair 的量化階層都不會太小，所以定訂一個最小量化索引值 q_{min} ，凡小於 q_{min} 之索引值均強制設定為 q_{min} 。量化值定義為 $Q[x_n(j)]_{q_n} = round(x_n(j))_{\Delta_n(j)}$ ，其中量化階層大小 $\Delta_n(j)$ 定義為：對於所有的 j ， $\Delta_n(j) = 2^{a_n+1}$ ，亦即 q_n 所在位置左側座標的上一層。如果第 n 個浮水印位元 w_n 為 -1，則 super tree T_{2n-1} 依其對應之 q_n 被量化。反之， T_{2n} 被量化。也就是說，所有低於、等於量化索引值所在位元平面的位元，都將因量化而拋棄。這些被丟棄的位元於圖四的網狀區域中顯示。在第 n 個 tree

中第 j 個係數之量化誤差依其對應之 q_n 定義為 $e_n(j) = Q[x_n(j)]_{q_n} - x_n(j)$ 。

Super Tree pair (T_{2n-1}, T_{2n}) 其中一棵樹將會被依據相對應的第 n 個浮水印位元被量化，因此達到加入浮水印的目的。若未來在 Decoding 中想找出此浮水印位元，我們只要想辦法知道 (T_{2n-1}, T_{2n}) 哪一棵 Tree 被量化過，這樣浮水印就能解出。

3.2 浮水印取出方法之設計

在解浮水印的整個過程中，根據圖二(b)可知我們將待測的圖形經過小波轉換，接下來利用與 encode 相同的亂數種子，找到我們在 encode 中的 super tree 順序，以利 decode 中能正確找出浮水印。而在 decode 找到的 super tree 此時以 T'_l 來表示，其中 l 的範圍由 1 到 1536，並將它們配對成 n 個 super tree pair (T'_{2n-1}, T'_{2n}) ， $n=1\sim 768$ 。

3.2.1 非均量化浮水印技術之缺點

在參考論文[1]中提到的解浮水印方法，主要是對於每一個 super tree pair (T'_{2n-1}, T'_{2n}) ，在一定的 reference error 之下，尋找每個 pair 的 re-quantization index q'_n 。找 q'_n 的做法與 encode 相似，預測 T'_{2n-1} 誤差的累積量或 T'_{2n} 誤差的累積量達到 reference error 時停止，進而發現 q'_n 。一旦有了 q'_n ，那麼隨即對 T'_{2n-1} 與 T'_{2n} 進行 requantized error 之統計。理論上來說 requantized error $(\varepsilon_{2n-1}(q'_n), \varepsilon_{2n}(q'_n))$ 一定會有一個的數值大小的分佈是比較小的，如此便可以猜測何者在 encode 中被量化過，因此藏在 (T'_{2n-1}, T'_{2n}) 中的浮水印位元便被找出，被量化過的 super tree 其符合 $|e'_l(j)/\Delta'_l(j)| < \varepsilon$ ($j=1\sim 42$) 之個數大於未被量化過的另一棵 super tree。例如： T'_{2n-1} 中符合 $|e'_{2n-1}(j)/\Delta'_{2n-1}(j)| < 0.1$ 共有 38 組，而 T'_{2n} 中符合 $|e'_{2n}(j)/\Delta'_{2n}(j)| < 0.1$ 僅有 20 組，那麼我們即認定浮水印藏於 T'_{2n-1} 中。不過此一方法有隱含了兩個潛在的問題，使得浮水印位元的尋找出現誤判，如下說明之：

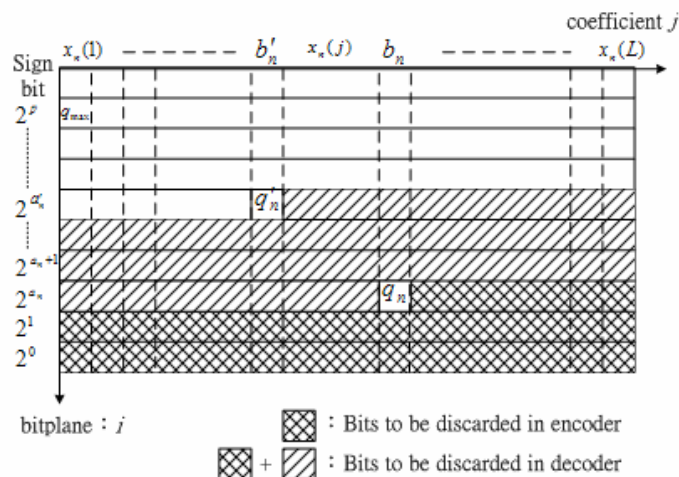
Case 1：(當 $q'_n > q_n$ 時發生)

一般而言演算法期望 decode 找到的 q'_n 能夠小於等於 q_n 是最理想的情形，此時能完美找出 (T'_{2n-1}, T'_{2n}) 何者被量化過。但是當 $q'_n > q_n$ 時，如圖五所示，中間斜線為誤差的部份這些誤差的位元裡有些為 1，有些為 0，然而值為 1 的位元對浮水判斷構成大的誤判， q'_n 與 q_n 相差量愈大則誤差愈高。

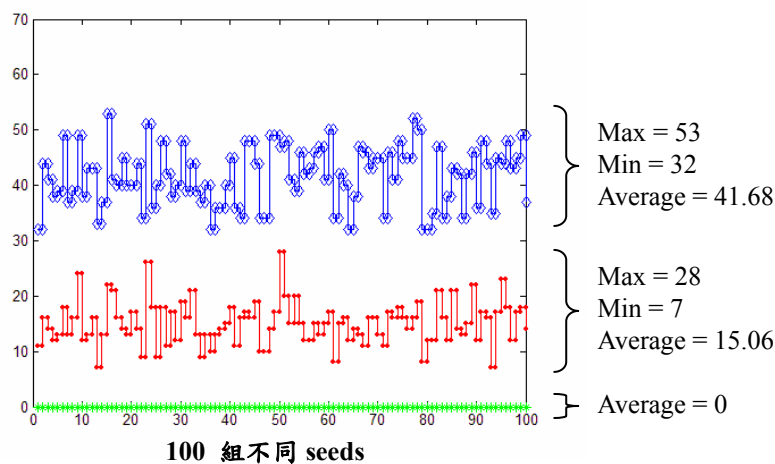
Case 2：(當 $q'_n = q_{\max}$ 時發生)

$q'_n = q_{\max}$ 的發生有兩個主因。其一為 encode 時， (T_{2n-1}, T_{2n}) 中有一棵 Super

Tree 的 magnitude 小於我們所訂定之參考誤差。其二為 decode 時 (T'_{2n-1}, T'_{2n}) 之 magnitude 均小於參考誤差。當 $q'_n = q_{\max}$ ，則 $\Delta'_i = \Delta_{\max}$ 。因為 $(\varepsilon_{2n-1}(q'_n), \varepsilon_{2n}(q'_n))$ 均非常小且 $\Delta'_i = \Delta_{\max}$ ，當判別浮水印時參考論文[1]使用 $|e'_i(j')/\Delta'_i(j')| < \varepsilon$ 來定判定，很有可能造成 (T'_{2n-1}, T'_{2n}) 符合 $|e'_i(j')/\Delta'_i(j')| < \varepsilon$ 之個數相同，而誤判因而產生。



圖五、在 encoder 與 decoder 中可能發生的量化結果， $q'_n > q_n$ 則中間區域即產生誤判。



圖六、以 100 組 seeds 為測試，上方藍色線條表示 $q'_n = q_{\max}$ 發生次數，中間紅色線條表示在 $q'_n = q_{\max}$ 之下，浮水印位元發生誤判次數。下方綠色線條是改用均勻量化演算法後，將誤判情形改善之結果。

如圖六所示，我們針對 100 random seeds，上方藍色線條為發生 $q'_n = q_{\max}$ 的次數統計，中間紅色線條為發生 $q'_n = q_{\max}$ 且浮水印位元誤判之次數。很清楚地看到在未受到任何攻擊之情況下平均會有 15 個浮水印位元有錯誤判斷。因此這個部份是有很大的改善空間，特別是如果影像遭受惡意攻擊後，誤判之情形會更為嚴重。綠色部份為改用均勻量化演算法改善後的結果，可以看到已將此誤判情形改善，並不會有誤判產生(平均誤判組數為 0)，詳細演算法將在 3.2.2 節提出。

3.2.2 Uniform Quantization for Watermarking

為了尋找藏在 (T'_{2n-1}, T'_{2n}) 的浮水印位元，並改善在 3.2.1 節中提出之誤判狀況，使用 Uniform Quantization 是一個很好的解決辦法。使用均勻量化方法能讓 case1 的發生的情況降到最低，減少圖五中間斜線部份面積；同時引入 Δ_{\min} 及 **Refinement 程序**，確保不會發生 case2 的情況。不論在 Encoding 或 Decoding 中，當量化階層太小時，表示 (T'_{2n-1}, T'_{2n}) 的 Magnitude 太小，因此我們以 Δ_{\min} 當做量化階層，所以不會發生 Case 2 的情形。

利用 Uniform Quantization 尋找藏在 (T'_{2n-1}, T'_{2n}) 的浮水印位元，應進行尋找 Re-quantization Index 及 Re-quantization Error 等工作，最後驗證解出之浮水印 W' 與原浮水印 W 之相關性。

尋找 Re-quantization Index

給予一定的參考誤差(reference error)的量(與 Encoding 相同)，接著根據此參考誤差對 (T'_{2n-1}, T'_{2n}) 尋找參考索引值，以確定在 Decoding 中的量化階層。將 Super Tree pair (T'_{2n-1}, T'_{2n}) 均以位元平面表示，並由平面的左下角開始，由右至左由下到上，進行能量的累積，當達到 T'_{2n-1} 中能量的累積達到參考誤差值或 T'_{2n} 中能量的累積亦達到參考誤差值時，此時 q'_n 就是我們要找的參考索引。並確定量化階層為 $2^{a'_n+1}$ 。

Re-quantization Error

實際對 (T'_{2n-1}, T'_{2n}) 進行量化，並計算 Requantization Error：

$$e'_l(j) = x'_l(j) - Q[x'_l(j)]_{2^{a'_n+1}}$$

其中， $x'_l(j)$ 代表在 T'_l 中的第 j 個係數值。

$Q[x'_l(j)]_{2^{a'_n+1}}$ 則是 $x'_l(j)$ 經過 $2^{a'_n+1}$ 量化後的值。

浮水印之解出：

計算出 (T'_{2n-1}, T'_{2n}) 中的誤差統計量 $(\varepsilon_{2n-1}(q'_n), \varepsilon_{2n}(q'_n))$ 之後，定義 N_{2n-1} 表示在 T'_{2n-1} 中，符合 $|e'_{2n-1}(j)/\Delta'_{2n-1}(j)| < y$ 之係數個數， N_{2n} 表示在 T'_{2n} 中，符合 $|e'_{2n}(j)/\Delta'_{2n}(j)| < y$ 之係數個數。而解出之浮水印 w'_n 如下：

$$w'_n = \begin{cases} -1, & N_{2n-1} > N_{2n} \\ 1, & N_{2n-1} < N_{2n} \\ \text{Re-} \text{finement}, & N_{2n-1} = N_{2n} \end{cases}$$

Refinement 能解決 $N_{2n-1} = N_{2n}$ 發生誤判的問題。我們比較 $e'_{2n-1}(j')$ 與 $e'_{2n}(j')$ 的大小 ($j=1\sim 42$)，如果 $e'_{2n-1}(j') < e'_{2n}(j')$ 的個數較多，那麼我們判定此浮水印位元為 -1，反之此位元為 1。

3.2.3 浮水印之驗證：

當所有的浮水印位元均經過演算法解出以後，為了判別解出的浮水印與原浮水印是否相同，我們使用 Normalized Correlation Coefficient 做為相關性的判斷 [7]，並比較與相關性門檻值 ρ_T 之大小，以宣稱浮水印的存在。Normalized Correlation Coefficient 之定義如下：

$$\rho(W, W') = \frac{\sum_{m=1}^{N_w} w_m w'_m}{\sqrt{\sum_{m=1}^{N_w} w_m^2 \sum_{m=1}^{N_w} w'_m^2}}$$

根據 []，可以得到 False Positive Probability 為：

$$P_f = \sum_{n=\lceil N_w \times (\rho_T + 1) / 2 \rceil}^{N_w} \binom{N_w}{n} P_E^{N_w - n} \cdot (1 - P_E)^n = \sum_{n=\lceil N_w \times (\rho_T + 1) / 2 \rceil}^{N_w} \binom{N_w}{n} 0.5^{N_w}$$

由上式可以看出，兩個 PN code 的 False positive probability 跟浮水印長度及門檻值有極大的關係，在選定門檻值與浮水印長度的時，為了建立與 [1] 相同的比較基礎，所以在本文的實驗，使用 512 的浮水印並設定 $\rho_T = 0.23$ ，在 $P_f = 1.03 \times 10^{-7}$ 底下進行比較。較完整的 ρ_T 與 N_w 關係如表一所示。

$N_w \backslash \rho_T$	0.15	0.2	0.23	0.25
768	1.61x10 ⁻⁵	1.5x10 ⁻⁸	1.13x10 ⁻¹⁰	2.14x10 ⁻¹²
512	4.5x10 ⁻⁴	3.78x10 ⁻⁶	1.03x10 ⁻⁷	8.45x10 ⁻⁹

表一、不同的 ρ_T 與 N_w 之 False positive probability

4. 浮水印嵌入與取出之演算法

4.1 詳細浮水印嵌入演算法

Step 1：將能夠證明版權的 logo、使用者與擁有者之描述或圖形相關資訊，經過一個 one way hash 後得 ± 1 的數字序列 (PN code)，以此 PN code 為浮水印。

Step 2：以小波轉換計算出頻率域係數值，並以一個虛擬亂數的方式，將兩個小波樹結合成為一個 super tree T_k ，其中 $k = 1, \dots, 2N_w$ 。設定 $n = 1$ 。

Step 3：如果 $M(T_{2n-1}) < \varepsilon$ 且 $M(T_{2n}) < \varepsilon$ ，則 $\Delta_n = \Delta_{\min}$ 並且移至 Step 6，否則移至 Step 4。

Step 4：設定 $q_n = 1$ ， $\varepsilon_{2^{n-1}}(1) = 0$ 以及 $\varepsilon_{2^n}(1) = 0$

Step 5：while $((\varepsilon_{2^{n-1}}(q_n) < \varepsilon) \text{ or } (\varepsilon_{2^n}(q_n) \geq \varepsilon) \text{ and } q_n < q_{\max})$ 計算 $\varepsilon_{2^{n-1}}(q_n)$ 與 $\varepsilon_{2^n}(q_n)$ 並設定 $q_n = q_n + 1$ 。

Step 6：如果 $2^{a_n+1} \leq \Delta_{\min}$ 則 $\Delta_n = \Delta_{\min}$ ，否則 $\Delta_n = 2^{a_n+1}$ (如 Fig. 1(c) 所顯示)

Step 7：如果 $w_n = -1$ 則對 $T_{2^{n-1}}$ 量化，否則對 T_{2^n} 做量化。

Step 8：設定 $n = n + 1$ ，如果 $n < N_w$ ，則回到 **步驟二**。

Step 9：將已修改過小波係數，經由反小波轉換得到空間域，得到一張已嵌入浮水印的影像。

4.2 詳細浮水印取出演算法

Step 1：同嵌入過程中的 Step 1。

Step 2：計算欲解浮水印影像中，每個像素的位元數 $b(\text{bits/pixel})$ 。以小波轉換計算出頻率域係數值，並以一個虛擬亂數的方式，將兩個小波樹結合成為一個 super tree T_k ，其中 $k = 1, \dots, 2N_w$ 。設定 $n = 1$ 。

Step 3：如果 $M(T'_{2^{n-1}}) < \varepsilon$ 且 $M(T'_{2^n}) < \varepsilon$ ，則 $\Delta'_n = \Delta_{\min}$ 並且移至 **Step 7**，否則移至 **Step 4**。

Step 4：設定 $q'_n = 1$ ， $\varepsilon_{2^{n-1}}(1) = 0$ 以及 $\varepsilon_{2^n}(1) = 0$

Step 5：while $((\varepsilon_{2^{n-1}}(q'_n) < \varepsilon) \text{ and } (\varepsilon_{2^n}(q'_n) \geq \varepsilon) \text{ and } q'_n < q_{\max})$ 計算 $\varepsilon_{2^{n-1}}(q'_n)$ 與 $\varepsilon_{2^n}(q'_n)$ 並設定 $q'_n = q'_n + 1$ 。

Step 6：如果 $2^{a'_n+1} \leq \Delta_{\min}$ 則 $\Delta'_n = \Delta_{\min}$ ，否則 $\Delta'_n = 2^{a'_n+1}$ 並移至 **Step 7**。

Step 7：重新量化與計算 $N_{2^{n-1}}$ 及 N_{2^n} 。如果 $N_{2^{n-1}} = N_{2^n}$ 則並移至 **Step 9**，否則移至 **Step 8**。

Step 8：如果 $N_{2^{n-1}} > N_{2^n}$ 則 $w'_n = -1$ 否則 $w'_n = 1$ 並移至 **Step 10**。

Step 9：For ($j=1$ to SuperTree_length)

If $|x_{2^{n-1}}(j)| < |x_{2^n}(j)|$ then $N'_{2^{n-1}}++$ Else $N'_{2^n}++$

如果 $N'_{2^{n-1}} > N'_{2^n}$ 則 $w'_n = -1$ 否則 $w'_n = 1$

Step 10：設定 $n = n + 1$ ，如果 $n < N_w$ ，則回到 Step 2。

Step 11：計算 normalized correlation ρ 。

Step 12：如果 ρ 值大於門檻值 ρ_T ，則表示浮水印存在；否則此驗證之圖形中不存在浮水印。

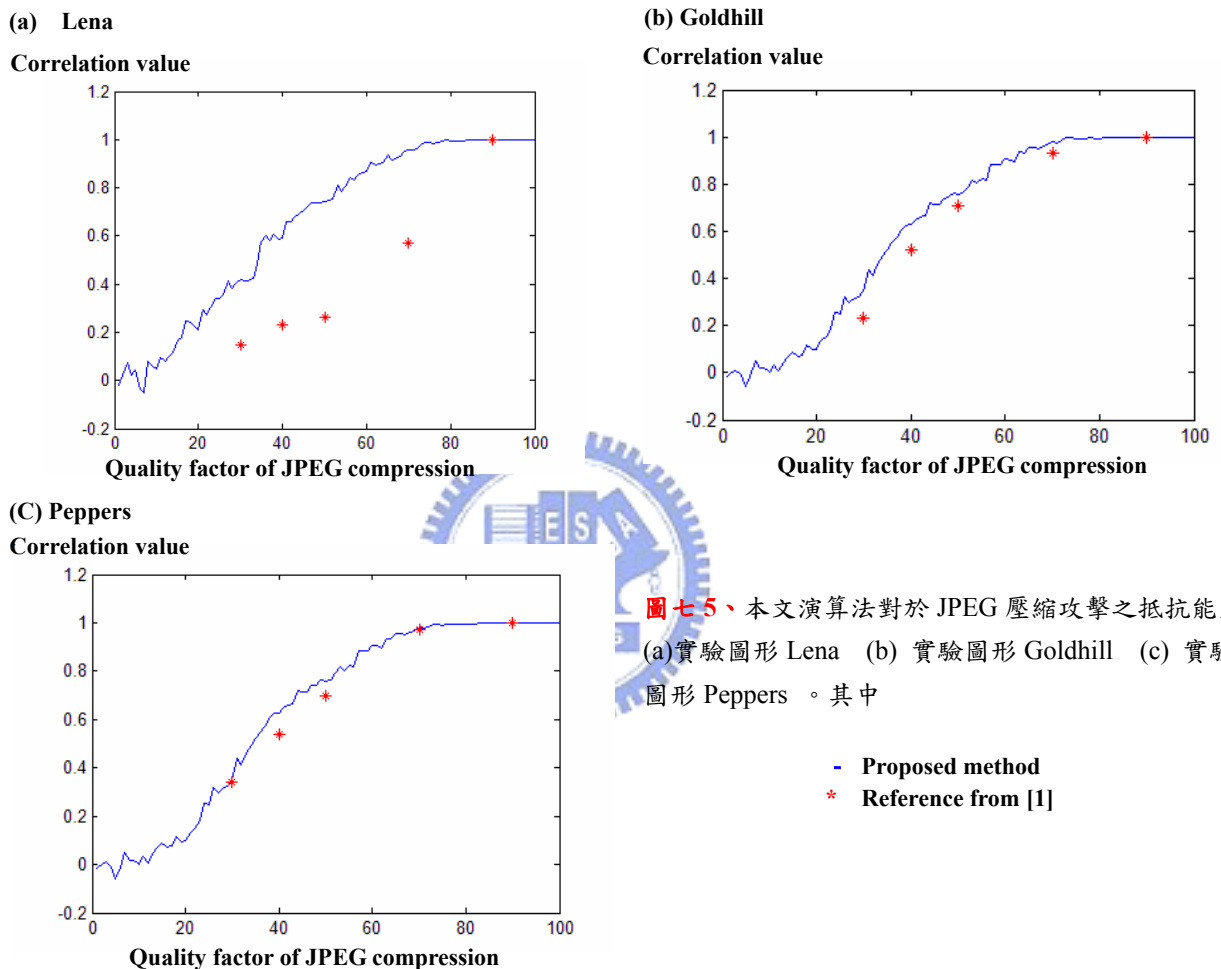
5. 實驗結果與討論

在實驗中我們使用了灰階 512x512 的 Lena、Goldhill 和 Peppers 圖形作為實驗主要測試圖形，加入浮水印後的 PSNR 分別限定在 38.2、38.7 和 39.8dB，取

得與[1]之比較的基礎。此外 Super Tree 選取的組數為 1536，浮水印長度為 768，參考誤差值為 70，最小量化階層為 8，最大量化索引值為 336。本方法在 Encoding 與 Decoding 的量化階層有 70%的比例位在 8，其他比例位在 16 左右，因此 Encoding 與 Decoding 能減少不必要的誤判情形。

5.1 JPEG Compression

圖七中，顯示出了本文提出的方法明顯地改善已加入浮水印的 Lena 圖形受



圖七5、本文演算法對於 JPEG 壓縮攻擊之抵抗能力 (a)實驗圖形 Lena (b) 實驗圖形 Goldhill (c) 實驗圖形 Peppers。其中

到 JPEG 壓縮攻擊之承受能力。當 JPEG 之品質係數達小至 30，本文所提出方法仍然能判別浮水印的存在，不過使用非均勻量化之浮水印演算法無法達到 [1]。

5.2 Median Filter :

Median filter 為一般影像處理常用之攻擊，它能夠使影像產生模糊化的效果，進而移除影中浮水印，當 Mask size 愈大時模糊化的程度愈大，表二為本方法對於 Median filter 的抵抗程度，其中 ρ 為本文之 correlation， $\bar{\rho}$ 為 [1] 之結果。

(a) Lena

Mask	2*2	3*3	4*4	5*5	6*6
ρ	0.48	0.82	0.33	0.38	0.19
$\bar{\rho}$	0.38	0.51	0.23		

(b) Goldhill

Mask	2*2	3*3	4*4	5*5	6*6
ρ	0.49	0.79	0.30	0.31	0.19
$\bar{\rho}$	0.35	0.56	0.24		

(c) Peppers

Mask	2*2	3*3	4*4	5*5	6*6
ρ	0.58	0.87	0.36	0.42	0.21
$\bar{\rho}$	0.46	0.71	0.25		

表二、本文演算法對於 Median filter 之抵抗能力

5.3 Bitplane Remove Attack

在實驗中所使用的 Lena、Peppers、Goldhill 中，每個像素是由八位元所組成的灰階影像。Bitplane Remove 的方法是將影像中，每個像素的 LSB(Least Significant Bits)移除，以達到破壞影像的目的。表三(a)為本方法經過移除 LSB 之後之結果，(b)為參考論文[1]之結果。

(a) Proposed Results

移除位元	1	2	3	4	5
Lena	1	1	0.98	0.74	0.18
Peppers	1	1	1	0.77	0.19
Goldhill	1	1	0.99	0.70	0.10

(b) Results from [1]

移除位元	1	2	3	4	5
Lena	1	1	0.99	0.52	0.11
Peppers	1	1	0.97	0.38	0.14
Goldhill	1	1	0.99	0.70	0.10

表三、經過 Bitplane Remove Attack 之實驗結果 (a).Proposed Results (b).對照論文[1]之結果

6. 參考文獻

1. S. H. Wang and Y. P. Lin, "Wavelet tree quantization for copyright protection watermarking", *IEEE Trans. On Image Processing*, vol. 13, no. 2, pp. 154-165, Feb, 2004.
2. S. H. Wang and Y. P. Lin, "Blind watermarking using wavelet tree quantization", in Proc. *IEEE ICME*, vol. 1, 2002, pp.589-592.
3. I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol.6, pp.1673-1687, Jan. 1997.
4. Yiwei Wang, John F. Doherty and Robert E. Van Dyck, "A Wavelet-Based Watermarking Algorithm for Ownership Verification of Digital Images", *IEEE Transactions on Image Processing*, Vol. 11, No. 2, Feb 2002, pp. 77-87.
5. 單維彰 (1999), 「凌波初步」, 民 88, 台北: 全華科技圖書股份有限公司。
6. G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video", *IEEE Trans. Image Processing*, vol.10, pp.148-158, Jan. 2001.
7. D. Kundur and D. Hazinakos, "Digital watermarking, using multiresolution wavelet decomposition", in Proc. *IEEE ICASSP*, vol.5, 1998, pp.2969-2972.