

國立交通大學

管理學院(資訊管理學程)碩士班

碩 士 論 文

以動態能量差異化為基礎之小波樹數位影像浮水印

**Dynamic Energy Enabled Differentiation (DEED)
Based Wavelet Tree Watermarking for Digital Images**

研 究 生： 陳 白 駿

指 導 教 授： 蔡 銘 箴 博 士

中 華 民 國 九 十 五 年 六 月

以動態能量差異化為基礎之小波樹數位影像浮水印
Dynamic Energy Enabled Differentiation (DEED) Based
Wavelet Tree Watermarking for Digital Images

研究生：陳白駿

Student : Bai-Jiun Chen

指導教授：蔡銘箴

Advisor : Min-Jen Tsai

國立交通大學

管理學院（資訊管理學程）碩士班



Submitted to Institute of Information Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

in

Information Management

June 2006

Hsinchu, Taiwan, the Republic of China

中華民國九十五年六月

以動態能量差異化為基礎之小波樹數位影像浮水印

學生：陳白駿

指導教授：蔡銘箴

國立交通大學資訊管理研究所

摘要

隨著網際網路的發達以及資訊數位化技術的進步，人們可以輕易的對數位資訊進行編輯、複製，並透過網路快速的傳送至世界各地。然而，資訊科技的進步卻也產生了一些安全性的問題，其中包含各種數位內容的非法複製、散佈或提供不被授權的下載等侵害智慧財產權的情事發生。



在數位內容的智財權保護方法中，數位浮水印技術可以將數位內容的版權資訊以浮水印方式嵌入數位內容中，當產生版權爭議時，即可將浮水印透過特定機制解讀出來，以作為智財權認定上的證明依據，使數位內容的版權受到保障。

本論文是以小波轉換理論為基礎，動態改變頻率域影像之小波係數，藉由係數之間特定的差異化以達到浮水印嵌入之目的。此外更利用影像切割重組之方法，有效提升浮水印抵抗空間域幾何攻擊之能力，進而建構出更具安全性與強韌性之小波樹數位影像浮水印演算法，提供更有效的數位智財權保護技術。

關鍵字：

動態能量差異化、資訊隱藏、數位影像浮水印、小波轉換、小波樹

Dynamic Energy Enabled Differentiation (DEED) Based Wavelet Tree Watermarking for Digital Images

Student : Bai-Jiun Chen

Advisor : Min-Jen Tsai

Institute of Information Management
National Chiao-Tung University

ABSTRACT

Because of the development of Internet and digitization of information technique, people can easily edit, copy, and transmit the digital information to any place of the world. But the progress of information technique also causes some problems about security, including the illegal copies, transmission, and unauthorized downloads of all kinds of digital content that infringe the intellectual property rights.

Among all the methods for digital content copyright protection, digital watermarking technique can embed the copyright information into the digital content with watermarks. If any dispute of copyright occurs, we can extract the watermark through specific procedure to distinguish it and make the copyright of digital content to be protected.

This thesis is based on wavelet transformation theory. We modify the wavelet coefficient values of images in frequency domain dynamically and use the differentiation to embed the watermark. We also use the image cutting-and-assembling mosaic method to improve the ability of resistance to malicious geometric attacks, and make our method more robust and securer.

Key words:

Dynamic Energy Enabled Differentiation, DEED, Digital Image Watermarking,
Information hiding, Wavelet Transform, Wavelet Tree

誌謝

本篇論文之所以能夠順利完成，首先必須要感謝我的論文指導教授蔡銘箴老師，不論是在演算法開發、實驗設計、結果分析或是論文寫作階段，蔡老師都給予我非常多的協助與指導，使我能夠很快的進入數位影像浮水印的研究領域。此外也必須感謝論文口試委員林源倍老師、林妙聰老師、李永銘老師所給予的各項建議與指正，使本篇論文的內容能夠更趨於嚴謹。

兩年的碩士生涯很快就過去了，在這兩年求學的過程中，特別要感謝實驗室的林承龍學長與林聖閔同學。承龍學長不但提供了許多與浮水印相關之參考資料與實驗用程式，對於平日的修課建議與作業討論，也都一直很熱心的給予協助。聖閔則經常與我就彼此的實驗方法與結果進行討論，對於論文演算法的修正與問題的釐清提供很大的助益。



在學科選修的同時，我也認識了許多好同學，如劉能、寶祺、世國、米鳳、佳郁、婉真、惠琪、稚農、銘信、仁峰、鋸賢、錫霖、志誠、國誠、文雅等，不論是平日修課或是假日出遊，這些同學都是我最優秀的伙伴。此外麻姐經常義務便車接送我往返公司與學校，我也一直感激在心。

最後必須感謝我的家人與女友香伶，由於你們的支持與鼓勵，方能使我在兼顧工作及家庭的情況下順利完成學業。每當夜深人靜，我獨自一人挑燈夜戰之時，你們就是我背後最大的動力。

陳白駿

2006年6月30日

謹於交通大學資訊管理研究所

目錄

頁次

摘要.....	i
ABSTRACT.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	viii
表目錄.....	x
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	1
1.3 研究目的.....	2
1.4 論文架構.....	3
第二章 文獻探討.....	5
2.1 資訊隱藏.....	5
2.2 數位影像浮水印.....	6
2.2.1 定義與應用.....	6
2.2.2 數位影像浮水印之分類.....	7
2.2.3 浮水印之特性需求.....	11
2.2.4 空間域與頻率域.....	14
2.3 小波轉換之原理與概念.....	16
2.3.1 濾波器組理論.....	16
2.3.2 小波頻率域特性.....	18

第三章 小波樹量化數位影像浮水印演算法	19
3.1 浮水印嵌入流程.....	19
3.1.1 浮水印產生機制	19
3.1.2 小波轉換程序	20
3.1.3 小波樹的組成	22
3.1.4 Super Tree的組成	23
3.1.5 浮水印的嵌入	24
3.2 浮水印取出流程.....	27
3.3 浮水印的驗證.....	29
3.4 安全性缺失.....	30
第四章 以動態能量差異化為基礎之浮水印演算法	33
4.1 以動態能量差異化為基礎之浮水印嵌入流程.....	33
4.1.1 浮水印產生	34
4.1.2 小波轉換程序與小波樹的組成	34
4.1.3 動態能量差異化之浮水印嵌入方式	35
4.1.4 第四階層小波係數之浮水印嵌入	36
4.1.5 第二、三階層小波係數之浮水印嵌入	37
4.1.5.1 垂直方向的浮水印嵌入.....	41
4.1.5.2 水平方向的浮水印嵌入.....	44
4.1.5.3 斜對角方向的浮水印嵌入.....	46
4.1.5.4 差異化方向的選擇.....	49
4.1.5.5 小波樹差異化方向之記錄.....	50
4.2 以動態能量差異化為基礎之浮水印取出流程.....	52
4.2.1 第四階層小波係數之浮水印取出	53
4.2.2 第二、三階層小波係數之浮水印取出	54

4.2.3 浮水印位元結果之考量分析	55
4.3 浮水印的驗證.....	58
4.4 浮水印演算法之改進.....	58
4.4.1 影像切割重組之前置處理改進方法	59
4.4.1.1 影像切割重組之前置處理演算法.....	59
4.4.1.2 加入影像前置處理後之浮水印嵌入與取出流程.....	61
4.4.2 隨機強制差異化方向改進方法	63
4.4.2.1 隨機強制差異化方向演算法.....	63
4.4.2.2 隨機強制差異化方向之浮水印嵌入與取出流程.....	65
4.5 浮水印嵌入與取出演算法.....	67
4.5.1 以動態能量差異化為基礎之浮水印演算法	67
4.5.1.1 浮水印嵌入演算法.....	67
4.5.1.2 浮水印取出演算法.....	68
4.5.2 影像切割重組之前置處理浮水印演算法	69
4.5.2.1 浮水印嵌入演算法.....	69
4.5.2.2 浮水印取出演算法.....	70
4.5.3 隨機強制差異化方向之浮水印演算法	71
4.5.3.1 浮水印嵌入演算法.....	71
4.5.3.2 浮水印取出演算法.....	72
第五章 實驗結果與討論	73
5.1 JPEG壓縮攻擊	78
5.2 SPIHT壓縮攻擊	84
5.3 Median Filter攻擊.....	88
5.4 Gaussian Filter and Sharpening攻擊	93
5.5 Pixel Shifting攻擊	95

5.6 Rotation and Scaling攻擊.....	101
5.7 Multiple Watermarking攻擊.....	107
5.8 Bitplane Removal攻擊.....	111
5.9 實驗結果整理比較.....	115
5.10 浮水印長度相關效益分析.....	117
第六章 結論與未來展望.....	121
6.1 結論.....	121
6.2 未來展望.....	122
參考文獻.....	123
附錄 Filter Banks.....	126



圖目錄

	頁次
圖 1 國立交通大學logo	8
圖 2 可視性浮水印.....	10
圖 3 不可視性浮水印.....	10
圖 4 LSB演算法範例說明圖	14
圖 5 浮水印嵌入與取出之基本流程圖.....	15
圖 6 一維小波轉換之分解與合成示意圖.....	17
圖 7 小波樹量化數位影像浮水印之嵌入流程圖.....	19
圖 8 小波轉換示意圖.....	21
圖 9 影像經四層小波轉換後之對照圖.....	21
圖 10 子頻帶係數間之四元樹對應關係圖.....	22
圖 11 小波樹結構圖.....	23
圖 12 Super Tree的組成示意圖	24
圖 13 Super Tree轉換為二進位位元平面	25
圖 14 Super Tree量化示意圖	26
圖 15 小波樹量化數位影像浮水印之取出流程圖.....	28
圖 16 以動態能量差異化為基礎之浮水印嵌入流程.....	33
圖 17 小波樹不同階層之係數結構示意圖.....	35
圖 18 第四階層之小波係數.....	36
圖 19 第二、三階層小波係數之分組示意圖.....	38
圖 20 第二、三階層各組之小波係數圖示.....	39
圖 21 以動態能量差異化為基礎之浮水印取出流程.....	52
圖 22 五層小波分解之子頻帶係數對比敏感度權重比例.....	56

圖 23 以混沌排列方式對影像進行切割重組.....	60
圖 24 以Lena影像進行切割重組之前後對照圖.....	60
圖 25 加入影像前置處理後之浮水印嵌入流程.....	61
圖 26 加入影像前置處理後之浮水印取出流程.....	62
圖 27 改進差異化方向演算法後之浮水印嵌入流程.....	65
圖 28 改進差異化方向演算法後之浮水印取出流程.....	66
圖 29 實驗所使用之三張數位影像.....	73
圖 30 Lena原始影像與以JPEG對嵌入浮水印的影像進行壓縮攻擊後之比較	78
圖 31 三張影像內之浮水印對JPEG壓縮攻擊之承受程度	82
圖 32 Lena原始影像與以SPIHT對影像進行壓縮攻擊後影像之比較	84
圖 33 Median Filter 運作原理.....	88
圖 34 Median Filter 運作範例圖.....	89
圖 35 原始Lena影像與Median Filter處理過影像之對照圖.....	89
圖 36 原始Peppers影像與向右位移 9 個像素後之影像對照圖	95
圖 37 原始Peppers影像與向下位移 9 個像素後之影像對照圖.....	96
圖 38 原始Goldhill影像與經過Rotation and Scaling攻擊後之影像對照圖.....	101
圖 39 原始Goldhill影像與經過Multiple Watermarking攻擊後之影像對照圖.....	107
圖 40 原始Lena影像與移除組成小波樹之頻率係數 5 個LSB之影像對照圖	111
圖 41 對Lena影像嵌入不同長度浮水印後之JPEG壓縮攻擊結果	119

表目錄

	頁次
表 1 子頻帶之對比敏感度權重比例數值對應表.....	57
表 2 浮水印抵抗影像處理攻擊之實驗列表.....	75
表 3 實驗結果代號與演算法差異一覽表.....	75
表 4 五組實驗之浮水印演算法運算時間比較表.....	76
表 5 以JPEG對影像進行壓縮攻擊後之實驗結果.....	80
表 6 以SPIHT對影像進行壓縮攻擊後之實驗結果.....	86
表 7 以Median Filter對影像進行攻擊後之實驗結果.....	91
表 8 以Gaussian Filter及Sharpening攻擊後之實驗結果.....	94
表 9 以Pixel Shifting Type-1 對影像進行攻擊後之實驗結果.....	97
表 10 以Pixel Shifting Type-2 對影像進行攻擊後之實驗結果.....	99
表 11 以Rotation and Scaling對影像攻擊後之實驗結果.....	106
表 12 以Multiple Watermarking對影像攻擊後之實驗結果.....	109
表 13 以Bitplane Removal對影像攻擊後之實驗結果.....	113
表 14 實驗結果之整理比較表.....	116
表 15 不同浮水印長度所產生之影像品質比較表.....	117
表 16 浮水印長度、差異化間距、影像品質與浮水印強韌性之相關性列表.....	120

第一章 緒論

1.1 研究背景

隨著資訊科技的快速發展與網際網路的日益普及，各種數位化資訊的儲存與傳遞早已變得既方便又迅速，人們可以輕易的使用各種電子數位產品，將傳統的文字、聲音、影像等平面或類比資訊轉化為數位資訊，不但攜帶方便，且更能達到長久儲存的目的。此外，數位資訊可以輕易的進行編輯與複製，並透過網際網路不斷的傳播至世界各地，當大量的數位資訊在網路上流通的同時，也產生了許多安全性的問題，數位內容的智慧財產權認定即為其中之一。

近年來，由於各種數位內容的非法複製、散佈或提供下載之事件愈來愈多，使得社會上重視智慧財產權的意識亦逐漸抬頭，許多人開始關心數位內容的創作權益，並期待能夠有一個完善的確認方式，以明確的認定數位內容的所有權歸屬問題，進而在有足夠保障的機制下，增進數位內容創作相關產業的發展。正因為如此，將浮水印嵌入數位多媒體創作的數位浮水印技術也就愈來愈受到重視了 [1]。

1.2 研究動機

在現今的多媒體及網路時代，數位內容的應用與傳輸愈來愈普及，尤其在傳統相機紛紛被數位相機所取代，且網路相簿、部落格(BLOG)風氣盛行的趨勢下，數位影像可以非常輕易的透過網路而取得，於日常生活中也變得更為常見，相對的，數位影像的安全性也就愈來愈受到重視。

在解決數位影像的安全性問題上，可概分為兩大防護技術，分述如下：

一、影像加解密技術 [2][3]：

影像加解密技術是將傳統密碼學(Cryptology)的技術運用在數位影像的資料處理上。在原始影像資料傳輸前，透過影像加密的處理，使得非法竊取者即得到加密過的數位影像，亦因為無法得知加密處理時所使用的私密金鑰，故無法將其還原成原始影像；而當合法接收者在取得此加密過的影像資料後，因為本身擁有私密金鑰，故得以透過影像解密處理流程，快速的獲得原始數位影像。運用影像加解密技術，可以防止數位影像在傳輸的過程中，被未經授權之第三者非法取得真正的原始影像，進而確保數位影像的安全性。

二、資訊隱藏技術：

資訊隱藏技術的主要精神是將機密的資訊藏入另一個具有意義但不重要的資料中，使人類的感官神經(視覺、聽覺)無法直接察覺到機密資訊的存在，進而除了得以確保通訊的安全之外，並可進一步的強化該機密資料的保護 [4]。運用在數位內容的安全性問題方面，則可以在數位內容中加入數位浮水印，當對於該數位內容的智慧財產權歸屬有所爭議時，可以將數位內容中的浮水印透過特定機制解析出來，以作為智財權認定上的證明依據。

雖然影像加解密技術可以防止未經授權之第三者非法取得真正的原始數位影像，但仍然無法阻止合法使用者進行未經授權的複製與傳播行為，當某一合法使用者將數位影像任意提供給其他未經授權的使用者使用時，將會造成影像原創者在著作權上的損害。為了能夠提供一種有效的仲裁機制，在數位影像的著作權及使用權有所爭議時，得以保障原創作者或合法使用者的權利，數位影像浮水印技術即成為一項重要的研究課題。

1.3 研究目的

本論文之研究目的，在於透過離散小波轉換的原理，以影像頻率域整體小波係數之

能量差異化特性，取代參考文獻 [1]所提出之係數位元平面量化方法，進而達到提升浮水印嵌入演算法效能之目的。本文針對欲嵌入之特定浮水印位元值，尋求最佳的係數能量差異化方向，並依差異化方向動態的改變小波係數能量，藉由係數間的能量差異特性以嵌入指定的浮水印位元值。

在提升浮水印安全性與抵抗影像處理攻擊之能力方面，本論文利用隨機亂數種子將原始影像重新切割重組，轉化為具有馬賽克(mosaic)效果的新影像，藉以將原始影像與新影像之關聯性在一定程度上作切割，並能夠將被破壞之原始影像局部範圍分散至新影像各處，之後再透過動態能量差異化之運算，將浮水印隱藏於特定之小波樹結構之中，並加以比較浮水印強韌性與安全性的提升程度。此外，在差異化方向的選擇演算法中，本論文更嘗試以隨機強制差異化方向取代最適方向之方法，由隨機亂數種子決定係數能量改變的方向，以捨棄原始浮水印演算法必須額外記錄的差異化方向資訊。

最後經由各種影像訊號處理之攻擊實驗，加以驗證本數位影像浮水印方法之效能，並比較浮水印強韌性與安全性在加入影像切割重組方法與改變差異化方向選擇演算法前後的變化，希望能夠建構出更完善的小波樹數位影像浮水印演算法，進而提供更值得信賴的數位智財權保護技術。

1.4 論文架構

本篇論文的架構共分為六大章節，分述如下：

第一章、緒論：

說明數位影像在智慧財產權保障方面的問題，以及數位浮水印所能夠提供的解決方法，並對於整篇論文的研研究方向與目的作概略性的陳述。

第二章、文獻探討：

簡述資訊隱藏技術的意義與精神，以及與浮水印技術之間的關係，並對於數位浮水印的定義、種類、特性、實作上的演算法類型以及小波轉換的原理作深入的探討，並針對小波分解後之小波係數特性作進一步的介紹。

第三章、小波樹量化數位影像浮水印演算法：

介紹參考文獻 [1]所提出之小波樹量化(WTQ, Wavelet Tree Quantization)數位影像浮水印演算法，並說明採用之浮水印產生機制及浮水印取出後的驗證流程。最後則是說明小波樹量化浮水印演算法在安全性方面所產生的問題，並敘述破壞該演算法的步驟。

第四章、以動態能量差異化為基礎之浮水印演算法：

提出以動態能量差異化為基礎之數位影像浮水印演算法，詳細說明浮水印嵌入與取出之程序步驟，以及為防範惡意攻擊者可能進行的係數結構分析破壞所提出的反制方法，期使浮水印之強韌性與安全性具有一定程度之保障。此外並對於本論文所提出之演算法嘗試作進一步的改良，在浮水印嵌入與取出程序中新增影像切割重組之前置處理步驟，以及使用隨機強制差異化方向演算法取代原先需額外記錄方向值的最適方向演算法，希望能夠得到更具安全性與強韌性之數位浮水印方法。

第五章、實驗結果與討論：

透過各種影像訊號處理的攻擊方式，對於實驗影像進行破壞，以測試本論文所提出之數位影像浮水印演算法的安全性與強韌性。我們分別針對選用不同差異化方向決定機制以及加入影像切割重組方法前、後之演算法進行實驗，並以 9-7、18-10 兩種filter參數進行對照，所得數據將與 [1]之實驗結果相比較並加以討論。

第六章、結論與未來展望：

針對實驗結果作出結論，以說明本論文之貢獻，並提出未來可能之研究方向。

第二章 文獻探討

2.1 資訊隱藏

資訊隱藏(Information Hiding)可視為一種迥異於傳統密碼技術的加解密機制，其主要精神是將機密的資訊藏入另一個有意義但卻不重要的資料中，使人類的感官神經(如視覺、聽覺等)無法直接察覺到機密資訊的存在，進而除了得以確保通訊的安全之外，並可進一步強化該機密資訊的保護 [4]。

由於傳統密碼技術雖然可藉由加密以提供原始資料的保護，然而惡意攻擊者仍然能夠得知該筆資料的存在，進而對加密後的資料進行破解攻擊。資訊隱藏技術則是運用偽裝的方式，將重要資訊隱藏在另一個掩護性資料當中，使惡意攻擊者被蒙蔽而無法發現其重要性，進而順利的將資訊傳送至目的地。這就如同自然界中，有些生物如烏龜、貝類等，牠們擁有堅硬的外殼，當遭遇掠食者攻擊時，僅能以外殼與之對抗，一但殼被敲碎，則性命危矣，但偏偏牠們的天敵通常都擁有破解堅硬外殼的能力；然而另外有一些生物如竹節蟲、比目魚等，牠們捨棄厚重堅硬的外表，而採取變色、擬態等偽裝技巧，當威脅狀況發生時，以儘量不引起敵人注意的方式進行藏匿，若是隱藏技巧夠好，通常可順利騙過敵人的視覺搜索而活命。

早期資訊隱藏是以「Steganography」一字來表示，其源自希臘文Steganos與graph，Steganos表示隱藏之意，graph則為文字寫作之意，故Steganography之主要意思即為將寫作文字之存在完全隱藏起來(covered writing)，也就是資訊隱藏的意思 [5]。近年來，資訊隱藏技術在不斷的演進下，其應用範圍則愈來愈廣泛，舉凡文字、聲音、影像、動畫等數位媒體內容，皆可運用各種技巧與方法將資訊進行隱藏，隱藏之目的小至個人隱私的保護，大至軍事計畫或商業機密在傳遞上的安全性等等，故相關之研究探討也相當眾多，數位浮水印技術即為其中之一。

然而，資訊隱藏與數位浮水印在用途上仍稍有不同，資訊隱藏的精神在於將機密資訊藏入掩護資料(Covered Data)當中，以便使機密資訊在傳遞過程中安全無虞，故重點在於機密資訊的防護，至於掩護資料的重要性則較無關緊要；但在數位浮水印技術之中，浮水印是為了提供被藏入之資料價值的驗證，即該份資料本身就具有極高的重要性，而浮水印可以提供關於這份資料的某些證明。由此可知，雖然數位浮水印被歸類於資訊隱藏之領域，但浮水印技術與一般的資訊隱藏技術仍有其相異之處。

2.2 數位影像浮水印

2.2.1 定義與應用

數位影像浮水印是一種可提供數位影像智慧財產權保障的技術，主要是利用資訊隱藏的原理，將具有代表性的影像合法版權資訊（如產品序號、註冊商標、營利事業統一編號、作者個人資料等）藏入數位影像中，當發生版權爭議時，可透過一特定且公開的程序將影像中的浮水印解析出來，以作為該影像版權歸屬的認證。



數位浮水印技術的應用範圍相當廣泛，依據不同的使用需求，可概分為下列四點 [6]:

一、版權的保護(Copyright Protection)：

版權的保護是目前數位浮水印技術最常見的應用。現今網路上的許多影像，常會被人們任意的下載散佈，致使影像創作者的權益遭到侵害，而數位浮水印本身具有強韌與明確的特性，不但可以抵抗各種影像處理的攻擊，更可明確的辨識出該浮水印所代表的影像版權資訊，故透過將版權資訊以浮水印型式嵌入影像之中的技術，可以在版權有所爭議時將浮水印取出，以確認影像合法擁有者的身份，進而達到保護版權的目的。

二、對於非法使用者的追蹤(Traitor Tracking)：

一般的數位浮水印機制可提供影像版權的確認，以得知使用者是否具有合法的授

權，然而一但發生侵害智財權的狀況，這種方法僅能查獲影像的非法使用者，卻無法得知最初非法散佈影像者的身份。為了能夠追蹤影像遭非法複製、傳播的來源，可以於授權使用之前，將可代表獲授權者身份的資料嵌入影像之中，使每張影像都藏有可確認其合法使用者身份的獨特資訊，此種浮水印技術的應用又稱為數位指紋(Fingerprinting)。日後若發生非法散佈影像的情形，則可將數位指紋由該影像中取出，以得知影像非法散佈的來源為何。

三、複製的防護(Copy Protection)：

在開放系統中，多媒體資訊常會遭人們非法複製與使用，要作到檔案複製的限制是較為困難的；然而於封閉系統中，則可以利用浮水印技術加以禁止未獲授權的複製行為發生，例如DVD影片撥放系統中，DVD撥放器會偵測其所播放的影片資訊中是否含有”禁止複製”或是”僅容許複製一次”的浮水印資訊，並依照浮水印內容決定是否可以允許複製檔案，進而達到防護多媒體檔案複製行為的目的 [7]。



四、影像完整性的確認(Image Authentication)：

對於某些不容任意竄改的重要影像，如軍事或醫學用途影像，可以使用強韌性低的易碎型浮水印技術(Fragile Watermarking Technique)以確保影像的完整性與正確性。當嵌入易碎型浮水印的影像遭到竄改時，浮水印機制可以偵測出此一事實，並得知影像遭竄改的位置或區域，以保障使用者不致因影像資訊錯誤而在應用上有所誤判。

2.2.2 數位影像浮水印之分類

數位影像浮水印會因使用者不同的功能需求或分類標準，而具有不同的種類，若是以浮水印本身的存在型式而言，可分為下列三種：

一、圖像(logo)表示：

此類浮水印本身為一個可代表版權所有者之圖像標誌，在一般可視性浮水印中頗為常見，如圖 1。



圖 1 國立交通大學 logo

二、文字訊息(Text message)表示：

即浮水印本身為一段有意義的文字，如作者、公司名稱、地址、授權日期等訊息，此類浮水印可明確說明版權的內容。

三、數字序列(Number sequence)表示：

此種浮水印本身為一連串的數字，通常可由特定的統計分配機制所產生，在驗證版權歸屬時，需要將解析出來的數字序列浮水印與原始的數字序列浮水印作統計上的比對，方能得到版權的確認。

若依浮水印對於影像訊號處理的抵抗程度及影像竄改的偵測功能取向作分類，可將浮水印技術區分為下列三種 [8]：

一、強健型浮水印(Robust Watermarks)：

強健性浮水印技術著重於對影像處理攻擊的抵抗能力，嵌入浮水印的影像在經過各種影像處理攻擊之後，隱藏其中的浮水印資訊依然可以偵測並解析出來。由於強健型浮水印能夠有效抵抗影像訊號處理，故主要是用於提供影像版權的保障機制，一旦版權歸屬產生爭議時，即可將浮水印取出以作為有效的證明依據。

二、易碎型浮水印(Fragile Watermarks)：

易碎性浮水印技術則是著重於對影像遭竄改的偵測能力，並且能夠偵測出遭竄改的位置或區域，甚至可修復被竄改的內容。由於易碎性浮水印對抗影像處理的能力非常薄弱，故其主要用途並非用來嵌入版權資訊，而是作為確認數位影像的真確性與完整性，通常用於不容許影像遭竄改的醫學影像或貴重影像中 [9]。

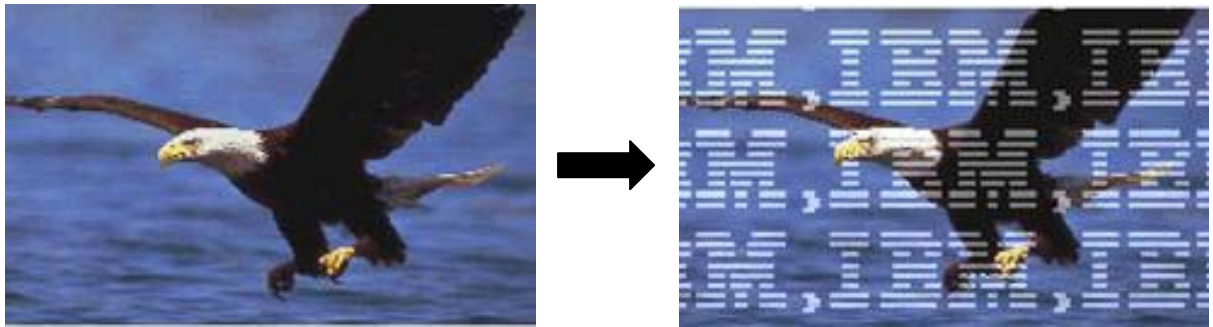
三、半易碎型浮水印(Semi-fragile Watermarks)：

半易碎型浮水印技術可以精確的指出影像遭竄改的區域，而且對於一般合理的影像壓縮處理有良好的抵抗能力。由於一般數位影像通常會利用失真性影像壓縮技術以減少儲存空間，影像在壓縮的過程中，雖然每個像素值都經過輕微的修改，但整體所表示的資訊意義卻與原始影像是一致的，故此種失真性影像壓縮處理應視為合法的操作，而非惡對於影像的惡意竄改。半易碎型浮水印能夠分辨出合法的影像壓縮操作與非法的影像內容竄改之差異，在經過失真性影像壓縮處理後，半易碎型浮水印仍然能夠偵測出來 [9]。

若是以浮水印加入後是否表現於影像的外觀而言，可將數位浮水印分為下列兩種類型：

一、可視性(Visible)浮水印 [10][11]：

當浮水印嵌入影像後，浮水印可為人類肉眼所察覺，如圖 2 [12]。可視性浮水印最主要之功能在於著作權的聲明，以明顯標記的方式防止影像遭非法使用。其優點為不必經過特定的運算或驗證程序，即可立即確識出該浮水印所表示的版權資訊；缺點則是嵌入的浮水印會破壞影像原本的外觀，且加入的浮水印容易為不法人士使用影像處理技術去除，甚至以另一個浮水印所取代，造成版權驗證上的困擾。



原始影像

加入可視性浮水印後之影像

圖 2 可視性浮水印

二、不可視性(Invisible)浮水印：

當浮水印嵌入影像後，人類無法用肉眼察覺浮水印的存在，如圖 3 [13]。其優點為影像原本的美觀不會遭到破壞，且不法人士無從得知嵌入的浮水印為何，故提高了浮水印遭破壞的門檻；缺點則是必須要透過特定的運算或驗證程序，才能將浮水印解析出來以證明版權的歸屬。由於不可視性浮水印技術是將浮水印隱藏在特定影像之中，故我們將之歸類為資訊隱藏(Information hiding)技術。



原始影像

浮水印

嵌入浮水印之影像

圖 3 不可視性浮水印

此外，從設計藏入浮水印演算法的觀點而言，不可視性浮水印技術又可區分為兩大類，即空間域(Spatial domain)浮水印與頻率域(Frequency domain)浮水印，將於 2.2.4 節作定義說明與進一步的探討。


2.2.3 浮水印之特性需求

數位影像浮水印技術為了達到保障數位影像智財權的目的，必須要能夠抵抗各種影像訊號處理的攻擊，避免浮水印遭到不法人士的破壞，以維持浮水印的安全性。有鑑於此，浮水印技術在發展的過程中，必須考量以下幾種特性需求，分述如下 [14][15]：

一、透明性(Transparency)：

此特性是針對不可視性浮水印技術而言。加入浮水印後的影像，必須使人類無法用肉眼辨識出與原始影像的差異，意即影像的品質不致受到浮水印嵌入的影響，如此則可保持影像的美觀，維持其商業上的價值。

在影像品質的客觀評定方法中，學術研究上一般是使用SNR(Signal-to-Noise Ratio)或PSNR(Peak-Signal-to-Noise Ratio)來對嵌入浮水印前後的兩張影像作品質比較，其中又以PSNR較常被使用 [4]。SNR與PSNR之定義如下：


$$SNR = \frac{\sum_{i=1}^M \sum_{j=1}^N \hat{X}_{ij}}{\sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \hat{X}_{ij})^2} \quad (1)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2)$$

在(2)式中，MSE(Mean-Square Error)之定義如下：

$$MSE = \left(\frac{1}{M \times N} \right) \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \hat{X}_{ij})^2 \quad (3)$$

在(1)式至(3)式中，M、N參數分別為影像的寬度及高度， X_{ij} 與 \hat{X}_{ij} 則分別表示在座標點 (i,j) 的原始影像像素值與嵌入浮水印後的影像像素值。原則上，PSNR值愈高則表示影像品質愈佳，當PSNR值高於 30dB時，人眼即很難分辨出兩張影像間的差異 [4]。然而PSNR值與影像品質的關係並非絕對，若某張影像的少數像素值由黑變白，因為被改變的像素很少，使計算出的PSNR值仍然偏高，但影像的改變之處卻可為人眼輕易察覺出來，故有些研究會使用人類視覺系統(HVS, Human Visual System)，作為嵌入浮水印之影像對人類視覺影響程度判斷的依據 [16][17]。

二、安全性(Security)：

數位影像浮水印技術必須要能夠防止非法人士的偵測，僅合法使用者方能偵測出影像中的浮水印，此外即使知道影像中已加入浮水印，也無法在未經授權的情況下，將合法使用者所加入的浮水印移除。由於數位浮水印嵌入與析出的演算法則必須是公開的，故我們無法將安全性維繫在使用者不知道系統如何運作的假設之上，而必須在演算法中建構一套完善的防護機制，例如增加秘密金鑰(Secret Key)的防護機制 [4][8][14]，以確保浮水印的安全性。

三、明確性(Unambiguous)：

自數位影像中解析出來的浮水印，必須能夠明確的辨識版權擁有者等資訊，且在不影響到影像外觀商業價值的前提下，即使遭到各種影像處理的攻擊，都應該要能夠自影像中解析出具明確資訊的浮水印，如此方能保障影像創作者的智慧財產權。

四、強韌性(Robustness)：

浮水印技術必須具備一定的強韌性，使影像在承受各種數位訊號處理技術攻擊後，在品質可接受的範圍內，浮水印仍然存在於影像之中，且能夠從影像中將浮水印正確的解析出來，以證明版權的歸屬。

五、多重性(Multiple Watermarking)：

一張影像必須能夠允許多個浮水印同時存在，如此才能達到在商業用途上的買賣轉售行為中，每一位銷售者與使用者對該影像所有權的追蹤與保障。對於浮水印技術的多重性而言，每一個合法擁有該影像的使用者都可以在影像中加入專屬的浮水印，且後面加入的浮水印無法對已存在的浮水印造成損害。

六、容量大小(Capacity)：

容量大小係指在一張影像中所能嵌入的浮水印長度，當能夠加入的浮水印長度愈長，則在浮水印驗證階段的精確度就愈能提高，故原則上倘若一個浮水印演算法能夠容許影像嵌入愈多的浮水印資訊，則該演算法被認為是較佳的。此外若浮水印演算法的容量較大，則使用者在決定欲嵌入的浮水印長度時亦較為彈性，否則可能會因為演算法所提供的浮水印容量上限不足而被迫刪減欲嵌入的浮水印長度。

七、不需與原始影像比對(Blindness)[18]：

在浮水印取出階段，若是必須使用原始影像以進行比對，則最初浮水印嵌入者即需要將已嵌入浮水印的影像與原始影像同時傳送至合法使用者處，此種方法既浪費網路資源又佔用儲存空間，且因為原始影像在傳送過程中勢必降低其安全性，故近期發展之浮水印技術，皆是以不需與原始影像比對即可解析出浮水印的演算法為主。

在以上數種浮水印的特性需求中，某些項目之間其實存在著對立與矛盾，無法同時滿足其最佳化，例如若某浮水印演算法所能提供的容量愈大，表示可嵌入的浮水印資訊愈多，在驗證浮水印時亦可提升其精確度，然則也因為浮水印長度較長，在嵌入時勢必造成影像的改變量增加、影像品質降低，進而導致透明性變差，此外浮水印的強韌性通常與透明性相衝突，在浮水印長度不變的情況下，若欲強行提高透明性，則可能會導致浮水印強韌性不足。故當我們在設計浮水印演算法時，需要同時考量以上數種浮水印之特性，在平衡中尋求一個最佳法則，以避免顧此失彼。

2.2.4 空間域與頻率域

不可視性浮水印技術由於具有資訊隱藏的特性，且對各類型訊號處理的攻擊一般具有較佳的抵抗能力，故此一領域的研究也受到較多的重視。不可視性浮水印技術依嵌入方式之不同，可區分為兩大類，分述如下：

一、空間域浮水印(Spatial domain watermark)[19][20][21][22]：

空間域浮水印技術是直接藉由改變影像像素(Pixel)的數位資料值，以達成浮水印嵌入的目的。此種方法的嵌入技術較簡單，且具有運算速度快的優點，但抵抗各種影像訊號處理攻擊的能力較弱，目前多半用於易碎型浮水印的製作。

空間域浮水印技術中，LSB(Least Significant Bit)[21][22]是較常被使用的方法之一，其基本的演算法則為將欲嵌入的浮水印資訊放入影像特定像素值的較低位元，因此能在有限度改變像素值大小的前提下，達到對影像品質破壞程度較低且順利藏入浮水印的目的。如圖 4 所示，某原始像素值為 164，若在 LSB 位置嵌入 2 位元浮水印「11」，則嵌入後像素值被修改為 167，由於像素值的變化很小，故影像品質僅受到極微小的影響，但不法人士亦可輕易的將雜亂數值放入 LSB 位置以破壞浮水印。

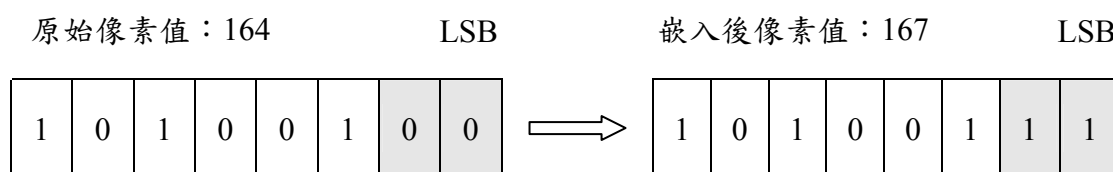


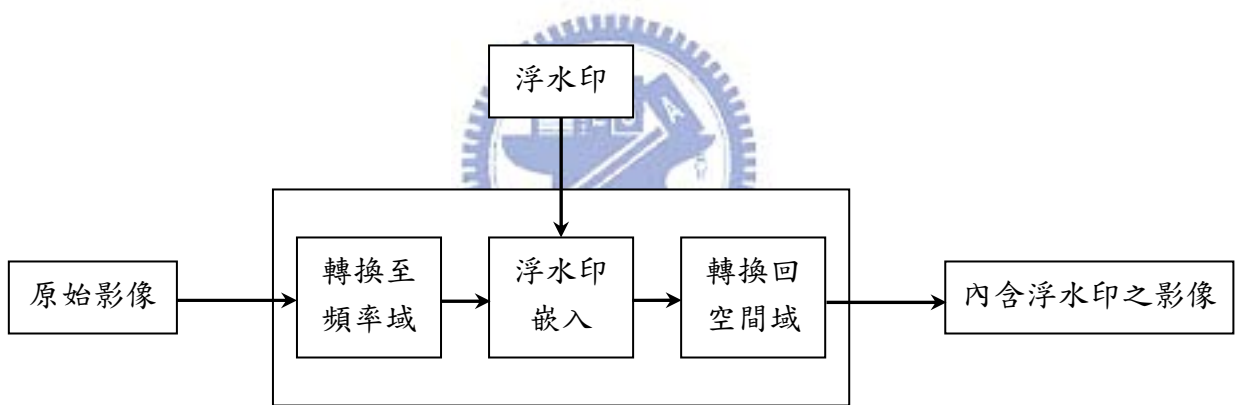
圖 4 LSB 演算法範例說明圖

由於將浮水印嵌入像素的 LSB 位置容易遭到破壞，嵌入 MSB(Most Significant Bit) 位置則雖然具有較強的抵抗影像處理之攻擊能力，但相對的會造成影像品質的嚴重破壞，

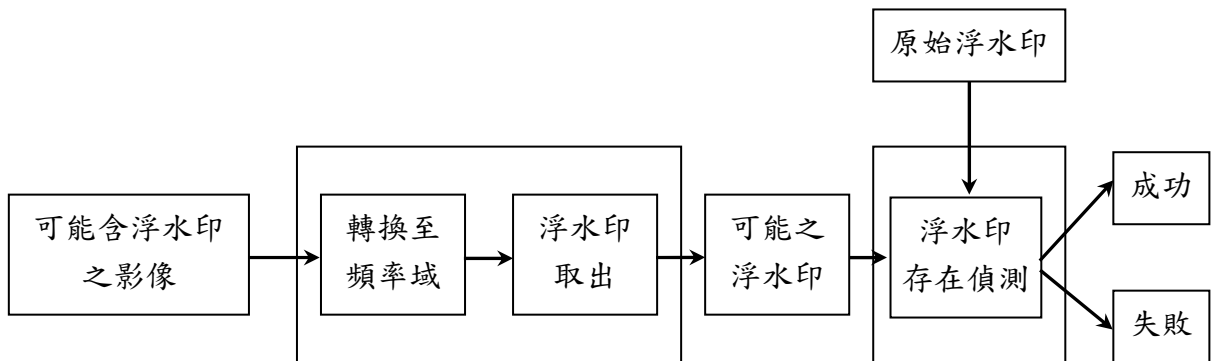
故有學者提出失真補償技術(Distortion Reduction)[20]，調整較高位元被嵌入浮水印的像素值，以提高嵌入浮水印後的影像品質。

二、頻率域浮水印(Frequency domain watermark)[1][23]：

頻率域浮水印技術是先將原始影像由空間域轉換至頻率域，再將浮水印嵌入轉換後所得到的頻率域係數值，之後再將改變後的頻率域係數逆轉換回原先的空間域，以完成浮水印的嵌入動作。當需要將浮水印解析出來以驗證影像的版權時，同樣必須先將影像由空間域轉換至頻率域，再取出隱藏於頻率域係數中的浮水印，並將此浮水印與原本的浮水印作存在偵測的相關性比較，以確認取出的浮水印確實是當初嵌入的浮水印，其嵌入與取出的基本流程如圖 5。



(a)浮水印嵌入基本流程



(b)浮水印取出基本流程

圖 5 浮水印嵌入與取出之基本流程圖

頻率域浮水印技術常用的轉換方式有三種，分別為離散餘弦轉換(DCT, Discrete Cosine Transform)、離散傅立葉轉換(DFT, Discrete Fourier Transform)與離散小波轉換(DWT, Discrete Wavelet Transform)。在影像處理方面，傳統的影像壓縮大部份是以離散餘弦轉換(DCT)為基礎，例如JPE壓縮法，就是以離散餘弦轉換來進行圖形資料的壓縮。至於離散小波轉換則為JPEG 2000所使用到的技術，同時也是本篇論文所使用的頻率域轉換方法，我們將於2.3節介紹其原理。

2.3 小波轉換之原理與概念

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。

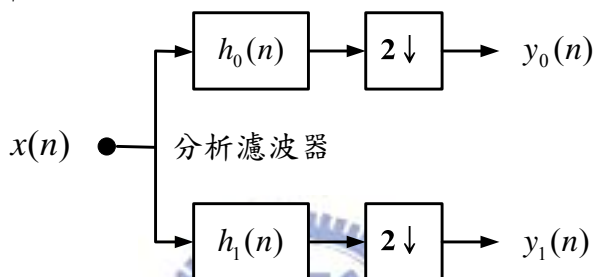
小波轉換方法的提出可追溯到1910年Haar提出小波概念，將訊號處理的方法往前跨進一大步[24]。其後1984年，法國地球物理學J. Morlet在分析地震波的局部性質時，發現傳統的傅利葉轉換，難以達到其要求，因此引進小波概念於信號分析中，對信號進行分解。1988年Daubechies建構了具有正交性(Orthonormal)及緊支集(Compactly Supported)這兩種特性，並且只在一個有限的區域是非零的小波，如此小波轉換的系統理論得到了初步建立[25]。直到1992年，Daubechies等人又提出以小波轉換為基礎的影像壓縮技術，自此為小波影像壓縮建立了新的里程碑[26]。

2.3.1 濾波器組理論

小波轉換是以將輸入訊號通過濾波器組(Filter Band)的方式而達到訊號的分解與合成。濾波器乃是訊號處理系統代稱，濾波器的功能就是選擇性地處理訊號。經由濾波器

可以將訊號做高頻、低頻的分解，此種將訊號分解的過程一般稱之為分析(Analysis)。與分析過程相對的，則是藉由濾波器的作用，將所輸入訊號的每個頻率成分合成而得到一個輸出訊號，此種程序稱為合成(Synthesis)。假設一個訊號經過分析濾波器的分解，且分解後的結果再經過合成濾波器重組而產生成為一個新的訊號，若此新的訊號與原始訊號相同，則我們稱這樣的分析濾波器與合成濾波器能達到訊號的完美重建(Perfect Reconstruction)，且我們稱這樣的分析濾波器與合成濾波器為一組濾波器組。

(a). 訊號分解



(b). 訊號合成

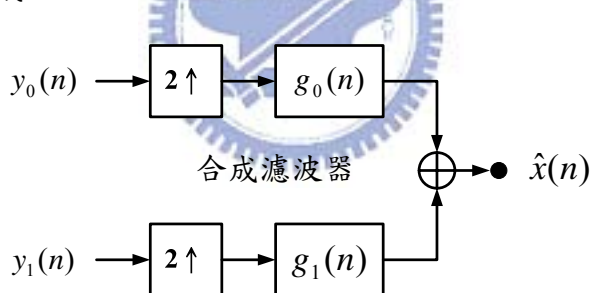


圖 6 一維小波轉換之分解與合成示意圖 (a)訊號分解 (b)訊號合成

上圖 6(a)中，原始訊號 $x(n)$ 經由 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)，能使高頻訊號通過。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 之訊號必須先降低取樣(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 6(b)中，當我們利用合成濾波器 $g_0(n)$ 、 $g_1(n)$ 對低頻與高頻訊號進行合成前，必須先行提高取樣(Up-sample)，才能使訊號的合成順利進行。能夠達成完美重建的濾波器組，則必須滿足下面(4)式：

$$\begin{cases} H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0 \\ H_0(z)G_0(z) + H_1(z)G_1(z) = 2 \end{cases} \quad (4)$$

2.3.2 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：

一、低頻區：

低頻區內之資訊，即表示在空間域影像的像素間數值差異不大、較平滑的部份。一般而言，人類肉眼的視覺感觀對低頻區的敏感度較高，假使我們在低頻區加入些許外來資訊，則當影像轉換回空間域後，其外觀便會被明顯的改變，故由此可知低頻區為影像最重要的區域，該區之係數實際上代表影像整體的特性與資訊。

二、高頻區：

高頻區則是在空間域影像中像素與像素之間變化較大的部份，例如影像中黑白相間的部份，人類肉眼的視覺感觀對高頻區的敏感度較低。假使高頻區的資訊遭到更動，當影像轉換回空間域後，並不會造成影像本身的強烈破壞，故一般的失真性影像壓縮，皆會以減少影像的高頻區資訊作為降低資料量的方法。

三、中頻區：

中頻區是介於低頻與高頻之間的區域，若是在此區域加入額外的資訊，並不像低頻區般易對影像本身特性產生影響，同時亦不會如高頻區般資訊易被移除，故一般的數位影像浮水印技術均選擇中頻區域以嵌入浮水印。

第三章 小波樹量化數位影像浮水印演算法

本章將介紹參考文獻 [1]所提出的小波樹量化(WTQ, Wavelet Tree Quantization)數位影像浮水印演算法，並說明此方法在安全性與強韌性方面所產生的問題。

3.1 浮水印嵌入流程

小波樹量化數位影像浮水印之基本作法，首先須使影像進行四層小波轉換，並將轉換後得到的小波係數組合為小波樹，再令兩棵小波樹組成一棵 Super Tree。之後以兩棵 Super Tree 為單位，依欲嵌入之浮水印位元值之不同，對其中一棵 Super Tree 之位元平面進行量化處理，另一棵則維持不變，我們即可藉由這兩棵 Super Tree 在量化運算後能量之差異，以決定嵌入之浮水印位元值為何，達到浮水印隱藏的目的。其浮水印嵌入流程如圖 7 所示：



圖 7 小波樹量化數位影像浮水印之嵌入流程圖

3.1.1 浮水印產生機制

本篇論文所使用之浮水印為數字序列式浮水印，我們藉由亂數種子的選擇，以隨機方式產生一連串值為+1或-1的PN序列(Pseudo Noise Sequence)整數，用以模擬一個有意義的符號或一段文字訊息，經過單向雜湊函數(one-way hash function)而轉換成數字序列的過程。浮水印產生機制所使用的演算法如下：

浮水印產生之演算法

輸入參數： seed 亂數種子
 watermark_length 浮水印長度

輸出參數： watermark[] PN 序列浮水印

```
1:        srand (seed)
2:        for ( i=0 ; i< watermark_length ; i++ ) {
3:            if ( rand() % 2 == 1)  watermark[i] = 1
4:            else  watermark[i] = -1
5:        }
```

演算法 1

在演算法 1 中，我們指定亂數種子並輸入亂數產生函式中，以產生個數相當於浮水印長度之隨機亂數，所得亂數若是奇數，則浮水印數值設為+1；若亂數為偶數，則將浮水印數值設為-1，如此則可得到所需之隨機 PN 數字序列浮水印。

3.1.2 小波轉換程序

在將浮水印嵌入影像之前，我們必須先將影像經過多樣解析度(multiresolution)的二維小波轉換，使影像自空間域轉換至頻率域，以分離影像中之低頻、中頻與高頻資訊，而後我們才能從中選擇中頻區域的子頻帶(subband)係數以嵌入浮水印，使浮水印既能具有一定之強韌性，且又能使影像品質達到不影響人類視覺觀感的水準。

首先我們必須將欲嵌入浮水印之影像經過四層小波轉換，如圖 8 所示，原始影像經第一層小波轉換後，可分解為 LL₁、HL₁、LH₁、HH₁ 等四個子頻帶，而後再將低頻係數之 LL₁ 子頻帶作第二層小波轉換，又可分解為 LL₂、HL₂、LH₂、HH₂ 等四個子頻帶，如

此經過四層小波轉換後，可將原始影像轉換成為 13 個子頻帶，分別為第四層的 LL₄、HL₄、LH₄、HH₄，第三層的 HL₃、LH₃、HH₃，第二層的 HL₂、LH₂、HH₂ 及第一層的 HL₁、LH₁、HH₁。

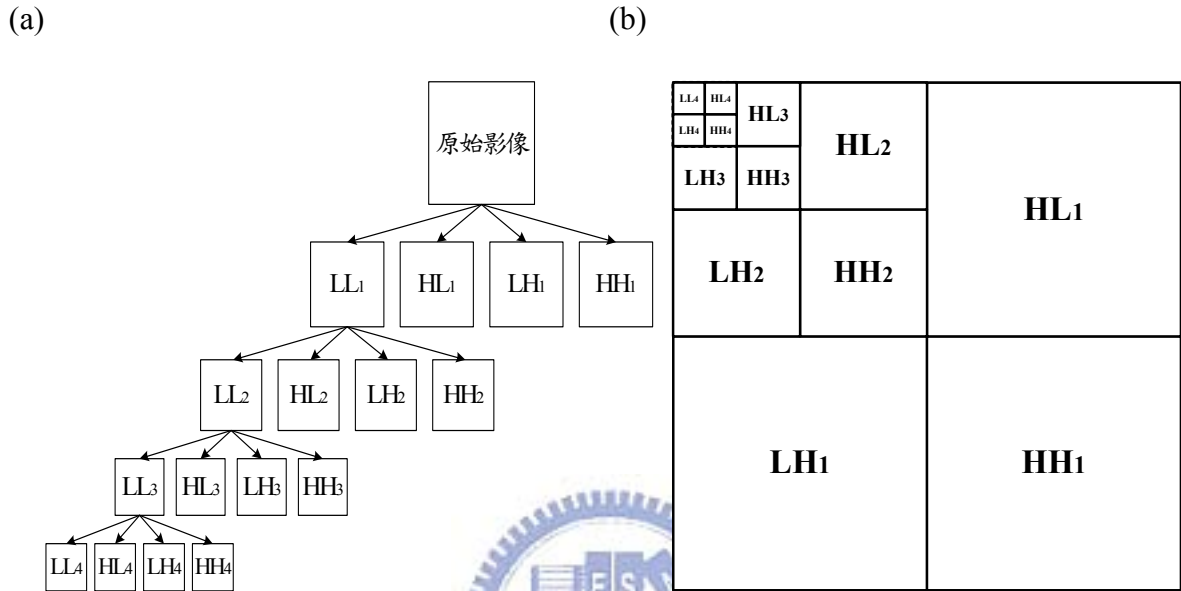


圖 8 小波轉換示意圖 (a)每一次小波轉換均可將原始頻帶(band)分解為 LL、HL、LH、HH 等四個子頻帶。(b)經過四層小波轉換後，可得到 13 個子頻帶，每一個子頻帶實際上均有其對應於原始影像之相對位置。

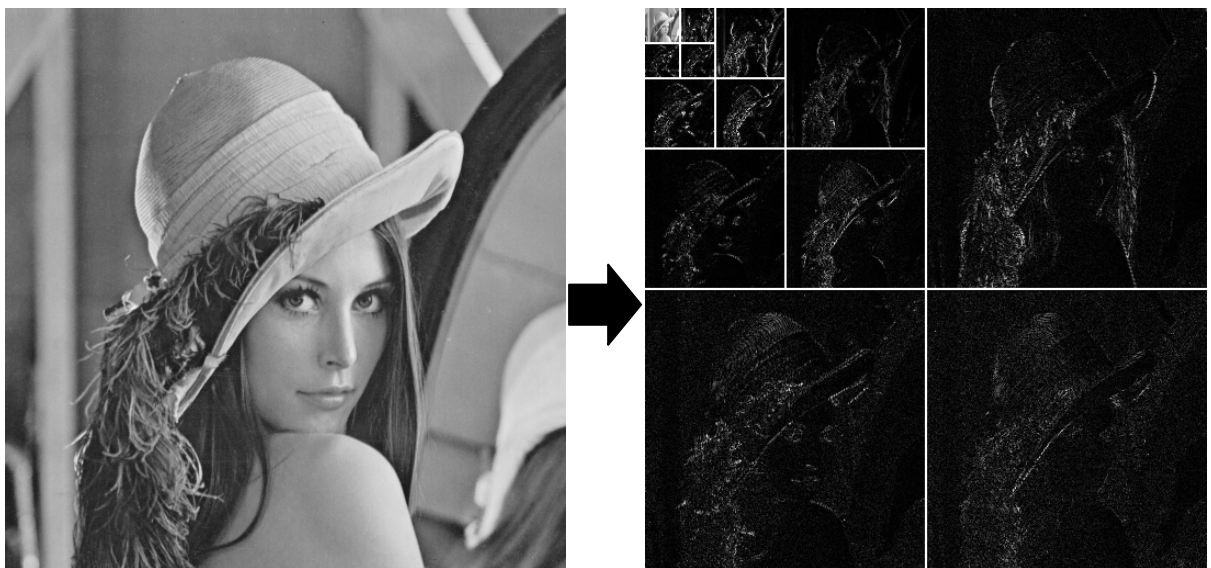


圖 9 影像經四層小波轉換後之對照圖 (a)原始影像 (b) 四層小波轉換後之影像

在圖 9 中，我們可以看出 Lena 影像在經過四層小波轉換後，被分解為 13 個子頻帶，且影像之重要資訊均集中於左上角低頻區的 LL4 子頻帶，我們在嵌入浮水印時，須避免改變 LL4 子頻帶之係數，否則將會嚴重降低影像的品質。

3.1.3 小波樹的組成

影像在經過四層小波轉換後，分解成為不同階層與不同特性之子頻帶，其中具有相對應空間位置但屬不同階層的子頻帶係數之間，可形成四元樹(quadtree)[27]的特殊結構關係，如圖 10 所示：

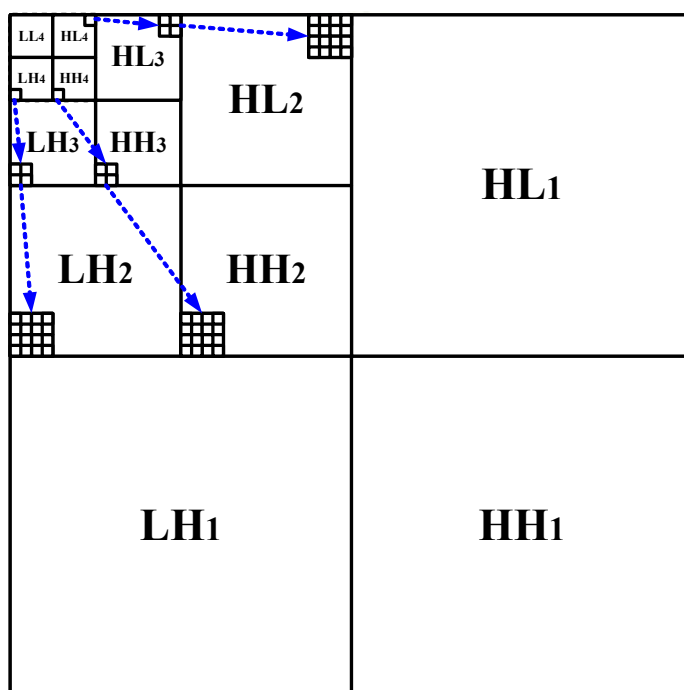


圖 10 子頻帶係數間之四元樹對應關係圖

在圖 10 中，第四階層的 HL4、LH4、HH4 子頻帶中任何一個係數，均擁有其相對應空間位置之第三階層的 4 個子係數，而第三階層的每一個係數亦擁有其相對應空間位置之第二階層的 4 個子係數，此上一階層係數具有相對應下一階層之 4 個子係數之關係，

即稱之為四元樹關係。我們將這具有空間對應關係的 21 個小波頻率域中頻係數組合起來，即成為小波樹結構(Wavelet Tree Structure)，如圖 11。若一張影像經過四層小波轉換後，其擁有之小波樹個數即為第四階層 HL₄、LH₄、HH₄ 子頻帶的總係數個數，由於原始影像的大小為 512×512，第四階層的子頻帶大小則為 32×32，因此可以計算出小波樹總個數為 $32^2 \times 3 = 3072$ 棵。

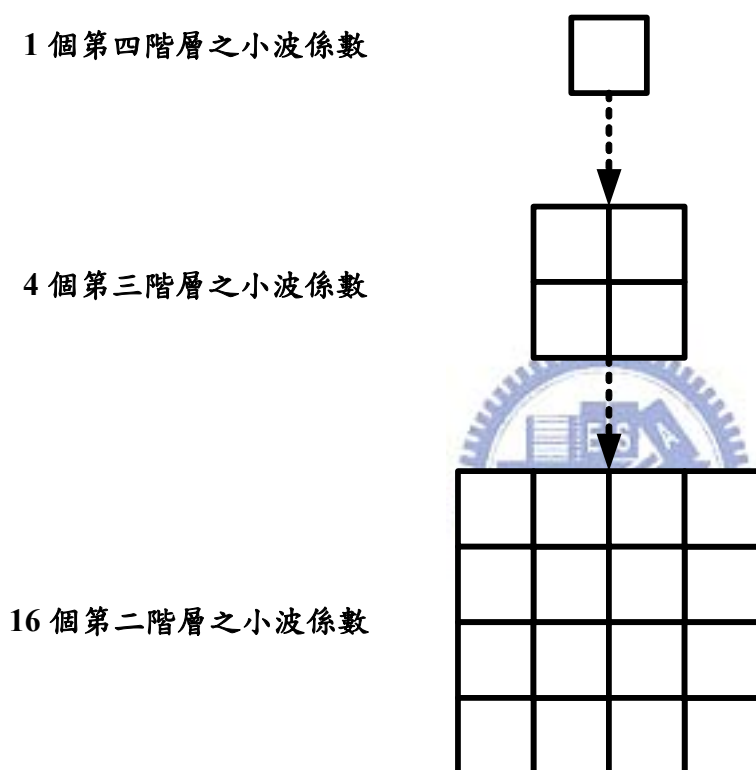


圖 11 小波樹結構圖

3.1.4 Super Tree 的組成

在小波樹組成之後，我們同樣藉由與浮水印產生機制相同的亂數種子，以隨機且不重覆的方式由 3072 棵小波樹中任取兩棵小波樹，並將這兩棵小波樹組成一棵 Super Tree，如圖 12 所示。如此則可組成 1536 棵 Super Tree。

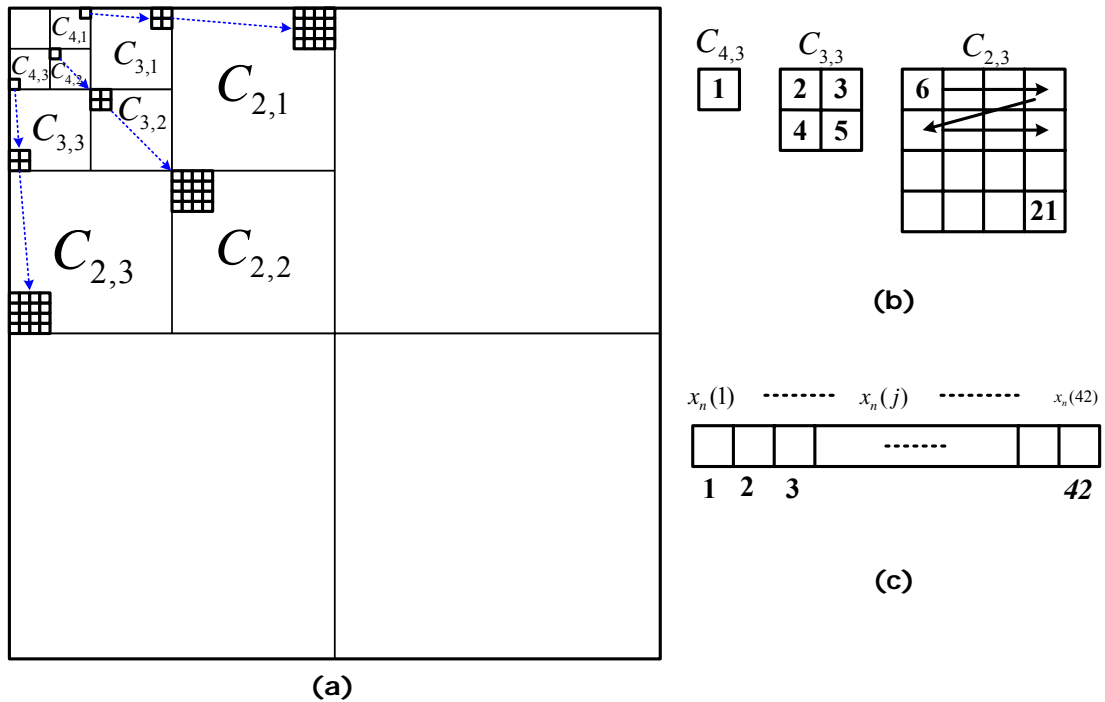


圖 12 Super Tree 的組成示意圖 (a)四層小波分解 (b)小波樹的組成順序，由 21 個小波係數組成一棵小波樹 (c)2 棵小波樹組成一棵 Super Tree，故包含 42 個係數值

隨後我們再由 1536 棵 Super Tree 中，取每兩棵 Super Tree ST_{2i-1} 與 ST_{2i} 為一個單位，成為 Super Tree Pair P_i ，如此則可分為 768 組 Super Tree Pair，如(5)式。在 Super Tree 係數量化以嵌入浮水印的階段中，每一組 Super Tree Pair P_i 都將決定 1 個浮水印位元 W_i 的嵌入，故此方法最多可嵌入之浮水印長度即為 768 個位元值。Super Tree Pair P_i 之表示式如下：

$$P_i = \{ ST_{2i-1}, ST_{2i} \} \quad \text{for } 1 \leq i \leq 768 \quad (5)$$

3.1.5 浮水印的嵌入

當我們欲將浮水印位元 W_i 嵌入 Super Tree Pair P_i 時，必須由 P_i 內挑選其中一棵 Super Tree，並對其位元平面進行量化，而另一棵 Super Tree 則保持不變，如此在量化運算之後， P_i 中的兩棵 Super Tree 之係數能量必有所差異，我們即可根據此差異而決定該

Super Tree Pair P_i 所隱藏之浮水印位元值 W_i 為何。

在選擇 Super Tree Pair P_i 中之 Super Tree 以進行位元平面量化時，我們採用的方法為當欲嵌入之浮水印位元 $W_i = -1$ 時，即對於 Super Tree ST_{2i-1} 作量化；若是 $W_i = 1$ 時，則對 ST_{2i} 進行量化，若以 $Q[\]$ 表量化運算式，則其關係式如下：

$$\begin{cases} Q[P_i] = Q[ST_{2i-1}] & , \text{if } W_i = -1 \\ Q[P_i] = Q[ST_{2i}] & , \text{if } W_i = 1 \end{cases} \quad (6)$$

在進行 Super Tree 之位元平面量化時，我們須將該 Super Tree 之所有小波係數以二進位數值型式表示，使其成為一個位元平面，如圖 13 所示。在位元平面中，最下方為最低位元 2^0 ，最上方則為正負位元(Sign Bit)，因為一棵 Super Tree 是由 42 個係數所組成，故位元平面共有 42 行。

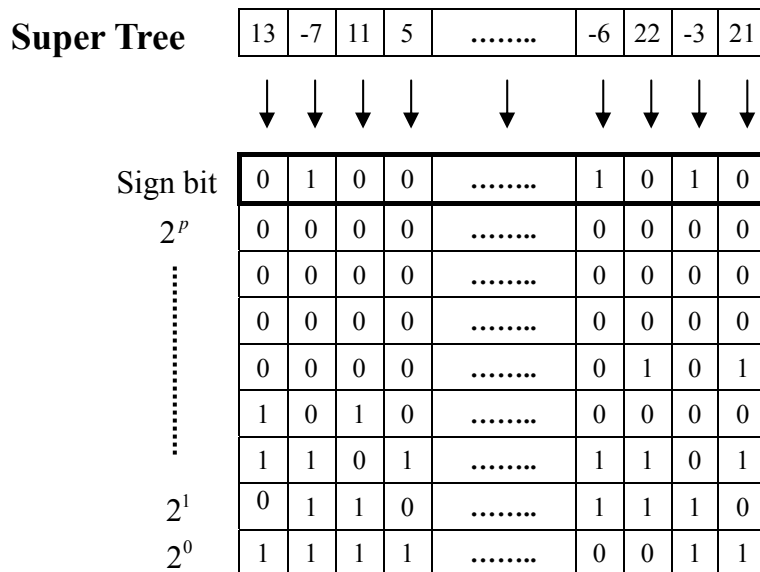


圖 13 Super Tree 轉換為二進位位元平面

在進行位元平面量化時，是由平面的右下角位元開始，依由右至左、由下而上的順序進行量化。在量化過程中，我們須設定一個量化參考誤差 ε (Reference Error)，以決定量化的量，之後必須找到一個量化參考索引值 q_n (Reference Index)， $n = (2i - 1, 2i)$ ，使 P_i 中的兩棵 Super Tree ST_{2i-1} 及 ST_{2i} 量化前後係數能量所產生的差值(即量化的總量)大於或等於參考誤差 ε ，即 $\varepsilon_{2i-1}(q_n) \geq \varepsilon$ 且 $\varepsilon_{2i}(q_n) \geq \varepsilon$ 且 $q_n \leq q_{\max}$ ，如此則 q_n 即決定了 $P_i(ST_{2i-1}, ST_{2i})$ 在量化過程中須進行量化的位元數。

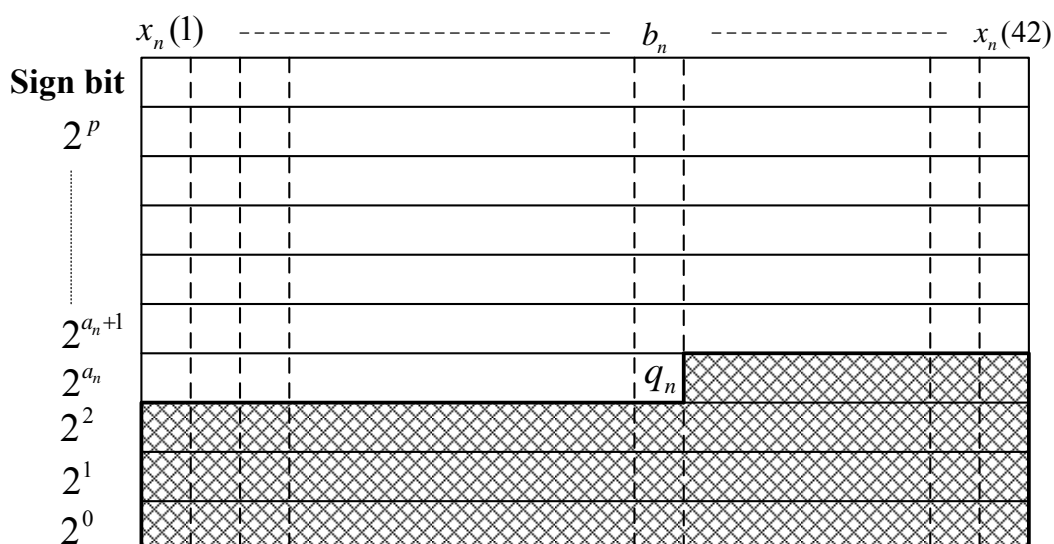


圖 14 Super Tree 量化示意圖

我們以 $ST_n = \{P_i(ST_{2i-1}, ST_{2i})\}$ 表示 $P_i(ST_{2i-1}, ST_{2i})$ 其中一棵 Super Tree，則在圖 14 中之斜線處即為 ST_n 係數中已量化的部份， q_n 則表示斜線部份之位元總數，假設 q_n 所在座標為 (a_n, b_n) ，在 q_n 左側的每個 Super Tree 係數之量化間距(quantization step) $\Delta_n(j)$ 為 2^{a_n} (包含 $j=b_n$ 時)，位於 q_n 右側的每個 Super Tree 係數之量化間距為 2^{a_n+1} ，則該 Super Tree 係數 $x_n(j)$ 在量化前後之差值 $e_n(j)$ 定義如下：

$$e_n(j) = Q[x_n(j)]_{q_n} - x_n(j) \quad (7)$$

對於不同量化間距之係數，其量化之關係式如下：

$$Q[x_n(j)] = \begin{cases} \text{round}(x_n(j))_{a_n} & , j \leq b_n \\ \text{round}(x_n(j))_{a_n+1} & , \text{otherwise} \end{cases} \quad (8)$$

如此則該棵 Super Tree 係數量化的總和為 $\varepsilon_n(q_n) = \sum_{j=1}^{42} |e_n(j)|$ ，此即為因量化而造成的資料損失量，若 $\varepsilon_n(q_n)$ 愈大則影像品質受到的影響即愈大，但相對的浮水印也較具強韌性。

若是欲嵌入之浮水印位元 $W_i = -1$ 時，則我們對於 Super Tree Pair P_i 中的 ST_{2i-1} 作量化，且係數量化之位元數須達 q_n 個位元；若是 $W_i = 1$ 時，則是對 ST_{2i} 量化 q_n 個位元，如此在量化完成後，不論 W_i 值為何，其相對應之 Super Tree 在量化中所減少之能量均會大於或等於 ε ，也就是我們當初所設定之量化強度。

之後我們依序在 Super Tree Pair P_i 中找到正確的 q_n 值，並針對每一個浮水印位元值 W_i 而進行 $P_i(ST_{2i-1}, ST_{2i})$ 的量化，直到完成所有浮水印位元的嵌入。當所有浮水印皆完成嵌入後，必須將 1536 棵 Super Tree 中的小波樹排列順序恢復成初始狀態，再透過反小波轉換(inverse-DWT)將影像自頻率域轉換回空間域，成為一張已加入浮水印的影像，至此則完成了整個浮水印的嵌入程序。

3.2 浮水印取出流程

在浮水印取出流程中，我們必須對同一組 Super Tree Pair 內的兩棵 Super Tree 作再量化(Re-quantization)，由再量化前後係數能量所產生的差值，我們即可依據 [1] 所定義之常態再量化誤差量累進分佈函數(CDF of the Magnitude of Normalized Re-quantization

Errors)，以判斷出浮水印位元值。之後則再經由浮水印的驗證程序，以確認該浮水印是否存在。小波樹量化數位影像浮水印之浮水印取出流程如圖 15 所示：

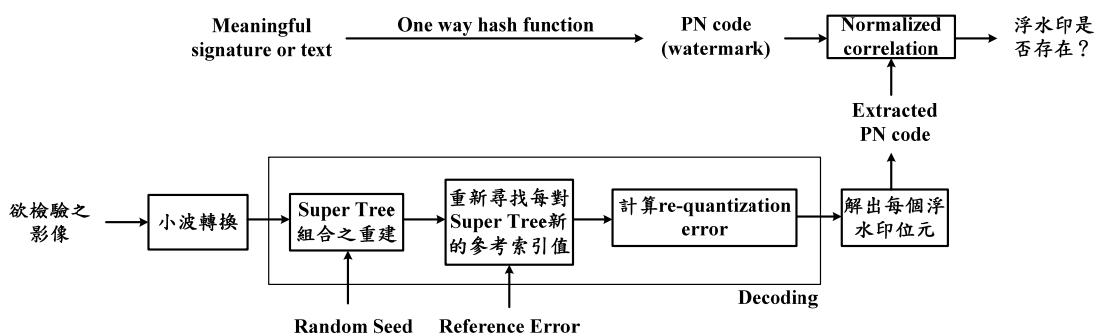


圖 15 小波樹量化數位影像浮水印之取出流程圖

在浮水印取出流程的初始步驟，我們同樣必須將待測影像經過四層的二維小波轉換，使影像自空間域轉換至頻率域，並分離出 13 個子頻帶。接著依四元樹之空間位置對應原則，將不同階層之小波係數組合為小波樹，共可產生 3072 棵小波樹。

之後我們使用與浮水印嵌入時相同的亂數種子，以同樣方式將 3072 棵小波樹中每兩棵小波樹組成一棵 Super Tree，再將每兩棵 Super Tree 結合成一組 Super Tree Pair，如此可分為 768 組 Super Tree Pair，且組成之順序與浮水印嵌入時之順序相同。

在取出浮水印時，我們必須將 Super Tree Pair P'_i 的兩棵 Super Tree ST'_{2i-1} 及 ST'_{2i} 所有小波係數以二進位數值位元平面型式表示，之後須找到一個最小值 q'_n ，使 ST'_{2i-1} 或 ST'_{2i} 再量化前後係數能量所產生的差值大於或等於原先嵌入時所設定的參考誤差 ε ，即 $\varepsilon'_{2i-1}(q'_n) \geq \varepsilon$ 或 $\varepsilon'_{2i}(q'_n) \geq \varepsilon$ ，且 $q_n \leq q_{\max}$ ，接著則根據 q'_n 對 ST'_{2i-1} 及 ST'_{2i} 進行再量化，以得到兩棵 Super Tree 每個係數的再量化誤差值(Re-quantization Error) $e'_{2i-1}(j)$ 及 $e'_{2i}(j)$ ，其定義如下式：

$$e'_l(j) = x'_l(j) - Q[x'_l(j)]_{q'_l} \quad (9)$$

上式中， $l = 2i - 1, 2i$ ，即Super Tree的順序索引， $x'_l(j)$ 表示第 l 個Super Tree中的第 j 個係數值， $Q[x'_l(j)]_{q'_l}$ 則為依據 q'_l 對 $x'_l(j)$ 重新量化後的係數值。當 $P'_i(ST'_{2i-1}, ST'_{2i})$ 中的每個係數再量化誤差值 $e'_{2i-1}(j)$ 與 $e'_{2i}(j)$ 被計算之後，[1]定義了常態再量化誤差量累進分佈函數，定義如下：

$$f(y) = \text{prob} \left[\left| \frac{e'_l(j)}{\Delta_l(j)} \right| < y \right] \quad (10)$$

上式中， $f(y)$ 為Super Tree l 的第 j 個係數值之再量化誤差量(re-quantization error magnitude) $e'_l(j)$ 與量化間距 $\Delta_l(j)$ 之比例的絕對值，低於門檻值 y 的機率。假設 ST'_{2i-1} 滿足 $f(y)$ 的個數共有 N_{2i-1} 個，而 ST'_{2i} 中滿足 $f(y)$ 的個數共有 N_{2i} 個，則偵測出的浮水印位元值 W'_i 之判斷式如下：

$$W'_i = \begin{cases} -1 & , \text{if } N_{2i-1} > N_{2i} \\ 1 & , \text{otherwise} \end{cases} \quad (11)$$

當所有浮水印位元 W'_i 皆由 p'_i 中取出後，解出之浮水印 W' 與原始浮水印 W 即可進行浮水印之驗證程序。

3.3 浮水印的驗證

在自待測影像中取出浮水印後，若沒有經過一套完善的驗證程序，我們仍無法確定此浮水印是否與原先嵌入之浮水印相同。[1]所使用之浮水印驗證方法為相關性浮水印偵

測(correlation-based watermark detection)，是將取出之浮水印 W' 與原先嵌入之浮水印 W 進行相關係數 ρ 的計算，運算方式採用常態相關性(Normalized Correlation)作為相關係數運算式(Correlation Value Calculator)，若令浮水印長度為 N_w ，則其定義如(12)式：

$$\rho(W, W') = \frac{\sum_{n=1}^{N_w} W_n W'_n}{\sqrt{\sum_{n=1}^{N_w} W_n^2 \sum_{n=1}^{N_w} W'_n{}^2}} \quad (12)$$

在經過常態相關性運算後，可得到一個相關係數 ρ ，我們再將 ρ 與設定之相關性門檻值 ρ_T 進行比較，若 $\rho \geq \rho_T$ ，則表示 W' 與 W 是相同的，即浮水印確實存在於此待測影像中；但若 $\rho < \rho_T$ ，則表示 W' 與 W 並不相同，即浮水印並不存在於待測影像中。



在本論文中，我們將相關性門檻值 ρ_T 設定為 0.23，以使誤判率(False positive probability)維持在 1.03×10^{-7} ，此亦與 [1]所設定之門檻值相同。

3.4 安全性缺失

由上述可以得知，參考文獻 [1]所提出之演算法，主要是藉由一組Super Tree Pair中的兩棵Super Tree之能量差異，以嵌入浮水印，又因為每棵Super Tree皆由兩棵小波樹所組成，故最多僅能嵌入 768 個浮水印位元。

然而參考文獻 [28]針對小波樹量化浮水印演算法，指出了該方法在安全性上之重大缺失，惡意攻擊者可以利用此一缺失，輕易的將隱藏其中的浮水印資訊消除。其所使用

的方法主要是透過密碼學方面的分析，基於對小波樹的組合方式為已知，對小波係數值加以分析推算，進而將係數間的差異消除。其分析方法大致可分為三個步驟：

一、依量化與否對影像之所有小波樹作分類：

由於小波樹之結構固定為第四、三、二階層具有空間對應關係的 21 個小波中頻係數所組成，故我們可以輕易的將影像經由小波轉換至頻率域，以得到正確的小波樹組成。之後我們再將每一棵小波樹之係數以二進位型式排列成位元平面，並依序檢查該位元平面的最低兩行位元值，若位元值大多數皆為 0，則該棵小波樹則可能被量化過，我們將其歸類為已量化的小波樹；若是位元值有 0 有 1，則歸類為未量化的小波樹，如此則所有小波樹將被依是否量化而區分為兩種群組。

二、推算出可能的量化誤差值：

在所有小波樹皆依是否量化而有所區分之後，我們即可經由對已量化的小波樹作分析，推算出每棵小波樹之量化範圍，進而得到浮水印嵌入時所設定的量化參考誤差，即量化運算時至少須達到之量化的量。

三、使用量化誤差值對於未量化過的小波樹進行量化：

在計算出可能的量化參考誤差後，我們即可依參考誤差值對每一棵未量化的小波樹進行量化，如此則所有小波樹皆為已量化，而原本為嵌入浮水印而造成的能量差異即因此被消除，當欲取出浮水印時，會因為無法判斷而造成錯誤，達到成功消除浮水印的目的。

由以上三個步驟可以得知，[28]提出之攻擊方法雖然無法得知Super Tree的排列順序，但它採用對組成Super Tree之小波樹係數進行數值分析，進而以量化方式將不同小波樹之間的能量差異降至最低，最後亦能達到除去浮水印資訊之目的。

有鑑於此，本論文即提出另一種以小波轉換為基礎之浮水印演算法，將 [1]所使用之不同小波樹間的能量差異比較方法轉變為單一小波樹內部係數間的能量差異比較，針對小波樹不同階層之小波係數進行動態能量差異化運算，並藉由差異化以嵌入浮水印。因為每一棵小波樹皆能嵌入一個浮水印位元，故此方法能夠嵌入的浮水印位元長度增加到 3072 個位元。此外我們並於演算法中加入防止攻擊者使用係數分析攻擊的機制，以及利用不同階層係數進行浮水印位元重覆嵌入的方法，進而有效確保此浮水印演算法之強韌性與安全性。



第四章 以動態能量差異化為基礎之浮水印演算法

本章將提出以動態能量差異化為基礎之浮水印演算法，於 4.1 節說明浮水印的嵌入流程，包含本文所使用的浮水印產生機制、係數能量差異化方向的選擇與計算差異化的數學運算式；4.2 節則說明浮水印的取出流程，取出之浮水印將於 4.3 節作相關性係數的計算與比較，以驗證該浮水印是否確實存在。

此外，本論文於 4.4 節提出一種影像切割重組的前置處理方式，在影像進行浮水印嵌入程序之前，先行將原始影像切割成 4×4 小區塊子影像，並依亂數排列重新組合成具馬賽克效果的新影像，之後再以新影像進行的浮水印嵌入流程。此前置處理方法可有效提升浮水印演算法的安全性，並能夠增加浮水印對於空間域影像處理攻擊的抵抗能力。

4.1 以動態能量差異化為基礎之浮水印嵌入流程

以動態能量差異化為基礎之浮水印嵌入流程，主要是透過將小波轉換後的中頻係數組成小波樹，再利用小波係數間之能量差異以嵌入浮水印，其流程如圖 16 所示，本節將針對每一個步驟之內容進行探討。

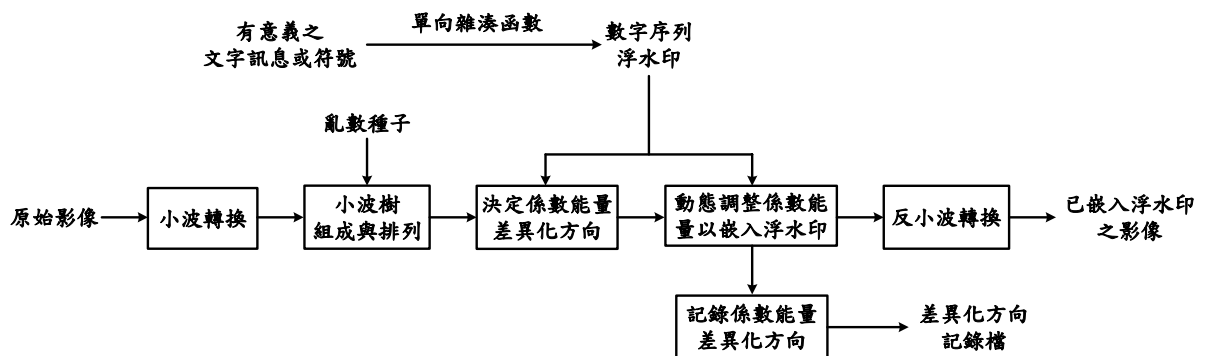


圖 16 以動態能量差異化為基礎之浮水印嵌入流程

4.1.1 浮水印產生

本篇論文所使用之浮水印亦為數字序列式浮水印，我們同樣是藉由亂數種子的選擇，以隨機方式產生一連串值為+1或-1的PN序列整數，至於浮水印的長度則可隨需求而自由決定。浮水印產生機制所使用的演算法與[1]相同。

4.1.2 小波轉換程序與小波樹的組成

本方法所使用之小波轉換程序與小波樹的組成，其設定架構與[1]相同。首先我們必須將原始影像經過四層的二維小波轉換，使影像自空間域轉換至頻率域，並分解為13個子頻帶。接著依四元樹之空間位置對應原則，將不同階層之21個小波係數結合為小波樹，如此將可得到3072棵小波樹。在嵌入浮水印位元時，我們即必須針對組成小波樹的係數能量進行差異化運算。小波樹之係數表示式如下：

$$\mathcal{T}_k(i) \quad \text{for } 1 \leq k \leq 3072 \quad \text{and} \quad 1 \leq i \leq 21 \quad (13)$$



在(13)式中， k 為小波樹之順序指標(index)， i 則為小波樹內之小波係數指標。由於一張影像的小波樹個數為3072棵，其中每一棵小波樹皆可嵌入一個浮水印位元，故此方法所能嵌入的最大浮水印長度即為3072個位元。為了增加浮水印的安全性，我們在進行小波樹係數能量差異化運算以嵌入浮水印之前，必須先使用與產生浮水印相同之亂數種子，透過隨機排序涵式 $f(\cdot)$ 對3072棵小波樹進行隨機的排序，使惡意攻擊者無法得知小波樹在嵌入浮水印時真正的排列組合順序，自然也無法取出正確順序的浮水印位元。小波樹隨機排序表示式如下：

$$\mathcal{T}_n = f(\mathcal{T}_k) \quad \text{for } 1 \leq n \leq 3072 \quad \text{and} \quad 1 \leq k \leq 3072 \quad (14)$$

4.1.3 動態能量差異化之浮水印嵌入方式

動態能量差異化之浮水印嵌入方式，主要是藉由小波樹係數能量經動態改變後所產生的差異以藏入浮水印。小波樹係數結構是由 1 個第四階層小波係數、4 個第三階層小波係數與 16 個第二階層小波係數所組成，因為各個階層的小波係數性質與數量皆不同，故在進行係數能量差異化的過程中，我們將分別以不同方法處理第四階層與第二、三階層之係數能量，如下圖 17：

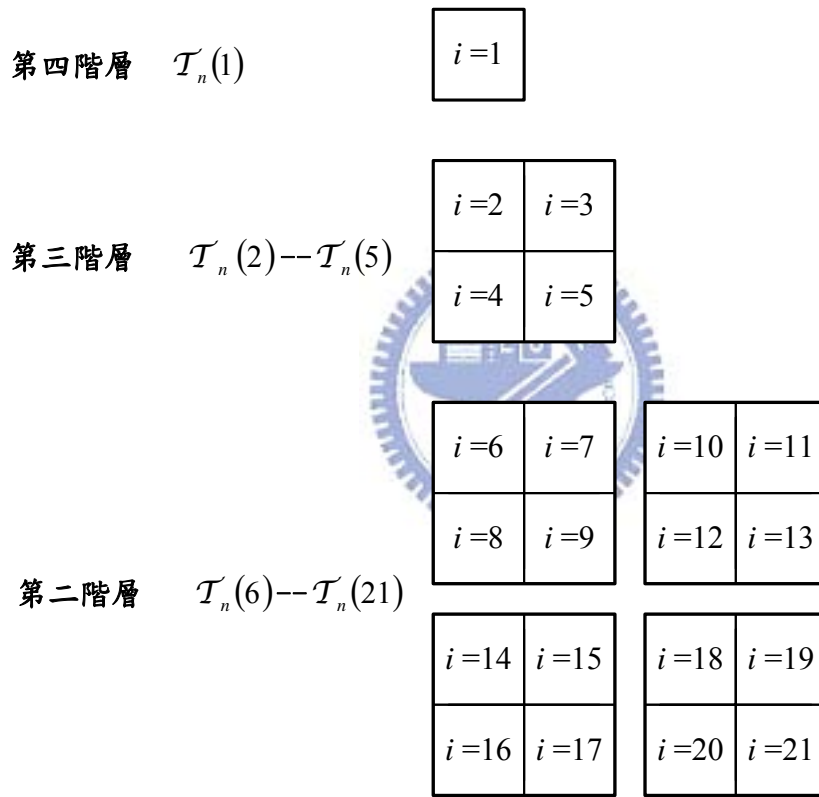


圖 17 小波樹不同階層之係數結構示意圖

浮水印嵌入時，須按照重新排列後的小波樹順序，將每一個浮水印位元 W_n 依次重覆嵌入小波樹 \mathcal{T}_n 不同階層的小波係數之中，如(15)式。

$$\mathcal{T}'_n = f[\mathcal{T}_n(i)] \quad \text{for } 1 \leq n \leq 3072 \quad \text{and} \quad 1 \leq i \leq 21 \quad (15)$$

在第四階層的小波係數部份，因為僅含有 1 個係數，故採用單一係數能量差異化的方式 [29]，將該係數值經由特定運算方式加以量化，以嵌入浮水印位元值；而第二、三階層的小波係數部份，由於小波係數之數量皆為 4 的倍數，故我們取二維位置相鄰 2×2 合計 4 個小波係數為一組，並依每組係數之能量差異特性選定差異化方向，最後再依各組係數之差異化方向將浮水印位元 W_n 進行重覆嵌入。如此原則上在一棵小波樹 T_n 之中，浮水印位元嵌入之次數為第四階層 1 次、第三階層 1 次、第二階層 4 次，合計共嵌入 6 次浮水印位元值。

然而我們為了防止惡意攻擊者利用係數能量差異分析法以消除嵌入之浮水印，故在第三階層之浮水印位元嵌入時，運用不同之差異化方向將 W_n 及 $-W_n$ 均嵌入其中；而在第二階層之浮水印位元嵌入程序中，則是選取其中兩組係數嵌入 W_n ，而另兩組係數卻嵌入 $-W_n$ ，如此則攻擊者無法確認應該對小波樹第二階層的哪幾組係數進行分析，也無法從第三階層係數中分析出唯一的浮水印位元值。當我們欲取出浮水印時，則可以針對 6 個嵌入的有意義浮水印位元值加以比對，以提高取出之浮水印的真確性。

4.1.4 第四階層小波係數之浮水印嵌入

小波樹 T_n 在第四階層之小波係數僅有 1 個，如圖 18，無法與同階層其他係數形成能量差異化之關係，故在浮水印嵌入部份，我們採用單一係數能量差異化的方式，對係數值加以量化，以達到浮水印位元值嵌入之目的。

Level 4

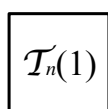


圖 18 第四階層之小波係數

首先我們必須設定一個 modular 值 S ，並定義兩個差異化門檻值(threshold) T_1 與 T_2 如下：

$$T_1 = \frac{S}{4} \quad \text{and} \quad T_2 = \frac{3S}{4} \quad (16)$$

modular 值 S 將決定係數能量差異化的大小，當 S 愈大，則表示係數在量化過程中的差異化愈大，同時浮水印的強韌性亦提高，但影像品質會因此下降。隨後我們將係數能量值以(17)式進行量化運算如下：

$$\delta_n(1) = \mathcal{T}_n(1) \bmod S \quad (17)$$

之後我們將視欲嵌入的浮水印位元 W_n 為何，以及小波係數值 $\mathcal{T}_n(1)$ 之正負號性質，動態進行能量差異化之運算如下：

$$\mathcal{T}'_n(1) = \begin{cases} \mathcal{T}_n(1) - \delta_n(1) + T_1, & \text{if } W_n = -1 \quad \text{and} \quad \mathcal{T}_n(1) \geq 0 \\ \mathcal{T}_n(1) - \delta_n(1) + T_2, & \text{if } W_n = +1 \quad \text{and} \quad \mathcal{T}_n(1) \geq 0 \\ \mathcal{T}_n(1) + \delta_n(1) - T_1, & \text{if } W_n = -1 \quad \text{and} \quad \mathcal{T}_n(1) < 0 \\ \mathcal{T}_n(1) + \delta_n(1) - T_2, & \text{if } W_n = +1 \quad \text{and} \quad \mathcal{T}_n(1) < 0 \end{cases} \quad (18)$$

經(18)式運算過後之第四階層小波係數 $\mathcal{T}'_n(1)$ ，其能量之正負號不變，但數值會因欲嵌入的浮水印位元 W_n 之不同而產生差異化的改變，我們即是利用此差異化將 W_n 嵌入小波係數 $\mathcal{T}'_n(1)$ 之中，至此則完成了第四階層小波係數之浮水印嵌入。

4.1.5 第二、三階層小波係數之浮水印嵌入

在小波樹結構 \mathcal{T}_n 之中，第三階層之小波係數有 4 個，第二階層之小波係數則有 16

個，在同階層之中均有多個係數可形成能量差異化之關係，故在浮水印嵌入部份，我們採用 4 個係數為一組的能量差異化方式，對同一組的係數值加以動態改變之間的差異，以達到浮水印位元值嵌入之目的。

我們取二維位置相鄰之 4 個小波係數為一組，故第三階層之小波係數可組成一組，而第二階層之小波係數則組成四組，如圖 19，之後我們須依每組係數之能量差異特性選定其最適當的差異化方向，再依各組係數之差異化方向進行這 5 組係數之浮水印位元嵌入程序。

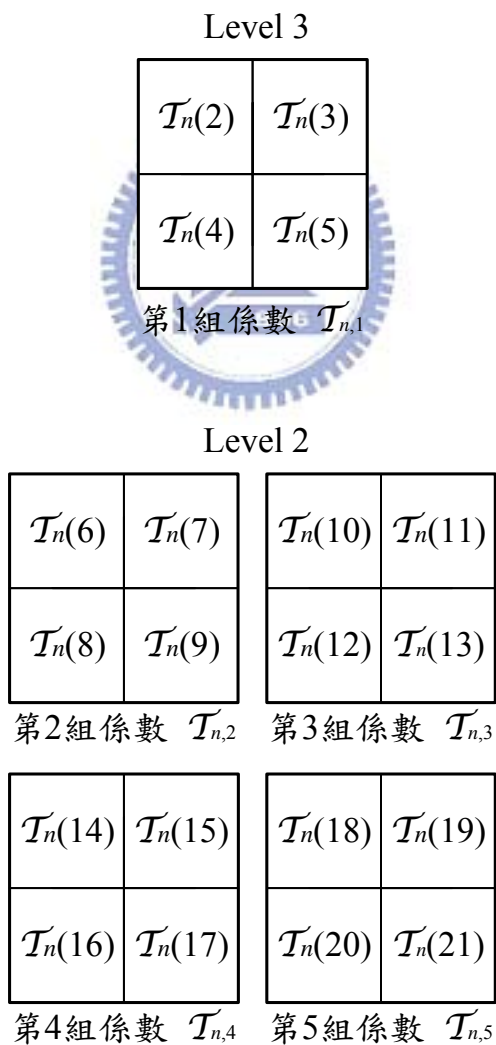


圖 19 第二、三階層小波係數之分組示意圖

為了方便說明，我們將小波樹 \mathcal{T}_n 第二、三階層之每一組小波係數能量以 $\mathcal{T}_{n,g}(j)$ 表示如下：

$$\mathcal{T}_{n,g}(j) = \mathcal{T}_n(j + 4g - 3) \quad \text{for } 1 \leq g \leq 5 \quad \text{and } 1 \leq j \leq 4 \quad (19)$$

在(19)式中， n 為小波樹排列之索引， g 為小波樹中第二、三層係數編組之索引， j 則為同組係數中係數之索引，由(19)式之表示法可得各組之小波係數圖示如下：

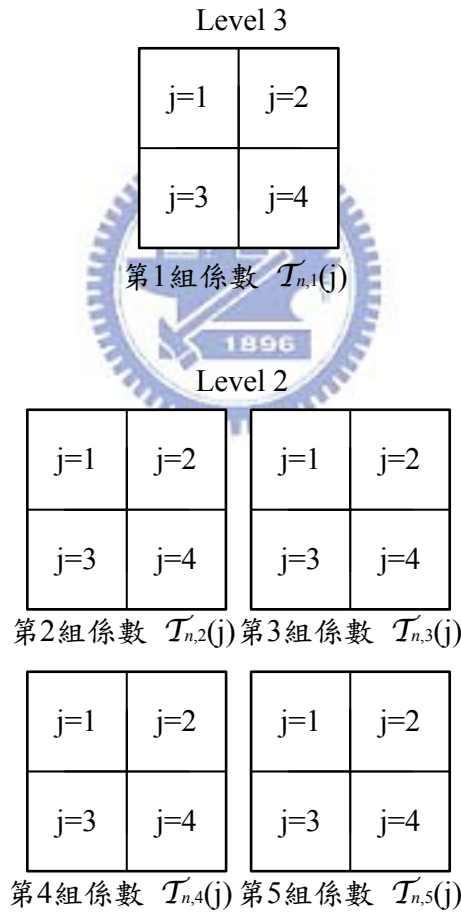


圖 20 第二、三階層各組之小波係數圖示

由上圖我們可以得知，在一棵小波樹中，5 組係數之分組如下：

$$\left\{ \begin{array}{l} \mathcal{T}_{n,1} = \{ \mathcal{T}_n(2), \mathcal{T}_n(3), \mathcal{T}_n(4), \mathcal{T}_n(5) \} \\ \mathcal{T}_{n,2} = \{ \mathcal{T}_n(6), \mathcal{T}_n(7), \mathcal{T}_n(8), \mathcal{T}_n(9) \} \\ \mathcal{T}_{n,3} = \{ \mathcal{T}_n(10), \mathcal{T}_n(11), \mathcal{T}_n(12), \mathcal{T}_n(13) \} \\ \mathcal{T}_{n,4} = \{ \mathcal{T}_n(14), \mathcal{T}_n(15), \mathcal{T}_n(16), \mathcal{T}_n(17) \} \\ \mathcal{T}_{n,5} = \{ \mathcal{T}_n(18), \mathcal{T}_n(19), \mathcal{T}_n(20), \mathcal{T}_n(21) \} \end{array} \right. \quad (20)$$

在以動態能量差異化嵌入浮水印時，若一棵小波樹的每一組係數均嵌入相同的浮水印位元值，則惡意攻擊者可能藉由參照比對每組係數可能嵌入的浮水印位元值，進而推算出該小波樹最有可能隱藏的浮水印位元。有鑑於此，我們藉由亂數種子所隨機排列出的小波樹順序，以選擇某幾組係數嵌入真的浮水印位元值，剩餘幾組係數則嵌入相反的浮水印位元值。因為攻擊者無法得知小波樹的排列順序，故將無法確認應該對小波樹第二、三階層的哪幾組係數進行分析。



為了防止惡意攻擊者可能藉由係數能量差異進行分析，進而推算出嵌入之浮水印位元值並消除浮水印，故當我們在進行第二、三階層小波係數之浮水印嵌入時，必須採用下列運算關係式：

$$\mathcal{T}'_{n,g} = E[\mathcal{T}_{n,g}(j)] \left\{ \begin{array}{l} g=1,2,4 \text{ for } W_n \text{ and } g=1,3,5 \text{ for } -W_n, 1 \leq j \leq 4, \text{ if } n \% 2 = 1 \\ g=1,3,5 \text{ for } W_n \text{ and } g=1,2,4 \text{ for } -W_n, 1 \leq j \leq 4, \text{ if } n \% 2 = 0 \end{array} \right. \quad (21)$$

在(21)式中， n 為小波樹的指標，即經過亂數排列後的順序編號，當欲嵌入浮水印位元的小波樹順序指標為 1、3、5...時，我們將浮水印位元 W_n 嵌入該小波樹第二、三階層的第一、二、四組係數中，並將浮水印位元之相反值 $-W_n$ 嵌入第一、三、五組係數中；若小波樹順序指標為 2、4、6...時，我們則將浮水印位元 W_n 嵌入該小波樹第二、三階層

的第一、三、五組係數中，並將浮水印位元之相反值 $-W_n$ 嵌入第一、二、四組係數中。因為第一組小波係數屬於第三階層係數，相較於第三階層係數而言，其數值具有較不易被修改的優點，故不論小波樹的指標 n 數值為何，都應該要將浮水印位元 W_n 嵌入其中。然而為了避免攻擊者透過分析第一組係數之能量差異狀態而得知 W_n 之值，故我們重複將 W_n 及 $-W_n$ 均嵌入第一組係數，如此則攻擊者在分析第一組係數之能量差異狀態時，會得出兩種相反之結論，致無法確定浮水印位元值。此外因為每一組係數之差異化方向會因 W_n 之正負值而必不相同，差異化之能量結構會同時存在，故重複嵌入 W_n 及 $-W_n$ ，並不致影響第一組係數之差異狀態。

每組係數之差異化方向可分為垂直方向(Vertical direction)、水平方向(Horizontal direction)與斜對角方向(Diagonal direction)三種，我們須依欲嵌入的浮水印位元值及該組係數之差異特性，從中選擇一種既符合需求且係數能量在運算後變化較微幅的最適差異化方向，以進行浮水印嵌入的運算。在 4.1.5.1 節至 4.1.5.3 節中，我們先說明三種差異化方向的浮水印嵌入運算方式，而後在 4.1.5.4 節中再說明最適差異化方向的選擇演算法。

4.1.5.1 垂直方向的浮水印嵌入

若差異化方向選定為垂直方向時，我們首先要針對該組小波係數 $\mathcal{T}_{n,g}(j)$ 分別計算出上方(Upper)及下方(Lower)之係數平均值如下：

$$\sigma_{n,g}(Ur) = \frac{\mathcal{T}_{n,g}(1) + \mathcal{T}_{n,g}(2)}{2} \quad (22)$$

$$\sigma_{n,g}(Lr) = \frac{\mathcal{T}_{n,g}(3) + \mathcal{T}_{n,g}(4)}{2} \quad (23)$$

接著須計算出垂直方向係數平均值的差距 dif_v ，如下式：

$$dif_v = |\sigma_{n,g}(Ur) - \sigma_{n,g}(Lr)| \quad (24)$$

我們取 $\sigma_{n,g}(Ur)$ 及 $\sigma_{n,g}(Lr)$ 作為係數差異化運算方式的依據。根據欲嵌入之浮水印位元 W_n 的值，我們將小波係數值經由特定運算式加以調整，以改變 $\sigma_{n,g}(Ur)$ 與 $\sigma_{n,g}(Lr)$ 的相對大小，(25)、(26)式即為一組小波係數上方與下方係數平均值改變後，須滿足的垂直方向之差異相對關係：

$$\sigma'_{n,g}(Ur) > \sigma'_{n,g}(Lr), \quad \text{if } W_n = -1 \quad (25)$$

$$\sigma'_{n,g}(Ur) < \sigma'_{n,g}(Lr), \quad \text{if } W_n = 1 \quad (26)$$

當我們在進行係數的差異化運算時，除了要使運算後 $\sigma'_{n,g}(Ur)$ 及 $\sigma'_{n,g}(Lr)$ 之大小關係符合上式之外，尚須決定係數能量差異化的強度。差異化強度取決於在特定方向差異化運算後之不同位置係數平均值差距，我們可設定適當之差異化間距 Δ ，並使上、下方係數平均值在差異化運算後所達到的差距值 $dif'_v \geq$ 差異化間距 Δ 。當差異化間距 Δ 愈大，則浮水印強度愈強，但影像品質也會跟著下降。若以 $E[\cdot]$ 表示差異化運算程序，則小波係數 $\mathcal{T}_{n,g}(j)$ 之能量調整計算公式如下：

$$E[\mathcal{T}_{n,g}(j)]_{W_n=-1}^{j=1,2} = \begin{cases} \mathcal{T}_{n,g}(j) + \frac{\Delta}{2} & , \text{if } \sigma_{n,g}(Ur) = \sigma_{n,g}(Lr) \\ \mathcal{T}_{n,g}(j) & , \text{if } \sigma_{n,g}(Ur) > \sigma_{n,g}(Lr) \text{ and } dif_v \geq \Delta \\ \mathcal{T}_{n,g}(j) + \frac{\Delta - dif_v}{2} & , \text{if } \sigma_{n,g}(Ur) > \sigma_{n,g}(Lr) \text{ and } dif_v < \Delta \\ \mathcal{T}_{n,g}(j) + \frac{dif_v + \Delta}{2} & , \text{if } \sigma_{n,g}(Ur) < \sigma_{n,g}(Lr) \end{cases} \quad (27)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=-1}^{j=3,4} = \begin{cases} \mathcal{T}_{n,g}(j) - \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(Ur) = \sigma_{n,g}(Lr) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(Ur) > \sigma_{n,g}(Lr) \text{ and } dif_v \geq \Delta \\ \mathcal{T}_{n,g}(j) - \frac{\Delta - dif_v}{2} & ,if \quad \sigma_{n,g}(Ur) > \sigma_{n,g}(Lr) \text{ and } dif_v < \Delta \\ \mathcal{T}_{n,g}(j) - \frac{dif_v + \Delta}{2} & ,if \quad \sigma_{n,g}(Ur) < \sigma_{n,g}(Lr) \end{cases} \quad (28)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=1}^{j=1,2} = \begin{cases} \mathcal{T}_{n,g}(j) - \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(Ur) = \sigma_{n,g}(Lr) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(Ur) < \sigma_{n,g}(Lr) \text{ and } dif_v \geq \Delta \\ \mathcal{T}_{n,g}(j) - \frac{\Delta - dif_v}{2} & ,if \quad \sigma_{n,g}(Ur) < \sigma_{n,g}(Lr) \text{ and } dif_v < \Delta \\ \mathcal{T}_{n,g}(j) - \frac{dif_v + \Delta}{2} & ,if \quad \sigma_{n,g}(Ur) > \sigma_{n,g}(Lr) \end{cases} \quad (29)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=1}^{j=3,4} = \begin{cases} \mathcal{T}_{n,g}(j) + \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(Ur) = \sigma_{n,g}(Lr) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(Ur) < \sigma_{n,g}(Lr) \text{ and } dif_v \geq \Delta \\ \mathcal{T}_{n,g}(j) + \frac{\Delta - dif_v}{2} & ,if \quad \sigma_{n,g}(Ur) < \sigma_{n,g}(Lr) \text{ and } dif_v < \Delta \\ \mathcal{T}_{n,g}(j) + \frac{dif_v + \Delta}{2} & ,if \quad \sigma_{n,g}(Ur) > \sigma_{n,g}(Lr) \end{cases} \quad (30)$$

在上述之係數能量差異化計算過程中，必須先忽略係數 $\mathcal{T}_{n,g}(j)$ 之正負位元 (sign bit)，一律取其絕對值作差異化計算，俟差異化完成後，再恢復其正負號，故運算過程中係數 $\mathcal{T}_{n,g}(j)$ 能量經調整後之最小值必須為0。

然而，若是某係數之能量在調整過程中之原始修正值為負數且 $\leq -2\Delta$ ，則當該係數為符合最小值必須為0之條件而將能量調整為0時，將會改變 $\sigma'_{n,g}(Ur)$ 及 $\sigma'_{n,g}(Lr)$ 之大小相對關係，進而影響到浮水印取出時之判斷，故若令 $\mathcal{T}'_{n,g}(j) = E[\mathcal{T}_{n,g}(j)]$ ，則我們須以下列式子作檢查與修正：

$$\left(\begin{array}{l} \mathcal{T}'_{n,g}(1) = \mathcal{T}'_{n,g}(1) - |\mathcal{T}'_{n,g}(2)| \quad ,if \quad \mathcal{T}'_{n,g}(2) \leq -2\Delta \quad and \quad W_n = 1 \\ \mathcal{T}'_{n,g}(2) = \mathcal{T}'_{n,g}(2) - |\mathcal{T}'_{n,g}(1)| \quad ,if \quad \mathcal{T}'_{n,g}(1) \leq -2\Delta \quad and \quad W_n = 1 \\ \mathcal{T}'_{n,g}(3) = \mathcal{T}'_{n,g}(3) - |\mathcal{T}'_{n,g}(4)| \quad ,if \quad \mathcal{T}'_{n,g}(4) \leq -2\Delta \quad and \quad W_n = -1 \\ \mathcal{T}'_{n,g}(4) = \mathcal{T}'_{n,g}(4) - |\mathcal{T}'_{n,g}(3)| \quad ,if \quad \mathcal{T}'_{n,g}(3) \leq -2\Delta \quad and \quad W_n = -1 \end{array} \right. \quad (31)$$

4.1.5.2 水平方向的浮水印嵌入

若差異化方向選定為水平方向時，我們則須針對該組小波係數 $\mathcal{T}_{n,g}(j)$ 分別計算出左方(Left)及右方(Right)之係數平均值如下：

$$\sigma_{n,g}(Lt) = \frac{\mathcal{T}_{n,g}(1) + \mathcal{T}_{n,g}(3)}{2} \quad (32)$$

$$\sigma_{n,g}(Rt) = \frac{\mathcal{T}_{n,g}(2) + \mathcal{T}_{n,g}(4)}{2} \quad (33)$$

接著須計算出水平方向係數平均值的差距 dif_h ，如下式：

$$dif_h = |\sigma_{n,g}(Lt) - \sigma_{n,g}(Rt)| \quad (34)$$

我們取 $\sigma_{n,g}(Lt)$ 及 $\sigma_{n,g}(Rt)$ 作為係數差異化運算方式的依據。根據欲嵌入之浮水印位元 W_n 的值，我們同樣將小波係數值經由特定運算式加以調整，以改變 $\sigma_{n,g}(Lt)$ 與 $\sigma_{n,g}(Rt)$ 的相對大小，(35)、(36)式即為改變後須滿足之水平方向差異相對關係：

$$\sigma'_{n,g}(Lt) > \sigma'_{n,g}(Rt), \quad \text{if } W_n = -1 \quad (35)$$

$$\sigma'_{n,g}(Lt) < \sigma'_{n,g}(Rt), \quad \text{if } W_n = 1 \quad (36)$$

當我們在進行係數的差異化運算時，除了要使運算後 $\sigma'_{n,g}(Lt)$ 及 $\sigma'_{n,g}(Rt)$ 之大小關係符合上式之外，尚須使左、右兩方係數平均值在差異化運算後所達到的差距值 $dif'_h \geq$ 差異化間距 Δ 。若以 $E[\cdot]$ 表示差異化運算程序，則小波係數 $\mathcal{T}_{n,g}(j)$ 之能量調整計算公式如下：

$$E[\mathcal{T}_{n,g}(j)]_{W_n=-1}^{j=1,3} = \begin{cases} \mathcal{T}_{n,g}(j) + \frac{\Delta}{2} & , \text{if } \sigma_{n,g}(Lt) = \sigma_{n,g}(Rt) \\ \mathcal{T}_{n,g}(j) & , \text{if } \sigma_{n,g}(Lt) > \sigma_{n,g}(Rt) \text{ and } dif_h \geq \Delta \\ \mathcal{T}_{n,g}(j) + \frac{\Delta - dif_h}{2} & , \text{if } \sigma_{n,g}(Lt) > \sigma_{n,g}(Rt) \text{ and } dif_h < \Delta \\ \mathcal{T}_{n,g}(j) + \frac{dif_h + \Delta}{2} & , \text{if } \sigma_{n,g}(Lt) < \sigma_{n,g}(Rt) \end{cases} \quad (37)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=-1}^{j=2,4} = \begin{cases} \mathcal{T}_{n,g}(j) - \frac{\Delta}{2} & , \text{if } \sigma_{n,g}(Lt) = \sigma_{n,g}(Rt) \\ \mathcal{T}_{n,g}(j) & , \text{if } \sigma_{n,g}(Lt) > \sigma_{n,g}(Rt) \text{ and } dif_h \geq \Delta \\ \mathcal{T}_{n,g}(j) - \frac{\Delta - dif_h}{2} & , \text{if } \sigma_{n,g}(Lt) > \sigma_{n,g}(Rt) \text{ and } dif_h < \Delta \\ \mathcal{T}_{n,g}(j) - \frac{dif_h + \Delta}{2} & , \text{if } \sigma_{n,g}(Lt) < \sigma_{n,g}(Rt) \end{cases} \quad (38)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=1}^{j=1,3} = \begin{cases} \mathcal{T}_{n,g}(j) - \frac{\Delta}{2} & , \text{if } \sigma_{n,g}(Lt) = \sigma_{n,g}(Rt) \\ \mathcal{T}_{n,g}(j) & , \text{if } \sigma_{n,g}(Lt) < \sigma_{n,g}(Rt) \text{ and } dif_h \geq \Delta \\ \mathcal{T}_{n,g}(j) - \frac{\Delta - dif_h}{2} & , \text{if } \sigma_{n,g}(Lt) < \sigma_{n,g}(Rt) \text{ and } dif_h < \Delta \\ \mathcal{T}_{n,g}(j) - \frac{dif_h + \Delta}{2} & , \text{if } \sigma_{n,g}(Lt) > \sigma_{n,g}(Rt) \end{cases} \quad (39)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=1}^{j=2,4} = \begin{cases} \mathcal{T}_{n,g}(j) + \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(Lt) = \sigma_{n,g}(Rt) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(Lt) < \sigma_{n,g}(Rt) \text{ and } dif_h \geq \Delta \\ \mathcal{T}_{n,g}(j) + \frac{\Delta - dif_h}{2} & ,if \quad \sigma_{n,g}(Lt) < \sigma_{n,g}(Rt) \text{ and } dif_h < \Delta \\ \mathcal{T}_{n,g}(j) + \frac{dif_h + \Delta}{2} & ,if \quad \sigma_{n,g}(Lt) > \sigma_{n,g}(Rt) \end{cases} \quad (40)$$

在上述之係數能量差異化計算過程中，我們同樣必須先忽略係數之正負位元，一律取其絕對值作差異化計算，俟計算完成後，再恢復其正負號，故運算過程中係數 $\mathcal{T}_{n,g}(j)$ 之能量經調整後最小值必須為 0。

同樣的，若係數之能量在調整過程中之原始修正值為負數，且數值 $\leq -2\Delta$ ，則當該係數能量調整為 0 時，將會改變 $\sigma'_{n,g}(Lt)$ 及 $\sigma'_{n,g}(Rt)$ 之大小相對關係，進而影響到浮水印取出時之判斷，故我們須以下列式子作水平方向差異化運算之檢查與修正：

$$\left\{ \begin{array}{l} \mathcal{T}'_{n,g}(1) = \mathcal{T}'_{n,g}(1) - |\mathcal{T}'_{n,g}(3)| \quad ,if \quad \mathcal{T}'_{n,g}(3) \leq -2\Delta \text{ and } W_n = 1 \\ \mathcal{T}'_{n,g}(3) = \mathcal{T}'_{n,g}(3) - |\mathcal{T}'_{n,g}(1)| \quad ,if \quad \mathcal{T}'_{n,g}(1) \leq -2\Delta \text{ and } W_n = 1 \\ \mathcal{T}'_{n,g}(2) = \mathcal{T}'_{n,g}(2) - |\mathcal{T}'_{n,g}(4)| \quad ,if \quad \mathcal{T}'_{n,g}(4) \leq -2\Delta \text{ and } W_n = -1 \\ \mathcal{T}'_{n,g}(4) = \mathcal{T}'_{n,g}(4) - |\mathcal{T}'_{n,g}(2)| \quad ,if \quad \mathcal{T}'_{n,g}(2) \leq -2\Delta \text{ and } W_n = -1 \end{array} \right. \quad (41)$$

4.1.5.3 斜對角方向的浮水印嵌入

若差異化方向為斜對角(Diagonal)方向時，我們則須針對該組小波係數 $\mathcal{T}_{n,g}(j)$ 分別計算出左斜對角(Diagonal 1)及右斜對角(Diagonal 2)之係數平均值如下：

$$\sigma_{n,g}(D_1) = \frac{\mathcal{T}_{n,g}(1) + \mathcal{T}_{n,g}(4)}{2} \quad (42)$$

$$\sigma_{n,g}(D_2) = \frac{\mathcal{T}_{n,g}(2) + \mathcal{T}_{n,g}(3)}{2} \quad (43)$$

接著須計算出斜對角方向係數平均值的差距 dif_d ，如下式：

$$dif_d = |\sigma_{n,g}(D_1) - \sigma_{n,g}(D_2)| \quad (44)$$

我們同樣取 $\sigma_{n,g}(D_1)$ 及 $\sigma_{n,g}(D_2)$ 作為係數差異化運算方式的依據。根據欲嵌入之浮水印位元 W_n 之值，將小波係數值經由特定運算式加以調整，以改變 $\sigma_{n,g}(D_1)$ 與 $\sigma_{n,g}(D_2)$ 的相對大小，(45)、(46)式即為改變後須滿足之水平方向差異相對關係：

$$\sigma'_{n,g}(D_1) > \sigma'_{n,g}(D_2), \quad \text{if } W_n = -1 \quad (45)$$

$$\sigma'_{n,g}(D_1) < \sigma'_{n,g}(D_2), \quad \text{if } W_n = 1 \quad (46)$$

當我們在進行係數的差異化運算時，除了要使運算後 $\sigma'_{n,g}(D_1)$ 及 $\sigma'_{n,g}(D_2)$ 之大小關係符合上式之外，尚須使雙方係數平均值在差異化運算後所達到的差距值 $dif'_d \geq$ 差異化間距 Δ 。若以 $E[\cdot]$ 表示差異化運算程序，則小波係數 $\mathcal{T}_{n,g}(j)$ 之能量調整計算公式如下：

$$E[\mathcal{T}_{n,g}(j)]_{W_n=-1}^{j=1,4} = \begin{cases} \mathcal{T}_{n,g}(j) + \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(D_1) = \sigma_{n,g}(D_2) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(D_1) > \sigma_{n,g}(D_2) \text{ and } dif_d \geq \Delta \\ \mathcal{T}_{n,g}(j) + \frac{\Delta - dif_d}{2} & ,if \quad \sigma_{n,g}(D_1) > \sigma_{n,g}(D_2) \text{ and } dif_d < \Delta \\ \mathcal{T}_{n,g}(j) + \frac{dif_d + \Delta}{2} & ,if \quad \sigma_{n,g}(D_1) < \sigma_{n,g}(D_2) \end{cases} \quad (47)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=-1}^{j=2,3} = \begin{cases} \mathcal{T}_{n,g}(j) - \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(D_1) = \sigma_{n,g}(D_2) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(D_1) > \sigma_{n,g}(D_2) \text{ and } dif_d \geq \Delta \\ \mathcal{T}_{n,g}(j) - \frac{\Delta - dif_d}{2} & ,if \quad \sigma_{n,g}(D_1) > \sigma_{n,g}(D_2) \text{ and } dif_d < \Delta \\ \mathcal{T}_{n,g}(j) - \frac{dif_d + \Delta}{2} & ,if \quad \sigma_{n,g}(D_1) < \sigma_{n,g}(D_2) \end{cases} \quad (48)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=1}^{j=1,4} = \begin{cases} \mathcal{T}_{n,g}(j) - \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(D_1) = \sigma_{n,g}(D_2) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(D_1) < \sigma_{n,g}(D_2) \text{ and } dif_d \geq \Delta \\ \mathcal{T}_{n,g}(j) - \frac{\Delta - dif_d}{2} & ,if \quad \sigma_{n,g}(D_1) < \sigma_{n,g}(D_2) \text{ and } dif_d < \Delta \\ \mathcal{T}_{n,g}(j) - \frac{dif_d + \Delta}{2} & ,if \quad \sigma_{n,g}(D_1) > \sigma_{n,g}(D_2) \end{cases} \quad (49)$$

$$E[\mathcal{T}_{n,g}(j)]_{W_n=1}^{j=2,3} = \begin{cases} \mathcal{T}_{n,g}(j) + \frac{\Delta}{2} & ,if \quad \sigma_{n,g}(D_1) = \sigma_{n,g}(D_2) \\ \mathcal{T}_{n,g}(j) & ,if \quad \sigma_{n,g}(D_1) < \sigma_{n,g}(D_2) \text{ and } dif_d \geq \Delta \\ \mathcal{T}_{n,g}(j) + \frac{\Delta - dif_d}{2} & ,if \quad \sigma_{n,g}(D_1) < \sigma_{n,g}(D_2) \text{ and } dif_d < \Delta \\ \mathcal{T}_{n,g}(j) + \frac{dif_d + \Delta}{2} & ,if \quad \sigma_{n,g}(D_1) > \sigma_{n,g}(D_2) \end{cases} \quad (50)$$

在上述之係數能量差異化計算過程中，係數之正負位元同樣必須先行忽略，一律取其絕對值作差異化計算，俟計算完成後，再恢復其正負號，故運算過程中係數 $\mathcal{T}_{n,g}(j)$ 之能量經調整後最小值必須為0。

和前面兩種差異化方向的浮水印嵌入程序相同，若係數之能量在調整過程中之原始

修正值 $\leq -2\Delta$ ，則當該係數能量調整為 0 時，將會改變 $\sigma'_{n,g}(D_1)$ 及 $\sigma'_{n,g}(D_2)$ 之大小相對關係，進而影響到浮水印偵測之判斷，故我們須以下列檢查式作斜對角方向差異化運算之修正：

$$\left(\begin{array}{l} \mathcal{T}'_{n,g}(1) = \mathcal{T}'_{n,g}(1) - |\mathcal{T}'_{n,g}(4)| \quad ,if \quad \mathcal{T}'_{n,g}(4) \leq -2\Delta \quad and \quad W_n = 1 \\ \mathcal{T}'_{n,g}(4) = \mathcal{T}'_{n,g}(4) - |\mathcal{T}'_{n,g}(1)| \quad ,if \quad \mathcal{T}'_{n,g}(1) \leq -2\Delta \quad and \quad W_n = 1 \\ \mathcal{T}'_{n,g}(2) = \mathcal{T}'_{n,g}(2) - |\mathcal{T}'_{n,g}(3)| \quad ,if \quad \mathcal{T}'_{n,g}(3) \leq -2\Delta \quad and \quad W_n = -1 \\ \mathcal{T}'_{n,g}(3) = \mathcal{T}'_{n,g}(3) - |\mathcal{T}'_{n,g}(2)| \quad ,if \quad \mathcal{T}'_{n,g}(2) \leq -2\Delta \quad and \quad W_n = -1 \end{array} \right. \quad (51)$$

4.1.5.4 差異化方向的選擇

我們須根據欲嵌入的浮水印位元值與該組係數之差異特性，從中選擇一種既符合需求且係數能量在運算前、後變化較微幅的最適差異化方向，以進行浮水印的嵌入。



差異化方向選擇的最大原則，在於依浮水印位元值與該組係數之差異特性，使係數能量在某方向之差異化運算後，相對的係數平均值差距能夠達到我們所設定之差異化間距 Δ ，且係數能量之變化愈平緩愈佳，如此既能使浮水印擁有一定程度之強韌性，且影像品質也能儘量保持在較佳之狀態。

為了達到降低係數能量變化程度之目的，當 $W_n = -1$ 時，我們優先選擇 $\sigma_{n,g}(Ur)$ 、 $\sigma_{n,g}(Lt)$ 、 $\sigma_{n,g}(D_1)$ 較相對之另一半係數平均值為大，且差距最多之最適差異化方向；若 $W_n = 1$ ，則優先選擇 $\sigma_{n,g}(Ur)$ 、 $\sigma_{n,g}(Lt)$ 、 $\sigma_{n,g}(D_1)$ 較相對之另一半係數平均值為小，且差距最多之最適差異化方向。我們可由下列演算法以決定該組係數之差異化方向：

小波係數能量最適差異化方向選擇演算法

輸入參數：	$\sigma_{n,g}(Ur)$	上方係數平均值
	$\sigma_{n,g}(Lr)$	下方係數平均值
	$\sigma_{n,g}(Lt)$	左方係數平均值
	$\sigma_{n,g}(Rt)$	右方係數平均值
	$\sigma_{n,g}(D_1)$	左斜對角係數平均值
	$\sigma_{n,g}(D_2)$	右斜對角係數平均值
輸出參數：	direction	差異化方向

- 1 : $V = \sigma_{n,g}(Ur) - \sigma_{n,g}(Lr)$
- 2 : $H = \sigma_{n,g}(Lt) - \sigma_{n,g}(Rt)$
- 3 : $D = \sigma_{n,g}(D_1) - \sigma_{n,g}(D_2)$
- 4 : if ($W_n == -1$) direction=find_max (V , H , D)
- 5 : else if ($W_n == 1$) direction=find_min (V , H , D)



演算法 2

4.1.5.5 小波樹差異化方向之記錄

一個浮水印位元 W_n 嵌入一個小波樹 T_n 的程序，即為該浮水印位元 W_n 連續對該小波樹 T_n 的第二、三、四階層係數進行能量差異化的過程，當三種階層之小波係數皆對於 W_n 完成差異化運算，則方能進行下一個浮水印位元對下一個小波樹的嵌入動作，直到所有的浮水印都完成嵌入程序為止。

在第二、三階層進行係數能量差異化運算時，需配合差異化方向之選定，每棵小波樹的各組係數差異化方向，會因浮水印位元的值及係數與係數間之能量差異狀態而有所不同，故我們需要將每組係數的差異化方向記錄下來，以作為解出浮水印時使用。

我們使用一差異化方向記錄值 $R(\mathcal{T}_n)$ 來記錄單一小波樹第二、三階層之各組係數差異化方向，由於差異化方向具有垂直、水平與斜對角等三種選擇，我們使用 $d(\mathcal{T}_{n,g})$ 表示各組係數之差異化方向，若是垂直差異化，則令 $d(\mathcal{T}_{n,g})=0$ ；若為水平差異化，則 $d(\mathcal{T}_{n,g})=1$ ；若是斜對角差異化，則 $d(\mathcal{T}_{n,g})=2$ ，可得差異化方向記錄值 $R(\mathcal{T}_n)$ 之計算式如下：

$$R(\mathcal{T}_n) = \sum_{g=1}^5 d(\mathcal{T}_{n,g}) \times 3^{g-1} \quad \text{for } 0 \leq d(\mathcal{T}_{n,g}) \leq 2 \quad (52)$$



假設我們將一浮水印位元值 $W_1 = -1$ 嵌入小波樹 \mathcal{T}_1 之中，則表示真實浮水印位元值 W_1 會依(21)式重覆嵌入第二、三階層的第一、二、四組係數中，而相反之浮水印位元值 $-W_1$ 則會嵌入第一、三、五組係數中。我們將嵌入真實浮水印位元的第一、二、四組係數差異化方向，與嵌入相反浮水印位元的第三、五組係數差異化方向記錄至差異化方向記錄值 $R(\mathcal{T}_1)$ 之中，若此 5 組係數之差異化方向均為垂直方向，則表 $d(\mathcal{T}_{1,1})$ 至 $d(\mathcal{T}_{1,5})$ 均為 0，可得 $R(\mathcal{T}_1)=0$ ；若該 5 組係數之差異化方向均為斜對角方向，則表 $d(\mathcal{T}_{1,1})$ 至 $d(\mathcal{T}_{1,5})$ 均為 2，可得 $R(\mathcal{T}_1)=242$ ，因此可知 $R(\mathcal{T}_n)$ 之範圍為 $0 \leq R(\mathcal{T}_n) \leq 242$ 。

接著我們將計算出的差異化方向記錄值儲存至差異化方向記錄檔中，以作為將來解出浮水印時之係數差異化方向依據。當所有浮水印皆完成嵌入後，必須將 3072 棵小波

樹之排列順序恢復成初始狀態，再透過反小波轉換(inverse-DWT)將影像自頻率域轉換回空間域，成為一張已加入浮水印的影像，至此則完成了整個浮水印的嵌入程序。

4.2 以動態能量差異化為基礎之浮水印取出流程

對於一張不知是否含有浮水印之待測影像，我們必須透過浮水印取出流程，將可能之浮水印解析出來，再經由浮水印的驗證程序，以確認該浮水印是否存在。若浮水印經驗證後確實存在，則該浮水印所表示之資訊就可以作為該影像版權歸屬的依據。浮水印取出流程如圖 21 所示：

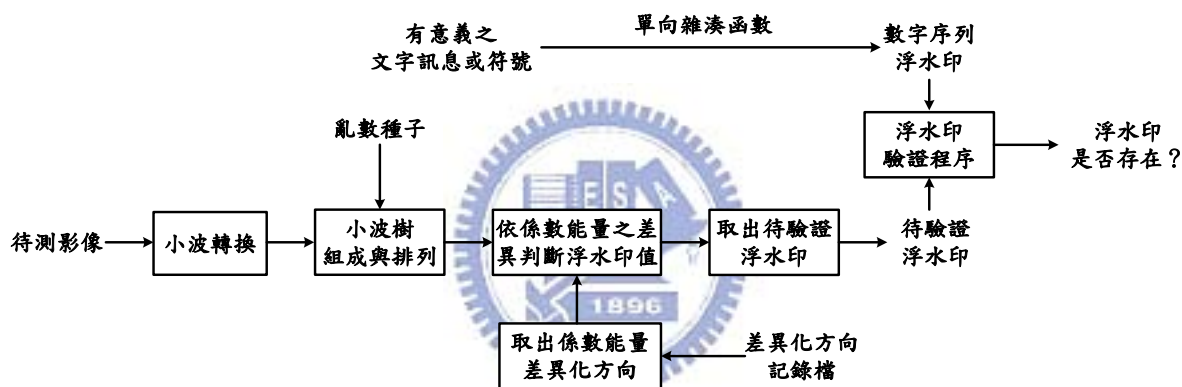


圖 21 以動態能量差異化為基礎之浮水印取出流程

在浮水印取出流程的初始步驟，我們同樣必須將待測影像經過四層的二維小波轉換，使影像自空間域轉換至頻率域，並分離出 13 個子頻帶。接著依四元樹之空間位置對應原則，將不同階層之小波係數組合為小波樹，共可產生 3072 棵小波樹。

之後我們使用與浮水印嵌入時相同的亂數種子，將 3072 棵小波樹依亂數結果重新排列，此排列順序將會與當初嵌入浮水印時的順序相同，當我們稍後依序將浮水印位元自小波樹中取出時，就得以確保取出之浮水印位元順序與當初嵌入時之順序一致。

在浮水印位元取出時的數值判斷程序中，因為第四階層與第二、三階層之浮水印嵌入方法不同，故我們同樣必須針對不同的浮水印取出流程分別作處理，俟得到不同階層所取出之浮水印位元值後，再依階層之不同而以權重關係來作通盤考量，以決定該小波樹所藏入之浮水印位元值。

4.2.1 第四階層小波係數之浮水印取出

因為小波樹 \mathcal{T}'_n 在第四階層之小波係數僅有 1 個，在浮水印嵌入時是採用單一係數能量差異化的方法，對係數值加以量化，故當我們欲取出浮水印時，必須使用當初作嵌入運算時所使用之量化參數，將其代入浮水印取出流程。

首先我們須將第四階層之係數能量值 $\mathcal{T}'_n(1)$ 與 modular 值 S 進行量化運算如下式：



$$\theta_n(1) = \mathcal{T}'_n(1) \bmod S \quad (53)$$

之後我們再以浮水印嵌入時所定義的兩個差異化門檻值 T_1 與 T_2 ，作為判斷浮水印位元值的依據，若以 $D[\cdot]$ 表示浮水印解出之運算程序，則第四階層浮水印位元取出的關係式如下：


$$D[\mathcal{T}'_n(1)] = \begin{cases} -1 & ,if \ |\theta_n(1)| < \frac{T_1 + T_2}{2} \\ +1 & ,if \ |\theta_n(1)| \geq \frac{T_1 + T_2}{2} \end{cases} \quad (54)$$

由(54)式我們可得到小波樹 \mathcal{T}'_n 第四階層係數所嵌入之浮水印位元值，然而此結果

尚需與第二、三階層係數所取出之浮水印位元值相互對應整合，才能最後確認小波樹 \mathcal{T}'_n 嵌入之浮水印位元值。

4.2.2 第二、三階層小波係數之浮水印取出

在浮水印嵌入流程中，我們將小波樹 \mathcal{T}'_n 之第二階層小波係數分為 1 組、第三階層係數則分為 4 組，且利用動態改變同組 4 個係數之能量差異化關係，達到浮水印位元值嵌入之目的。而在浮水印取出流程中，我們同樣是以相同方式將第二、三階層係數分為 5 組，藉由差異化方向記錄檔中，以小波樹為單位所儲存的差異化方向記錄值 $R(\mathcal{T}_n)$ ，將 $R(\mathcal{T}_n)$ 依(55)式以得知每棵小波樹第二、三階層各組係數在嵌入浮水印位元時的差異化方向 $d(\mathcal{T}_{n,g})$ 。 $d(\mathcal{T}_{n,g})$ 數值與其表示之差異化方向關係如下：



$$\left\{ \begin{array}{l} \text{差異化方向=垂直方向} \quad \text{if} \quad d(\mathcal{T}_{n,g})=0 \\ \text{差異化方向=水平方向} \quad \text{if} \quad d(\mathcal{T}_{n,g})=1 \\ \text{差異化方向=斜對角方向} \quad \text{if} \quad d(\mathcal{T}_{n,g})=2 \end{array} \right. \quad (55)$$

由各組係數之差異化方向與係數之能量差異結構比較結果，我們就可以得知該組係數所嵌入之浮水印位元數值為何，若以 $D[\cdot]$ 表示浮水印解出之運算程序，依不同的差異化方向作區分，其浮水印位元值之判斷式如下：

$$D[\mathcal{T}'_{n,g}] = \begin{cases} -1 & , \text{if } \sigma'_{n,g}(Ur) > \sigma'_{n,g}(Lr) \text{ and } d(\mathcal{T}_{n,g})=0 \\ 1 & , \text{if } \sigma'_{n,g}(Ur) < \sigma'_{n,g}(Lr) \text{ and } d(\mathcal{T}_{n,g})=0 \\ 0 & , \text{if } \sigma'_{n,g}(Ur) = \sigma'_{n,g}(Lr) \text{ and } d(\mathcal{T}_{n,g})=0 \end{cases} \quad (56)$$

$$D[\mathcal{T}'_{n,g}] = \begin{cases} -1 & ,if \ \sigma'_{n,g}(Lt) > \sigma'_{n,g}(Rt) \text{ and } d(\mathcal{T}_{n,g}) = 1 \\ 1 & ,if \ \sigma'_{n,g}(Lt) < \sigma'_{n,g}(Rt) \text{ and } d(\mathcal{T}_{n,g}) = 1 \\ 0 & ,if \ \sigma'_{n,g}(Lt) = \sigma'_{n,g}(Rt) \text{ and } d(\mathcal{T}_{n,g}) = 1 \end{cases} \quad (57)$$

$$D[\mathcal{T}'_{n,g}] = \begin{cases} -1 & ,if \ \sigma'_{n,g}(D_1) > \sigma'_{n,g}(D_2) \text{ and } d(\mathcal{T}_{n,g}) = 2 \\ 1 & ,if \ \sigma'_{n,g}(D_1) < \sigma'_{n,g}(D_2) \text{ and } d(\mathcal{T}_{n,g}) = 2 \\ 0 & ,if \ \sigma'_{n,g}(D_1) = \sigma'_{n,g}(D_2) \text{ and } d(\mathcal{T}_{n,g}) = 2 \end{cases} \quad (58)$$

在上式中，當得出之浮水印位元數值為 0 時，表示待測影像可能因遭受過攻擊，致使當初嵌入浮水印位元之該組係數能量差異化結構被破壞，使得差異化消失，因而無法確認浮水印位元之數值，故我們在作某小波樹取出之浮水印位元值考量時，將會捨棄該組係數之結果。

4.2.3 浮水印位元結果之考量分析

因為待測影像在嵌入浮水印之後，可能會遭受到各種影像訊號處理攻擊，當初嵌入某浮水印位元之係數能量差異化結構可能因此被破壞，故某些階層所偵測出之浮水印位元數值可能有誤，為了因應此種情形，我們在確認某棵小波樹所嵌入的浮水印位元值時，必須針對該小波樹內所有階層之各組係數所判斷出的浮水印位元結果作通盤考量，方能得以確保取出之浮水印的正確性。

一棵小波樹結構包含三種階層之小波係數，除第四階層係數會取出一種浮水印位元值結果外，第二、三階層係數亦會各取出一種及四種浮水印位元值結果，在沒有受到任何影像處理攻擊的情況下，這六種浮水印位元值結果應該是一致的，但若是待測影像遭到攻擊，則六種浮水印位元結果可能會不一致。

由人類視覺系統(HVS, Human Visual System)之相關研究可以得知，影像頻率域不同子頻帶之小波係數，對於人類視覺敏感度之影響亦有所差異 [30]。故在浮水印取出時，

我們必須考慮到不同子頻帶小波係數之相對重要性，進而設定從該子頻帶取出之浮水印位元結果的權重比例。[30]將小波係數相對於人類知覺重要性的對比敏感度函式(CSF, Contrast-Sensitive Function)曲線，轉換為五層小波分解後所產生的 11 種子頻帶係數之對比敏感度權重比例，如圖 22 所示：

<small>HL₄ 3.55</small>	<small>HL₄ 3.55</small>	HL₃	HL₂ 4.74	HL₁ 2.33
<small>LH₄ 3.55</small>	<small>HH₄ 3.48</small>	5.30		
LH₃ 5.30	HH₃ 7.20			
LH₂ 4.74		HH₂ 3.75		
LH₁ 2.33			HH₁ 1.00	

圖 22 五層小波分解之子頻帶係數對比敏感度權重比例

在圖 22 中，影像經五層小波分解後產生了 16 個子頻帶，在同一階層的子頻帶中，HL 及 LH 子頻帶的對比敏感度權重比例是相同的，故 16 個子頻帶共有 11 種不同的對比敏感度權重比例數值，我們以 α_1 至 α_{11} 表示此 11 種權重比例數值如表 1：

子頻帶	HH ₁	LH ₁ 、HL ₁	HH ₂	LH ₂ 、HL ₂	HH ₃
權重比例代號	α_1	α_2	α_3	α_4	α_5
權重比例數值	1.00	2.33	3.75	4.74	7.20

子頻帶	LH ₃ 、HL ₃	HH ₄	LH ₄ 、HL ₄	HH ₅	LH ₅ 、HL ₁	LL ₅
權重比例代號	α_6	α_7	α_8	α_9	α_{10}	α_{11}
權重比例數值	5.30	3.48	3.55	3.21	3.48	3.78

表 1 子頻帶之對比敏感度權重比例數值對應表

我們在對自不同子頻帶係數所取出之浮水印位元結果作考量時，即以此對比敏感度權重比例作為通盤考量時的依據，藉以提高浮水印位元值判斷的正確性。當小波樹為第二至四階層之 HL 或 LH 子頻帶係數所組成時，浮水印位元整體判斷之狀態值 ω_n 計算式如下：

$$\omega_n = \begin{cases} D[\mathcal{T}'_n(1)] \times \alpha_8 + D[\mathcal{T}'_{n,1}] \times \alpha_6 + (D[\mathcal{T}'_{n,2}] - D[\mathcal{T}'_{n,3}] + D[\mathcal{T}'_{n,4}] - D[\mathcal{T}'_{n,5}]) \times \alpha_4 & ,if \ n\%2=1 \\ D[\mathcal{T}'_n(1)] \times \alpha_8 + D[\mathcal{T}'_{n,1}] \times \alpha_6 + (-D[\mathcal{T}'_{n,2}] + D[\mathcal{T}'_{n,3}] - D[\mathcal{T}'_{n,4}] + D[\mathcal{T}'_{n,5}]) \times \alpha_4 & ,if \ n\%2=0 \end{cases} \quad (59)$$

若小波樹是由第二至四階層之 HH 子頻帶係數所組成時，則浮水印位元整體判斷之狀態值 ω_n 計算式如下：

$$\omega_n = \begin{cases} D[\mathcal{T}'_n(1)] \times \alpha_7 + D[\mathcal{T}'_{n,1}] \times \alpha_5 + (D[\mathcal{T}'_{n,2}] - D[\mathcal{T}'_{n,3}] + D[\mathcal{T}'_{n,4}] - D[\mathcal{T}'_{n,5}]) \times \alpha_3 & ,if \ n\%2=1 \\ D[\mathcal{T}'_n(1)] \times \alpha_7 + D[\mathcal{T}'_{n,1}] \times \alpha_5 + (-D[\mathcal{T}'_{n,2}] + D[\mathcal{T}'_{n,3}] - D[\mathcal{T}'_{n,4}] + D[\mathcal{T}'_{n,5}]) \times \alpha_3 & ,if \ n\%2=0 \end{cases} \quad (60)$$

在(59)、(60)式中， $D[\mathcal{T}'_n(1)]$ 是由第四階層係數所取出之浮水印位元結果， $D[\mathcal{T}'_{n,1}]$ 則是由第三階層係數所取出之浮水印位元結果，我們將這些結果分別乘上個別的權重比例。此外在第二階層係數嵌入浮水印時，我們刻意使用亂數種子所隨機排列出的小波樹順序，以選擇其中 2 組係數嵌入真的浮水印位元值，剩餘 2 組係數則嵌入相反的浮水印位元值，使攻擊者無法分析出真正的浮水印位元為何，故在取出浮水印位元時，我們針

對嵌入相反浮水印位元的 2 組係數，將所取出之浮水印位元結果取其相反值，以使該結果轉為正確之數值，故在(59)、(60)式中，我們依小波樹之順序指標 n 而對第二階層係數所取出之浮水印位元結果 $D[T'_{n,2}]$ 至 $D[T'_{n,5}]$ 作不同之處理。

由 ω_n 之值，我們再透過下列比較式，以最後確認此棵小波樹被嵌入的浮水印位元數值：

$$W'_n = \begin{cases} -1 & , \omega_n \geq 0 \\ 1 & , \omega_n < 0 \end{cases} \quad (61)$$

之後我們將所得到的浮水印位元數值記錄下來，並依序對接下來的小波樹進行以上之浮水印取出步驟，直到取出之浮水印長度符合當初嵌入的長度，至此則完成了整個自待測影像中取出浮水印的程序。



4.3 浮水印的驗證

在浮水印的驗證方面，我們依舊使用 3.3 節所述之常態相關性浮水印偵測法，將取出待驗證之浮水印 W' 與原先嵌入之浮水印 W 進行相關係數 ρ 的計算，我們再將 ρ 與設定為 0.23 之相關性門檻值 ρ_T 進行比較，以確認 W' 與 W 是否相同。

4.4 浮水印演算法之改進

在以動態能量差異化為基礎之浮水印演算法中，我們主要是藉由小波樹各階層係數之間的能量差異，達到將浮水印隱藏之目的，然而當影像遭受 Pixel Shifting、Rotation and Scaling 等空間域影像破壞攻擊，亦或 JPEG、SPIHT 等影像壓縮處理攻擊時，嵌入的浮水印可能會因係數能量遭到破壞而導致無法偵測，故為了提高浮水印的強韌性，我們必須尋求更好的演算法改進方式。

此外，在浮水印的嵌入程序中，係數能量差異化方向的選擇是以運算前、後變化較微幅的最適方向為主，且該方法必須將每一組係數的差異化方向記錄下來，以作為取出浮水印時之判定依據。雖然此法能夠提高嵌入浮水印後的影像品質，但由於差異化方向必須記錄於差異化方向記錄檔中，且此記錄檔必須妥善保存，否則一但損壞或遺失則可能導致浮水印無法取出，造成版權認定上的困擾，故該方法其實額化增加了浮水印在安全性方面的重大負擔。針對此問題，我們亦必須尋求不同的差異化方向決定方法，以取代差異化方向記錄檔的存在。

本節即針對 4.1 及 4.2 節所述之動態能量差異化數位影像浮水印演算法，分別提出在浮水印嵌入與取出程序中新增影像切割重組之前置處理方法，以及隨機強制差異化方向之演算法則，以解決上述之浮水印強韌性與安全性問題，進而使演算法整體效能有所提升。



4.4.1 影像切割重組之前置處理改進方法

透過影像切割重組之前置處理方法，我們可以將原先用以嵌入浮水印的數位影像，轉換為另一張具馬賽克效果的新影像，之後再以此新影像實施浮水印的嵌入。如此當攻擊者針對原始影像進行空間域或頻率域攻擊時，攻擊效果就會降低許多，以下將分別說明影像前置處理之演算法與改進後之浮水印嵌入與取出流程。

4.4.1.1 影像切割重組之前置處理演算法

為了增加浮水印演算法的安全性與強韌性，在影像進行小波轉換之前，我們可以使用一種影像混沌(chaos)排列演算法 [31]，將原始影像切割成為大小 4×4 且各自獨立的子影像(subimage)，如此可使一張 512×512 的影像產生出 16384 張子影像，我們再透過隨機亂數排列的方式，將所有的子影像重新組合成為一張新的影像，之後再以此張新影像替代原始影像，以進行後續小波轉換及浮水印之嵌入程序。混沌排列方式對影像進行切

割重組之示意圖如圖 23，實際影像進行切割重組之前後對照圖如圖 24。由於攻擊者無從得知新影像之重組順序，故難以取得分解後的小波係數值以進行分析攻擊，浮水印的安全性也因此而提高。

經隨機亂數排列後之子影像順序

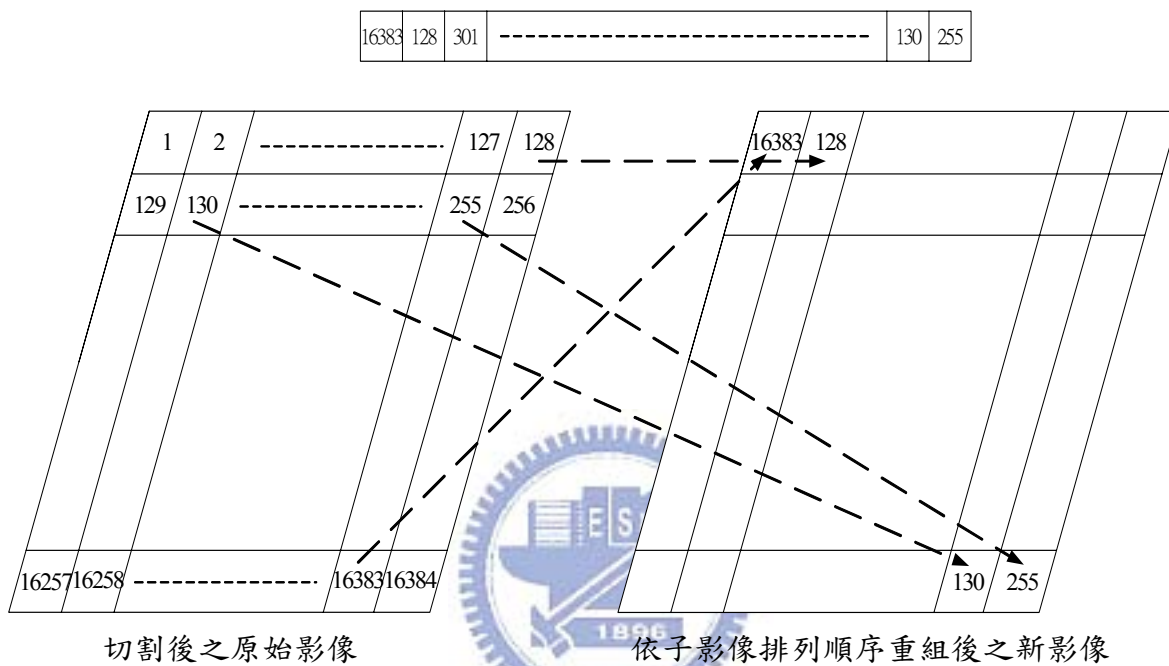


圖 23 以混沌排列方式對影像進行切割重組

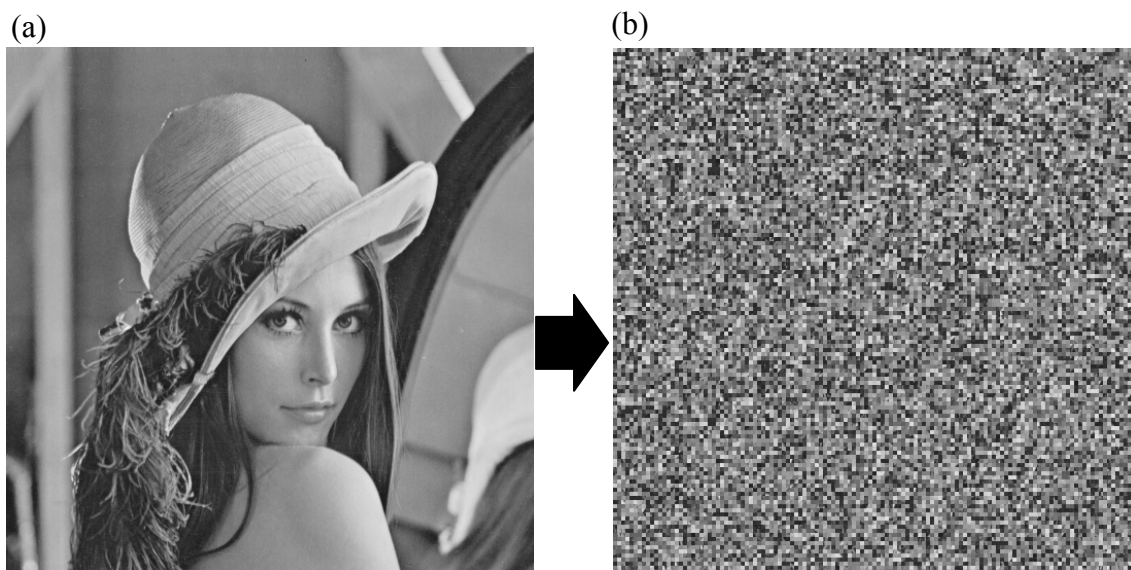


圖 24 以 Lena 影像進行切割重組之前後對照圖 (a)原始影像 (b)以 4×4 切割重組之新影像

4.4.1.2 加入影像前置處理後之浮水印嵌入與取出流程

我們在原本動態能量差異化浮水印方法的小波轉換步驟之前，加入影像切割重組之前置處理方法，以增強浮水印演算法之安全性與強韌性。改進後之浮水印嵌入流程如圖 25 所示：

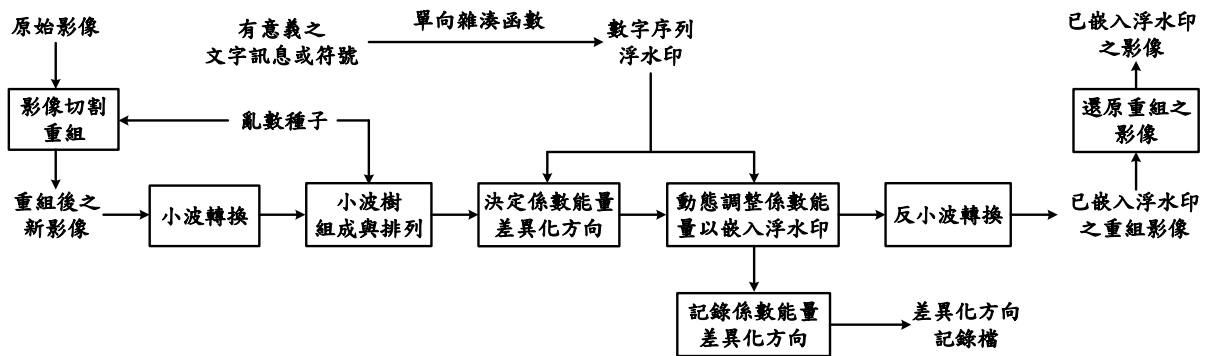


圖 25 加入影像前置處理後之浮水印嵌入流程

經過此影像切割重組之前置處理方法後，原始影像將轉換為具馬賽克效果的新影像，我們即以此張新影像進行後續小波轉換及浮水印之嵌入程序。在浮水印嵌入完成後，我們會再將內藏浮水印之新影像自頻率域轉換回空間域，並依原本組合之順序回復各個子影像之原本位置，使影像外觀還原至初始影像狀態，如此既不會因外觀改變而影響影像之商業價值，且當攻擊者欲改變影像局部區域(如 pixel shifting 攻擊)以期消除浮水印時，其實該局部區域在浮水印嵌入與取出流程中，會因為整張影像的切割重組而分散至新影像的非特定位置，對小波樹之係數結構影響也因此降低，故可有效提高浮水印對抗影像空間域幾何攻擊之能力。

同樣的，在浮水印取出流程中，待測影像必須先經過影像切割重組之前置處理，藉以轉換為與當初進行浮水印嵌入程序相同的新影像，之後再以此新影像進行小波轉換與小波樹不同階層之係數能量分析，以解析出隱藏於差異化能量中之浮水印位元。改進後之浮水印取出流程如圖 26 所示：

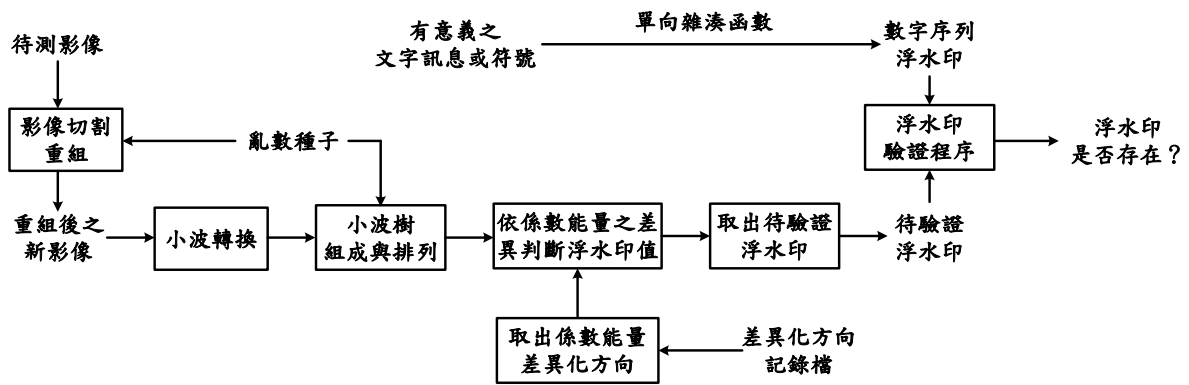


圖 26 加入影像前置處理後之浮水印取出流程

在浮水印取出流程的初始步驟，我們必須先將待測影像切割成為大小 4×4 且各自獨立的 16384 張子影像，並使用與浮水印嵌入時相同的亂數種子進行隨機亂數排列，以將這些子影像重新組合成新的待測影像。此時這張新待測影像之外觀，會與在浮水印嵌入流程中，原始影像經切割後再重組之影像相同。

隨後我們即以此張待測的新影像進行浮水印之取出程序，當浮水印自新影像之小波係數差異化能量比較中取出之後，尚須進行浮水印的驗證程序，以確認浮水印是否存在。加入影像切割重組之前置處理前、後之浮水印演算法效能提升狀況，將於第五章之實驗結果與討論中作進一步的說明與驗證。

4.4.2 隨機強制差異化方向改進方法

由於在原本的動態能量差異化浮水印演算法中，必須使用差異化方向記錄檔以記錄某浮水印嵌入時造成係數能量改變之差異化方向，且浮水印長度愈長，則該記錄檔內之資訊量會愈多。一旦嵌入的浮水印數量與長度增加時，差異化方向記錄檔的管理與保存就會成為一個嚴重的問題，倘若記錄檔損毀或遺失，則在浮水印取出程序中，將造成無法判定係數能量差異化方向的情形。

有鑑於此，本論文另外提出隨機強制差異化方向之改進方法，利用亂數排列後之小波樹順序，對於每一組係數之差異化方向作隨機且強制之分配。就係數組而言，此方法所決定之差異化方向雖然並不一定是最適宜的方向，且會因此造成影像品質的降低，但卻不需要額外記錄每組係數的差異化方向，在浮水印取出時亦不需要藉由差異化方向記錄檔以取得差異化方向的資訊，如此可有效增加浮水印演算法之安全性。

4.4.2.1 隨機強制差異化方向演算法

在動態能量差異化浮水印演算法中，當欲嵌入浮水印之影像的小波係數組成 3072 棵小波樹後，我們會使用一個指定的亂數種子，透過隨機排序涵式對小波樹進行隨機的排序，並依新的小波樹順序進行浮水印的嵌入，以增加嵌入時的安全性。

因為差異化方向含有垂直、水平與斜對角等三種方向，故本演算法即利用隨機亂數排列後新的小波樹順序，當作該小波樹中 5 組係數差異化方向分配時的亂數種子，並將產生之隨機亂數與 3 相除後取其餘數，若餘數為 0，則強制該組係數之差異化方向為垂直方向；餘數為 1，則為水平方向；餘數為 2，則是斜對角方向。因每棵小波樹之順序皆不同，故亂數種子亦不相同，且每一組係數之差異化方向是由隨機亂數所決定，不但增加了每棵小波樹係數差異化方向決定時的複雜度與安全性，對於惡意攻擊者而言，亦更加難以藉由係數分析方法進行攻擊。隨機強制差異化方向演算法如下所示：

隨機強制差異化方向演算法(對於某單一小波樹而言)

輸入參數： *order* 某小波樹經隨機排列後之新順序， $0 \leq order \leq 3071$

i 某小波樹中係數組之順序， $0 \leq i \leq 4$

輸出參數： *direction*[5] 某小波樹中 5 組係數之差異化方向

1 : *srand* (*order*)

2 : *for* (*i*=0 ; *i*< 5 ; *i*++)

3 : *direction*[*i*]= (*rand*() % 3)

演算法 3

由於在浮水印嵌入程序中，為了避免攻擊者透過分析第一組小波係數之能量差異狀態而得知浮水印 W_n 之值，我們必須重複將 W_n 及 $-W_n$ 嵌入第一組係數之中。原本的浮水印演算法在嵌入時之差異化方向會因 W_n 之正負值而必不相同，差異化之能量結構也會同時存在，但隨機強制差異化方向演算法則僅會產生一種差異化方向，為了解決這個問題，我們須依(62)式補充演算法 3 之內容，以決定嵌入第一組係數之兩種差異化方向與順序如下：

$$direction = \begin{cases} (rand()+1)\%3 & \text{for } -W_n \\ rand()\%3 & \text{for } W_n \end{cases} \quad (62)$$

在將浮水印嵌入第一組係數時，必須先將 $-W_n$ 依(62)式所得之差異化方向作嵌入，之後再將 W_n 依其差異化方向作嵌入，因為兩者之差異化方向不同，且 W_n 嵌入之方向是依循演算法 3 所產生，故在浮水印取出時則不須再作修正。至於第二組至第五組係數的浮水印嵌入方式，則與原先之浮水印方法相同，故不需要變更。

4.4.2.2 隨機強制差異化方向之浮水印嵌入與取出流程

我們以隨機強制差異化方向演算法取代原本的差異化方向選擇方法後，即不再需要使用差異化方向記錄檔以記錄每組係數的差異化方向，如此可降低檔案保存與維護的風險，並提升浮水印演算法的安全性。改進差異化方向演算法後之浮水印嵌入流程如圖 27 所示：

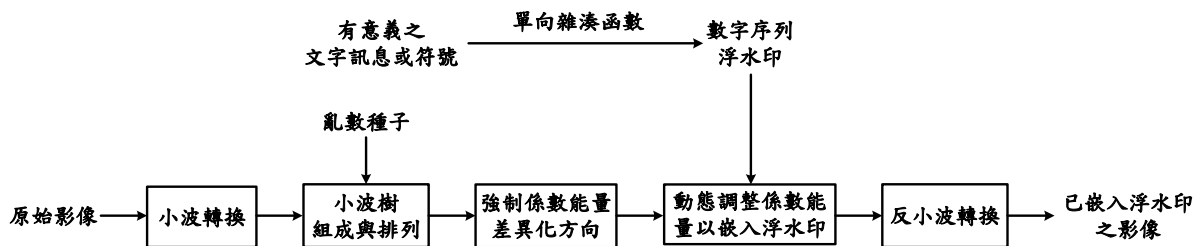


圖 27 改進差異化方向演算法後之浮水印嵌入流程

在改進差異化方向演算法後之浮水印嵌入流程中，除了差異化方向的選擇步驟不同之外，其餘部份皆未改變。因為隨機強制差異化方向演算法是以亂數排列後之小波樹順序作為隨機亂數種子，故差異化方向其實是受到小波樹排列時使用之亂數種子控制，即不同的小波樹排列亂數種子，會造成不同的小波樹排列順序，進而產生各組係數不同的差異化方向。至此我們成功的將隨機強制差異化方向的決定關鍵轉移至亂數種子之上。

至於在浮水印取出流程中，同樣是僅改變係數組差異化方向的取得步驟。透過相同的隨機強制差異化方向演算法，我們一樣使用亂數種子將小波樹重新排列，並以排列後之新順序，作為決定差異化方向之隨機亂數種子，進而取得各組係數在取出浮水印時的差異化方向。因為浮水印嵌入與取出時，必須使用相同的亂數種子以重新排列小波樹，故排列出的小波樹順序自然也相同，因小波樹順序即為決定係數組差異化方向之隨機亂數種子，當然嵌入與取出的係數組差異化方向也必然會相同。改進差異化方向演算法後之浮水印取出流程如圖 28 所示：

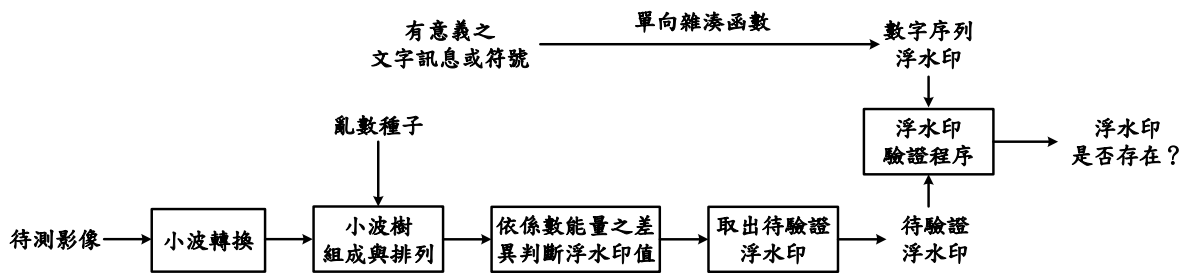


圖 28 改進差異化方向演算法後之浮水印取出流程

由於隨機強制差異化方向演算法在決定差異化方向時，並沒有考慮到係數間的差異特性，故可能會造成係數值大幅度的增減，進而降低嵌入浮水印後的影像品質。為了將影像品質維持在人類視覺可接受的程度之上，我們可採取將係數的差異化間距縮小之方法，然而此舉卻也會造成浮水印強韌性的降低。乍看之下，本方法所呈現出的浮水印效能並不如原先之最適差異化方向選擇演算法，但因為本方法不需額外記錄每組係數的差異化方向，在安全性的提升方面，確實具有相當高的價值。隨機強制差異化方向之浮水印演算法效能狀況與比較，亦將於第五章之實驗結果與討論中作進一步的說明與驗證。

4.5 浮水印嵌入與取出演算法

本節將本論文所提出之浮水印演算法以及改進後之演算法作完整的條列，如此可更清楚的瞭解不同浮水印方法的運作方式，並可輕易比較出相互之間的演算法差異。

4.5.1 以動態能量差異化為基礎之浮水印演算法

4.5.1.1 浮水印嵌入演算法

Step 1：配合亂數種子的選擇，依演算法 1 以隨機方式產生一連串值為 ± 1 的 PN 序列整數 W ，並以此數字序列作為欲嵌入影像中之浮水印，令 W_n 表示第 n 個浮水印位元值，浮水印長度為 N_w 。

Step 2：將原始影像經四層二維小波轉換至頻率域，並依四元樹空間位置對應原則，將第二階層至第四階層之 21 個小波係數結合為小波樹，如此可得 3072 棵小波樹。

Step 3：以亂數種子將小波樹順序重新排列，排列後之小波樹為 $\mathcal{T}_n, n=1, \dots, 3072$ ，令 $n=1$ 。

Step 4：以 $\mathcal{T}_n(k)$ 表示為小波樹 \mathcal{T}_n 之係數值， $k=1, \dots, 21$ 。

Step 5：設定 modular 值 S ，並依單一係數能量差異化方式，對第四階層係數值 $\mathcal{T}_n(1)$ 依 (16)、(17)、(18) 式加以量化，以得到量化後之第四階層係數值 $\mathcal{T}_n'(1)$ 。

Step 6：將 $\mathcal{T}_n(2)$ 至 $\mathcal{T}_n(21)$ 之係數值，取相鄰 2×2 個係數組成一組，並以 $\mathcal{T}_{n,g}(j)$ 表示第 g 組係數中第 j 個係數值， $g=1, \dots, 5, j=1, \dots, 4$ ，令 $g=1$ 。

Step 7：依(21)式決定係數組 $\mathcal{T}_{n,g}$ 嵌入之浮水印位元值，並由演算法 2 選擇該組係數之最適差異化方向，再依(22)至(51)式進行係數能量差異化運算。

Step 8：若 $g < 5$ ，則令 $g=g+1$ ，並回到 Step 7。

Step 9：將係數組 $\mathcal{T}_{n,1}$ 至 $\mathcal{T}_{n,5}$ 之差異化方向，依(52)式進行運算後，可得小波樹 \mathcal{T}_n 之差異化方向記錄值 $R(\mathcal{T}_n)$ ，並將 $R(\mathcal{T}_n)$ 儲存至差異化方向記錄檔中。

Step 10：若 $n < N_w$ ，則令 $n=n+1$ ，並回到 Step 4。

Step 11：將 3072 棵小波樹之排列順序恢復成初始狀態，再經由反小波轉換將影像自頻率域轉換回空間域，成為一張已嵌入浮水印的影像。

4.5.1.2 浮水印取出演算法

Step 1：同浮水印嵌入演算法中的 Step 1，取得原浮水印數字序列 W ，浮水印長度為 N_w 。

Step 2：將待測影像經四層二維小波轉換至頻率域，並依四元樹空間位置對應原則，將第二階層至第四階層之 21 個小波係數結合為小波樹，如此可得 3072 棵小波樹。

Step 3：以與浮水印嵌入演算法中相同之亂數種子，將小波樹順序重新排列，排列後之小波樹為 \mathcal{T}'_n ， $n=1,\dots,3072$ ，令 $n=1$ 。

Step 4：以 $\mathcal{T}'_n(k)$ 表示為小波樹 \mathcal{T}'_n 之係數值， $k=1,\dots,21$ 。

Step 5：將第四階層係數值 $\mathcal{T}'_n(1)$ 與 modular 值 S 依(53)、(54)式進行運算，可得到 $\mathcal{T}'_n(1)$ 所嵌入之浮水印位元值 $D[\mathcal{T}'_n(1)]$ 。

Step 6：將 $\mathcal{T}'_n(2)$ 至 $\mathcal{T}'_n(21)$ 之係數值，取相鄰 2×2 個係數組成一組，以 $\mathcal{T}'_{n,g}$ 表示第 g 組係數值， $g=1,\dots,5$ ， $j=1,\dots,4$ ，令 $g=1$ 。

Step 7：由差異化方向記錄檔中，取得差異化方向記錄值 $R(\mathcal{T}'_n)$ ，經由(55)式以得知 $\mathcal{T}'_{n,g}$ 在嵌入浮水印位元時的差異化方向 $d(\mathcal{T}'_{n,g})$ ，並依(56)至(58)式進行浮水印解出程序，得到浮水印位元值 $D[\mathcal{T}'_{n,g}]$ 。

Step 8：若 $g < 5$ ，則令 $g=g+1$ ，並回到 Step 7。

Step 9：若 \mathcal{T}'_n 為 HL 或 LH 子頻帶係數所組成，則依(59)式進行浮水印位元判斷狀態值 ω_n 之計算；若 \mathcal{T}'_n 為 HH 子頻帶係數所組成，則依(60)式進行浮水印位元判斷狀態值 ω_n 之計算。

Step 10：若 $\omega_n \geq 0$ ，則 $W'_n = -1$ ；若 $\omega_n < 0$ ，則 $W'_n = 1$ 。

Step 11：若 $n < N_w$ ，則令 $n=n+1$ ，並回到 Step 4。

Step 12：將取出之浮水印 W' 與原浮水印 W 依(12)式進行相關係數 ρ 的計算。

Step 13：將 ρ 與設定之相關性門檻值 ρ_T 進行比較，若 $\rho \geq \rho_T$ ，則表示浮水印確實存在於此待測影像中，若 $\rho < \rho_T$ ，則表示浮水印不存在於待測影像中。

4.5.2 影像切割重組之前置處理浮水印演算法

4.5.2.1 浮水印嵌入演算法

- Step 1：配合亂數種子的選擇，依演算法 1 以隨機方式產生一連串值為 ± 1 的 PN 序列整數 W ，並以此數字序列作為欲嵌入影像中之浮水印，令 W_n 表示第 n 個浮水印位元值，浮水印長度為 N_w 。
- Step 2：將原始影像切割成為大小 4×4 且各自獨立的子影像，並經由亂數種子將所有子影像重組為一張新影像。
- Step 3：將新影像經四層二維小波轉換至頻率域，並依四元樹空間位置對應原則，將第二階層至第四階層之 21 個小波係數結合為小波樹，如此可得 3072 棵小波樹。
- Step 4：以亂數種子將小波樹順序重新排列，排列後之小波樹為 $\mathcal{T}_n, n=1, \dots, 3072$ ，令 $n=1$ 。
- Step 5：以 $\mathcal{T}_n(k)$ 表示為小波樹 \mathcal{T}_n 之係數值， $k=1, \dots, 21$ 。
- Step 6：設定 modular 值 S ，並依單一係數能量差異化方式，對第四階層係數值 $\mathcal{T}_n(1)$ 依 (16)、(17)、(18) 式加以量化，以得到量化後之第四階層係數值 $\mathcal{T}_n'(1)$ 。
- Step 7：將 $\mathcal{T}_n(2)$ 至 $\mathcal{T}_n(21)$ 之係數值，取相鄰 2×2 個係數組成一組，並以 $\mathcal{T}_{n,g}(j)$ 表示第 g 組係數中第 j 個係數值， $g=1, \dots, 5, j=1, \dots, 4$ ，令 $g=1$ 。
- Step 8：依(21)式決定係數組 $\mathcal{T}_{n,g}$ 嵌入之浮水印位元值，並由演算法 2 選擇該組係數之最適差異化方向，再依(22)至(51)式進行係數能量差異化運算。
- Step 9：若 $g < 5$ ，則令 $g=g+1$ ，並回到 Step 8。
- Step 10：將係數組 $\mathcal{T}_{n,1}$ 至 $\mathcal{T}_{n,5}$ 之差異化方向，依(52)式進行運算後，可得小波樹 \mathcal{T}_n 之差異化方向記錄值 $R(\mathcal{T}_n)$ ，並將 $R(\mathcal{T}_n)$ 儲存至差異化方向記錄檔中。
- Step 11：若 $n < N_w$ ，則令 $n=n+1$ ，並回到 Step 5。
- Step 12：將 3072 棵小波樹之排列順序恢復成初始狀態，再經由反小波轉換將影像自頻率域轉換回空間域，此時之影像為已嵌入浮水印的新影像。
- Step 13：依原本組合之順序回復各個子影像之原本位置，使影像外觀還原至初始影像狀態，成為一張已嵌入浮水印的影像。

4.5.2.2 浮水印取出演算法

Step 1：同浮水印嵌入演算法中的 Step 1，取得原浮水印數字序列 W ，浮水印長度為 N_w 。

Step 2：將待測影像切割成為大小 4×4 且各自獨立的子影像，並經由與浮水印嵌入演算法中相同之亂數種子將所有子影像重組為一張新影像。

Step 3：將新影像經四層二維小波轉換至頻率域，並依四元樹空間位置對應原則，將第二階層至第四階層之 21 個小波係數結合為小波樹，如此可得 3072 棵小波樹。

Step 4：以與浮水印嵌入演算法中相同之亂數種子，將小波樹順序重新排列，排列後之小波樹為 \mathcal{T}'_n ， $n=1, \dots, 3072$ ，令 $n=1$ 。

Step 5：以 $\mathcal{T}'_n(k)$ 表示為小波樹 \mathcal{T}'_n 之係數值， $k=1, \dots, 21$ 。

Step 6：將第四階層係數值 $\mathcal{T}'_n(1)$ 與 modular 值 S 依(53)、(54)式進行運算，可得到 $\mathcal{T}'_n(1)$ 所嵌入之浮水印位元值 $D[\mathcal{T}'_n(1)]$ 。

Step 7：將 $\mathcal{T}'_n(2)$ 至 $\mathcal{T}'_n(21)$ 之係數值，取相鄰 2×2 個係數組成一組，以 $\mathcal{T}'_{n,g}$ 表示第 g 組係數值， $g=1, \dots, 5$ ， $j=1, \dots, 4$ ，令 $g=1$ 。

Step 8：由差異化方向記錄檔中，取得差異化方向記錄值 $R(\mathcal{T}'_n)$ ，經由(55)式以得知 $\mathcal{T}'_{n,g}$ 在嵌入浮水印位元時的差異化方向 $d(\mathcal{T}'_{n,g})$ ，並依(56)至(58)式進行浮水印解出程序，得到浮水印位元值 $D[\mathcal{T}'_{n,g}]$ 。

Step 9：若 $g < 5$ ，則令 $g=g+1$ ，並回到 Step 8。

Step 10：若 \mathcal{T}'_n 為 HL 或 LH 子頻帶係數所組成，則依(59)式進行浮水印位元判斷狀態值 ω_n 之計算；若 \mathcal{T}'_n 為 HH 子頻帶係數所組成，則依(60)式進行浮水印位元判斷狀態值 ω_n 之計算。

Step 11：若 $\omega_n \geq 0$ ，則 $W'_n = -1$ ； $\omega_n < 0$ ，則 $W'_n = 1$

Step 12：若 $n < N_w$ ，則令 $n=n+1$ ，並回到 Step 5。

Step 13：將取出之浮水印 W' 與原浮水印 W 依(12)式進行相關係數 ρ 的計算。

Step 14：將 ρ 與設定之相關性門檻值 ρ_T 進行比較，若 $\rho \geq \rho_T$ ，則表示浮水印確實存在於此待測影像中，若 $\rho < \rho_T$ ，則表示浮水印不存在於待測影像中。

4.5.3 隨機強制差異化方向之浮水印演算法

4.5.3.1 浮水印嵌入演算法

Step 1：配合亂數種子的選擇，依演算法 1 以隨機方式產生一連串值為 ± 1 的 PN 序列整數 W ，並以此數字序列作為欲嵌入影像中之浮水印，令 W_n 表示第 n 個浮水印位元值，浮水印長度為 N_w 。

Step 2：將原始影像經四層二維小波轉換至頻率域，並依四元樹空間位置對應原則，將第二階層至第四階層之 21 個小波係數結合為小波樹，如此可得 3072 棵小波樹。

Step 3：以亂數種子將小波樹順序重新排列，排列後之小波樹為 $\mathcal{T}_n, n=1, \dots, 3072$ ，令 $n=1$ 。

Step 4：以 $\mathcal{T}_n(k)$ 表示為小波樹 \mathcal{T}_n 之係數值， $k=1, \dots, 21$ 。

Step 5：設定 modular 值 S ，並依單一係數能量差異化方式，對第四階層係數值 $\mathcal{T}_n(1)$ 依

(16)、(17)、(18)式加以量化，以得到量化後之第四階層係數值 $\mathcal{T}_n'(1)$ 。

Step 6：將 $\mathcal{T}_n(2)$ 至 $\mathcal{T}_n(21)$ 之係數值，取相鄰 2×2 個係數組成一組，並以 $\mathcal{T}_{n,g}(j)$ 表示第 g 組係數中第 j 個係數值， $g=1, \dots, 5, j=1, \dots, 4$ ，令 $g=1$ 。

Step 7：依(21)式決定係數組 $\mathcal{T}_{n,g}$ 嵌入之浮水印位元值，並由演算法 3 決定該組係數之隨機強制差異化方向，若 $g=1$ ，則須另依(62)式以決定嵌入 $\mathcal{T}_{n,g}$ 之兩種差異化方向與順序，之後再依(22)至(51)式進行係數能量差異化運算。

Step 8：若 $g < 5$ ，則令 $g=g+1$ ，並回到 Step 7。

Step 9：若 $n < N_w$ ，則令 $n=n+1$ ，並回到 Step 4。

Step 10：將 3072 棵小波樹之排列順序恢復成初始狀態，再經由反小波轉換將影像自頻率域轉換回空間域，成為一張已嵌入浮水印的影像。

4.5.3.2 浮水印取出演算法

Step 1：同浮水印嵌入演算法中的 Step 1，取得原浮水印數字序列 W ，浮水印長度為 N_w 。

Step 2：將待測影像經四層二維小波轉換至頻率域，並依四元樹空間位置對應原則，將第二階層至第四階層之 21 個小波係數結合為小波樹，如此可得 3072 棵小波樹。

Step 3：以與浮水印嵌入演算法中相同之亂數種子，將小波樹順序重新排列，排列後之小波樹為 \mathcal{T}'_n ， $n=1,\dots,3072$ ，令 $n=1$ 。

Step 4：以 $\mathcal{T}'_n(k)$ 表示為小波樹 \mathcal{T}'_n 之係數值， $k=1,\dots,21$ 。

Step 5：將第四階層係數值 $\mathcal{T}'_n(1)$ 與 modular 值 S 依(53)、(54)式進行運算，可得到 $\mathcal{T}'_n(1)$ 所嵌入之浮水印位元值 $D[\mathcal{T}'_n(1)]$ 。

Step 6：將 $\mathcal{T}'_n(2)$ 至 $\mathcal{T}'_n(21)$ 之係數值，取相鄰 2×2 個係數組成一組，以 $\mathcal{T}'_{n,g}$ 表示第 g 組係數值， $g=1,\dots,5$ ， $j=1,\dots,4$ ，令 $g=1$ 。

Step 7：由演算法 3 決定 $\mathcal{T}'_{n,g}$ 之隨機強制差異化方向，並依(56)至(58)式進行浮水印解出程序，得到浮水印位元值 $D[\mathcal{T}'_{n,g}]$ 。

Step 8：若 $g < 5$ ，則令 $g=g+1$ ，並回到 Step 7。

Step 9：若 \mathcal{T}'_n 為 HL 或 LH 子頻帶係數所組成，則依(59)式進行浮水印位元判斷狀態值 ω_n 之計算；若 \mathcal{T}'_n 為 HH 子頻帶係數所組成，則依(60)式進行浮水印位元判斷狀態值 ω_n 之計算。

Step 10：若 $\omega_n \geq 0$ ，則 $W'_n = -1$ ；若 $\omega_n < 0$ ，則 $W'_n = 1$ 。

Step 11：若 $n < N_w$ ，則令 $n=n+1$ ，並回到 Step 4。

Step 12：將取出之浮水印 W' 與原浮水印 W 依(12)式進行相關係數 ρ 的計算。

Step 13：將 ρ 與設定之相關性門檻值 ρ_T 進行比較，若 $\rho \geq \rho_T$ ，則表示浮水印確實存在於此待測影像中，若 $\rho < \rho_T$ ，則表示浮水印不存在於待測影像中。

第五章 實驗結果與討論

本章將針對第四章所述之數位影像浮水印演算法作驗證，藉由實驗數據的比較，證明本論文所提出之演算法能夠在浮水印的透明、安全、多重與強韌等特性方面提供更佳的保障，並驗證影像切割重組之前置處理對於演算法效能的提升。在各項實驗的進行中，我們會先將浮水印嵌入數位影像中，再對影像進行各種攻擊，最後則由該影像中將浮水印取出，從浮水印的偵測結果可以證明本論文所提出之方法確實能夠有效抵抗影像處理之攻擊，並且能夠成功偵測出嵌入之浮水印。



(a) Lena



(b) Goldhill



(c) Peppers

圖 29 實驗所使用之三張數位影像

本章實驗所用的浮水印演算法共分為三種，分別為 4.1、4.2 節所述之動態能量差異化浮水印演算法，以及 4.4 節所述之兩種改進後的演算法，即原本的方法加上影像切割重組之前置處理，與隨機強制差異化方向之演算法則。此外在進行影像四層多樣解析度二維小波轉換時，我們分別使用了 9-7 與 18-10 兩種不同的 filter，藉以觀察在使用不同 filter 的情況下，演算法改進前、後之效能可能造成的差異。故本論文所產生的實驗結果共有五組，並與 [1] 之實驗結果作比較。

實驗中所使用的數位影像分別是在影像研究領域中知名的 Lena、Goldhill 及 Peppers 等三張灰階影像，與參考文獻 [1] 所使用之實驗影像相同，如圖 29，每張影像的大小均為 512×512 像素。在浮水印的選擇上，我們將浮水印設定為長度 512 的 ±1 數字序列，等同於參考文獻 [1] 之浮水印設定標準，以方便進行結果的比較。

在浮水印嵌入程序的參數設定方面，本論文藉由動態調整 modular 值與差異化間距，使 Lena、Goldhill 及 Peppers 三張影像在嵌入浮水印之後，影像品質維持在略高於 38.2 dB、38.7 dB 及 39.8 dB，以配合參考文獻 [1] 之設定，如此將可使本論文之實驗結果與 [1] 比較時，能夠具有較合理之比較基礎，且影像之品質亦能維持在一定的程度之上。在隨機亂數種子的使用上，浮水印產生、小波樹的排序以及在改進後的演算法中，影像切割重組前置處理之子影像重組順序，均使用相同的亂數種子。

浮水印解出程序的參數設定均與嵌入程序相同，解出之浮水印存在偵測計算則是使用 Normalized Correlation 的方法，門檻值 ρ_T 設為 0.23，如此將在 False positive probability 為 1.03×10^{-7} 之情況下進行原浮水印與待測浮水印之比較。

在實驗所使用的影像攻擊方法中，我們選擇了包含影像壓縮、空間域影像處理、頻率域小波係數位元平面移除、多重浮水印嵌入等攻擊，實驗的測試內容可歸納於表 2：

測試種類	測試實驗
抗壓縮能力	JPEG、SPIHT
空間域各種破壞	Median Filter、Gaussian Filter and Sharpening、Pixel Shifting、Rotation and Scaling
頻率域係數位元平面移除	Bitplane Removal
多重浮水印嵌入	Multiple Watermarking

表 2 浮水印抵抗影像處理攻擊之實驗列表

本章所進行的五組實驗結果，其代號表示方式、演算法差異及使用之 filter 種類如表 3 所示：

實驗結果代號	演算法差異	Filter 種類
D18-10	動態能量差異化浮水印演算法	18-10 filter
D18-10(M)	加入影像切割重組前置處理之改進後演算法	18-10 filter
D9-7	動態能量差異化浮水印演算法	9-7 filter
D9-7(M)	加入影像切割重組前置處理之改進後演算法	9-7 filter
D9-7-F	隨機強制差異化方向之浮水印演算法	9-7 filter

表 3 實驗結果代號與演算法差異一覽表

此外，為了測試浮水印方法在改進前、後的運算執行效能比較，本論文以上述五組實驗設定方法，對於 Lena、Goldhill 及 Peppers 三張影像分別進行 100 次浮水印嵌入與取出之運算，並計算出五組實驗之浮水印演算法執行所需之平均時間。測試時所使用之機器為一般的個人電腦，其軟、硬體規格如下：

- 作業系統：Microsoft Windows XP Professional SP2
- 處理器：Intel Pentium 4，3.00GHz CPU
- 記憶體：1.0 GB DDR II SDRAM

經過重覆 100 次的執行測試後，可得五組實驗之浮水印演算法平均運算時間比較表如下：

(a)使用 18-10 filter

單位：秒

測試影像	Lena		Goldhill		Peppers		平均	
	嵌入	取出	嵌入	取出	嵌入	取出	嵌入	取出
D18-10	1.168	1.112	1.166	1.118	1.176	1.113	1.170	1.114
D18-10(M)	1.985	1.930	1.988	1.932	1.975	1.925	1.983	1.929

(b)使用 9-7 filter

單位：秒

測試影像	Lena		Goldhill		Peppers		平均	
	嵌入	取出	嵌入	取出	嵌入	取出	嵌入	取出
D9-7	0.864	0.802	0.864	0.823	0.864	0.803	0.864	0.809
D9-7(M)	1.683	1.625	1.686	1.626	1.682	1.622	1.684	1.624
D9-7-F	0.849	0.796	0.849	0.795	0.848	0.795	0.849	0.795

表 4 五組實驗之浮水印演算法運算時間比較表 (a)使用 18-10 filter (b)使用 9-7 filter

由表 4 之實驗數據可以得知，當使用 18-10 filter 時，加入影像切割重組之前置處理後，浮水印嵌入之運算時間增加了 69%，取出之運算時間則增加了 73%；若使用 9-7 filter 並加入影像前置處理後，浮水印嵌入之運算時間增加了 95%，取出之運算時間則增加了 100%。雖然影像切割重組之前置處理程序確實會降低浮水印運算上的效率，但由於實際上浮水印嵌入與取出之運算時間最大皆不超過 2 秒鐘，故並不會造成很大的負擔，且

均在一般可接受的範圍內。

至於隨機強制差異化方向之浮水印演算法，其浮水印嵌入與取出之運算執行時間均為原本方法的 98%，運算時間較原本的浮水印方法稍短，其原因是在差異化方向的決定步驟中，不再需要進行係數值能量差異的比較，而直接使用亂數決定，故運算效率自然有所提升。



5.1 JPEG 壓縮攻擊

JPEG(Joint Photographic Experts Group)原本僅是一個影像研究組織名稱之縮寫，該組織起源於 1986 年，成員大部份為國際標準組織(ISO，International Organization for Standardization)與國際電報電話諮詢委員會(CCITT，International Telegraph and Telephone Consultative Committee)兩大組織的會員 [32]。自 1991 年 3 月 JPEG 技術首次發表至今，JPEG 已經成為一個國際標準的數位影像壓縮方法。

JPEG 是一種會失真的破壞性壓縮法，適用於靜態的全彩與灰階影像之壓縮，其主要是以離散餘弦變換 (DCT，Discrete Cosine Transform) 作為核心演算法，將影像作區塊切割並轉換至頻率域後，經過量化程序去除高頻區較不重要的資訊，保留低頻區較重要之資訊，以達到影像壓縮的目的。我們可以藉由控制壓縮係數(Quality factor of compression)以取得影像壓縮率與壓縮後影像品質間的平衡。



(a) Lena 原始影像

(b) 以 JPEG 壓縮過後之影像



壓縮前的原始影像

以 JPEG 對影像進行壓縮攻擊之結果
壓縮係數=40，PSNR=33.49， $\rho=1$

圖 30 Lena 原始影像與以 JPEG 對嵌入浮水印的影像進行壓縮攻擊後之比較

在圖 30 中，我們將 Lena 影像經「D18-10」之方法嵌入浮水印，再以 JPEG 壓縮方法進行攻擊。由實驗結果可知，當影像經 JPEG 壓縮攻擊之後，影像之容量減少，且影像品質也因此降低(PSNR=33.49)，然而本論文所提出之動態能量差異化浮水印技術，仍能夠由影像中偵測出浮水印的存在($\rho=1$)，可證明本浮水印技術確實具有一定的強韌性。

實驗結果：

(a) Lena

JPEG		10	30	40	50	70	90
Results from D18-10	ρ	0.86	1	1	1	1	1
Results from D18-10(M)	ρ	0.95	1	1	1	1	1
Results from D9-7	ρ	0.70	1	1	1	1	1
Results from D9-7(M)	ρ	0.92	1	1	1	1	1
Results from D9-7-F	ρ	0.42	1	1	1	1	1
Results from [1]	ρ		0.15	0.23	0.26	0.57	1

(b) Goldhill

JPEG		10	30	40	50	70	90
Results from D18-10	ρ	0.88	1	1	1	1	1
Results from D18-10(M)	ρ	0.93	1	1	1	1	1

Results from D9-7	ρ	0.81	1	1	1	1	1
Results from D9-7(M)	ρ	0.96	1	1	1	1	1
Results from D9-7-F	ρ	0.52	1	1	1	1	1
Results from [1]	ρ		0.15	0.23	0.26	0.57	1

(c) Peppers

JPEG		10	30	40	50	70	90
Results from D18-10	ρ	0.75	1	1	1	1	1
Results from D18-10(M)	ρ	0.93	1	1	1	1	1
Results from D9-7	ρ	0.63	1	1	1	1	1
Results from D9-7(M)	ρ	0.91	1	1	1	1	1
Results from D9-7-F	ρ	0.29	0.99	1	1	1	1
Results from [1]	ρ		0.15	0.23	0.26	0.57	1

表 5 以 JPEG 對影像進行壓縮攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

表 5 為在 JPEG 影像壓縮攻擊之下，本論文所提方法之五組實驗結果與參考文獻 [1] 之實驗結果比較。當壓縮係數為 30 時，三張影像中之五組實驗結果所測得之浮水印相關係數均高於 [1]，且係數值均維持在 0.99 以上；當壓縮係數降為 10 時，五組實驗結果皆仍然能夠成功偵測出浮水印之存在，所得之係數值分佈在 0.29 至 0.96 之間，雖然差距頗大，但確實可證明這五組方法均具有極佳的抗 JPEG 壓縮能力。

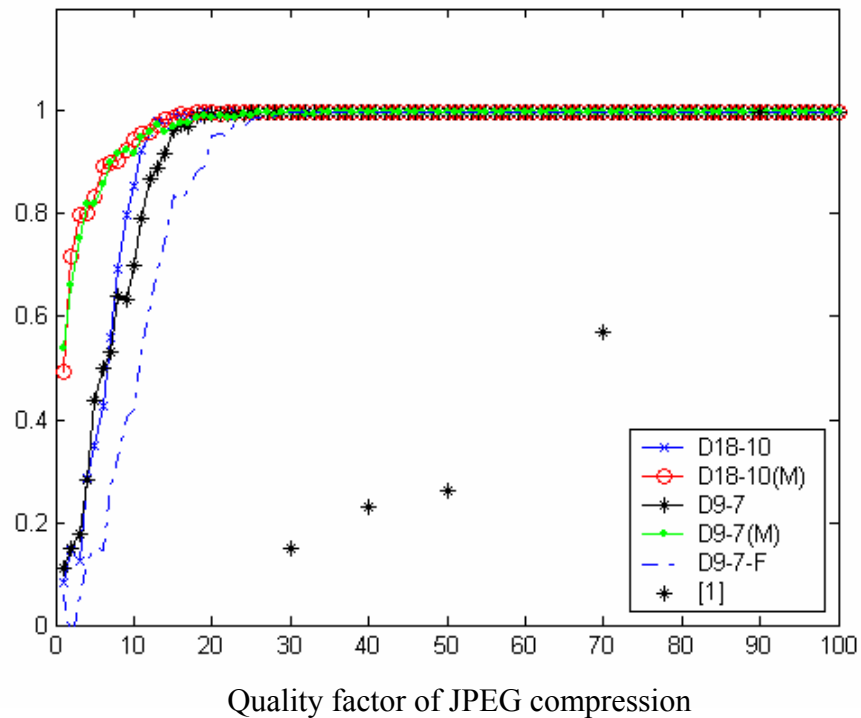
此外，我們發現在使用相同 filter 的情況下，加入影像切割重組之前置處理方法後所測得的相關係數值，皆高於原本的演算法所測得的數值，表示本論文所提出之動態能量差異化浮水印技術，在加上影像切割重組之前置處理方法後，確實能夠提升浮水印對於 JPEG 壓縮攻擊之抵抗能力，進而增加浮水印的強韌性。

至於隨機強制差異化方向之浮水印演算法，由於嵌入浮水印後的影像品質會較差，故在嵌入程序中必須將差異化間距調整為較低，以提升影像的品質，故在抵抗 JPEG 壓縮攻擊時，其抵抗效果自然也會較差。當同樣使用 9-7 filter 且壓縮係數設定為 10 的情況下，由「D9-7」與「D9-7-F」方法之實驗結果可以看出，隨機強制差異化方向浮水印方法所測得的相關係數值，皆低於原本的最適差異化方向演算法所測得的數值，且差距約在 0.3 左右，故可得知隨機強制差異化方向之浮水印強韌性較原本之浮水印方法為低。



(a)

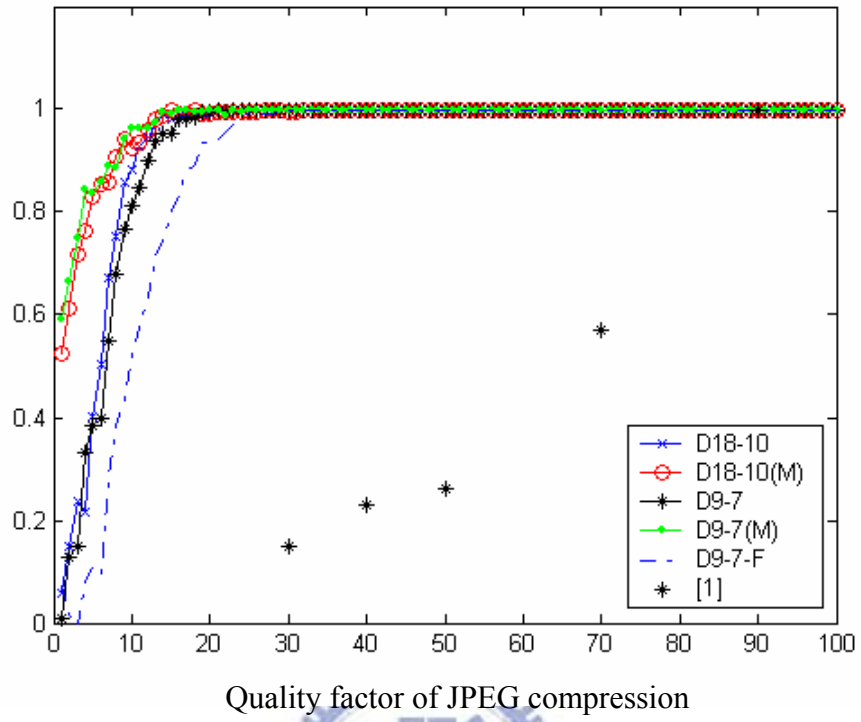
Correlation Value



(b)

Goldhill

Correlation Value



(c)

Peppers

Correlation Value

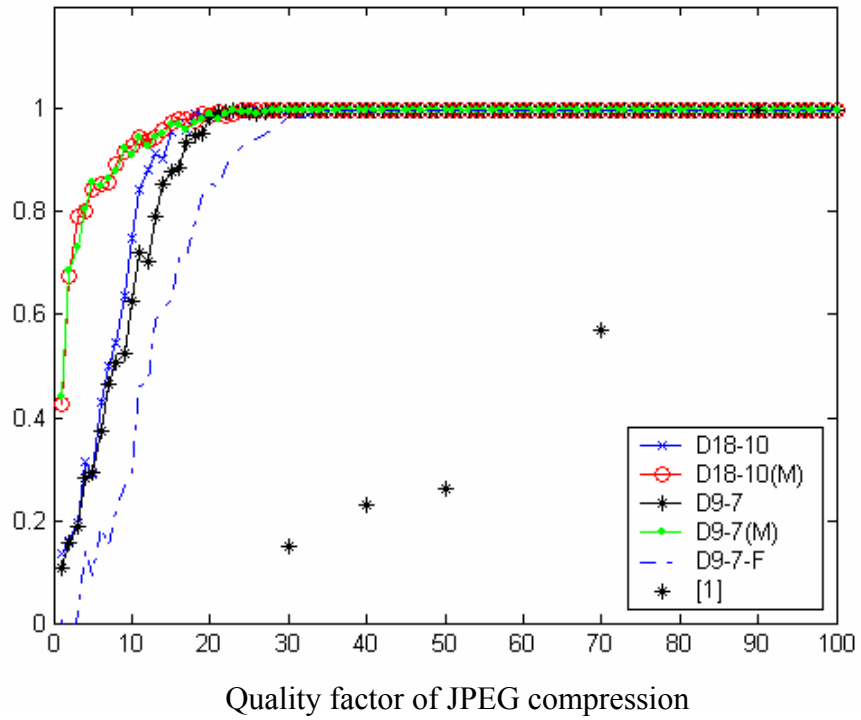


圖 31 三張影像內之浮水印對 JPEG 壓縮攻擊之承受程度 (a)Lena (b)Goldhill (c)Peppers

在圖 31 之實驗結果中，我們分別針對Lena、Godhill及Peppers等三張影像，將JPEG 壓縮係數設定為 1 至 100 以進行壓縮攻擊。由三張影像之實驗結果圖形中，可以看出本論文所提方法之五組實驗結果相較於 [1]而言，浮水印之強韌性皆獲得相當大的提升。

在實驗結果中亦可明顯比較出，加入影像切割重組之前置處理方法後，浮水印抵抗 JPEG 壓縮攻擊之能力會優於原本的浮水印演算法，而原本的浮水印演算法又優於隨機強制差異化方向之方法。

當壓縮係數降至 9 時，「D9-7-F」之方法在三張影像中仍可測得 0.27 以上之浮水印相關係數；當壓縮係數設定為 4 時，「D18-10」與「D9-7」之方法仍然能夠在三張影像中測得 0.24 以上之浮水印相關係數；而即使在壓縮係數為 1 時，「D18-10(M)」與「D9-7(M)」這兩組加入了影像前置處理的浮水印方法，也都能夠自三張影像中測得 0.43 以上之浮水印相關係數。由於當相關係數值高於門檻值 0.23 時，即表示可以成功偵測出浮水印的存在，使浮水印的強韌性獲得保障，故本論文所提之五組實驗結果均較[1]之實驗結果為佳，對於 JPEG 壓縮攻擊之抵抗能力亦較高。

5.2 SPIHT 壓縮攻擊

SPIHT(Set Partitioning in Hierarchical Trees)是一種以小波轉換為基礎的影像壓縮演算法，最早於 1996 年由 Said 等學者所提出，目前已經成為新一代影像壓縮標準 MPEG-4 與 JPEG-2000 之核心技術。

SPIHT 壓縮法是一種無失真的壓縮方法，與其他壓縮方法相較之下，SPIHT 的資料壓縮與解壓縮的速度相當迅速，且經壓縮過後的影像依然具有相當高的畫質。此外影像在傳輸時能提供漸近式的影像傳輸(Progressive Image Transmission)，即影像接收端能夠快速得到所接收影像之大致內容，可以有效節省影像傳輸的資料量與網路頻寬。



Lena 原始影像

以 SPIHT 對影像壓縮後之結果
Bitrate=0.3，PSNR=34.16， $\rho=0.76$

圖 32 Lena 原始影像與以 SPIHT 對影像進行壓縮攻擊後影像之比較

圖 32 為 Lena 影像以「D18-10」之方法嵌入浮水印，再以 SPIHT 壓縮法進行攻擊之影像前後對照。由實驗結果可知，當影像經 SPIHT 壓縮攻擊之後，本論文所提出之動

態能量差異化數位影像浮水印技術，仍然能夠自影像中偵測出浮水印的存在($\rho=0.76$)，故可證明本浮水印技術之強韌性確實具有抵抗 SPIHT 壓縮法的能力。

實驗結果：

(a) Lena

		Bitrate							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	
Results from D18-10	ρ	0.34	0.61	0.76	0.99	1	1	1	
	PSNR(dB)	30.15	32.80	34.16	34.79	35.33	35.79	35.97	
Results from D18-10(M)	ρ	0.88	0.95	0.96	0.98	1	1	1	
	PSNR(dB)	30.11	32.84	34.28	35.13	35.72	36.14	36.29	
Results from D9-7	ρ	0.41	0.99	1	1	1	1	1	
	PSNR(dB)	30.06	32.36	33.49	33.94	34.67	35.19	35.57	
Results from D9-7(M)	ρ	0.86	0.93	0.95	0.98	0.99	1	1	
	PSNR(dB)	30.14	32.84	34.24	34.94	35.54	36.00	36.16	
Results from D9-7-F	ρ	0.23	0.89	0.93	1	1	1	1	
	PSNR(dB)	29.96	32.17	33.27	33.70	34.33	34.83	35.17	
Results from [1]	ρ			0.21	0.41	0.85	0.83	0.85	
	PSNR(dB)			33.1	34.3	34.6	35.2	36.7	

(b) Goldhill

		Bitrate							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	
Results from D18-10	ρ	0.38	0.71	0.86	0.93	0.97	1	0.99	
	PSNR(dB)	27.89	29.64	30.85	31.70	32.40	33.07	33.57	
Results from D18-10(M)	ρ	0.82	0.89	0.93	0.95	0.97	0.99	1	
	PSNR(dB)	27.88	29.67	30.89	31.74	32.47	33.18	33.70	

Results from D9-7	ρ	0.41	0.85	0.99	1	1	1	1
	PSNR(dB)	27.82	29.45	30.60	31.38	31.94	32.64	33.18
Results from D9-7(M)	ρ	0.84	0.93	0.96	0.99	1	1	1
	PSNR(dB)	27.88	29.68	30.89	31.74	32.46	33.12	33.64
Results from D9-7-F	ρ	0.18	0.73	0.88	0.92	0.98	1	1
	PSNR(dB)	27.80	29.43	30.63	31.38	31.98	32.68	33.18
Results from [1]	ρ			0.21	0.41	0.85	0.83	0.85
	PSNR(dB)			33.1	34.3	34.6	35.2	36.7

(c) Peppers

		Bitrate	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Results from D18-10	ρ		0.34	0.55	0.82	0.98	0.99	1	1
	PSNR(dB)		29.95	32.62	33.77	34.44	34.88	35.20	35.46
Results from D18-10(M)	ρ		0.87	0.92	0.95	0.98	1	1	1
	PSNR(dB)		29.94	32.63	33.80	34.59	35.05	35.36	35.61
Results from D9-7	ρ		0.21	0.99	0.99	1	1	1	1
	PSNR(dB)		29.91	32.23	33.14	33.51	34.13	34.54	34.87
Results from D9-7(M)	ρ		0.86	0.91	0.95	0.98	0.99	1	1
	PSNR(dB)		29.94	32.63	33.81	34.53	34.99	35.29	35.55
Results from D9-7-F	ρ		0.10	0.51	0.92	1	1	1	1
	PSNR(dB)		29.84	32.30	33.29	34.01	34.47	34.79	35.08
Results from [1]	ρ				0.21	0.41	0.85	0.83	0.85
	PSNR(dB)				33.1	34.3	34.6	35.2	36.7

表 6 以 SPIHT 對影像進行壓縮攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

由表 6 所列之五組實驗結果與參考文獻 [1] 之實驗結果相比較，在三張不同的影像中，五組實驗結果所測得之浮水印相關係數均高於 [1]，當 Bitrate 為 0.3 時，測得之最低相關性係數值均在 0.76 以上，可以順利偵測出影像內嵌入之浮水印，故可證明本論文之方法在抵抗 SPIHT 壓縮攻擊的能力較 [1] 為佳。

此外在改進前、後之演算法效能比較方面，當使用 18-10 filter 作為小波轉換之濾波器組(Filter Bank)時，加入影像前置處理的改進後演算法所測得的相關係數值有明顯的提升，尤其在 Bitrate 為 0.1 時最為明顯，未加入影像前置處理之原始演算法所測得的相關性係數值在 0.34 至 0.38 之間，而加入影像前置處理後之相關性係數值則提高到 0.82 以上，差距值可達到 0.44 以上。

當我們使用 9-7 filter 作為小波轉換之濾波器組時，當 Bitrate 大於 0.2 時，未加入影像前置處理的原始演算法所測得的相關係數值較高，但差距甚小，最多僅 0.05；當 Bitrate 為 0.1 時，加入影像前置處理之方法所測得的相關性係數值反而較高，且差距值在 0.43 以上，故由實驗結果可得知，本論文所提出的影像切割重組之前置處理方法，確實能夠有效提升浮水印抵抗 SPIHT 壓縮攻擊的能力。

至於隨機強制差異化方向之浮水印演算法，與原始的動態能量差異化演算法相較之下，當 Bitrate 為 0.3 時，前者所測得的相關係數值就逐漸下降，且均低於後者；而當 Bitrate 為 0.1 時，隨機強制差異化方向之方法即無法偵測出浮水印的存在，故可知該方法在抵抗 SPIHT 壓縮攻擊方面之效能，不如原始的動態能量差異化浮水印演算法。

5.3 Median Filter 攻擊

Median Filter 是一種空間域影像處理技術，通常用於移除影像中多餘的雜訊 (Noise)，不過也會因此造成影像的模糊化。Median Filter 是非線性 filter 的一種，但較一般的非線性 filter 具有更高的強韌性，因為它能夠保存影像中的邊緣部份 (Sharp Edges)，不過缺點在於運算效率較低，原因為對於 Mask 內的數值進行排序時極為耗時。一般可用快速搜尋法 (quicksort) 解決排序時間過長的問題。

Median Filter 之運作原理介紹如下：通常我們使用一個 2-D 的 Mask 對一個大小為 $m \times n$ 的影像中所有的點進行掃描，當 Mask 運作至影像中任一個區域時，便將此區域的中間值取出，當作新的影像中的一個點。如圖 33 所示，以 3×3 的 Mask 對一張原始影像進行處理，最後取出中間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 中心位置為 (3,3)，最後從 Mask 中得到的中間值便存在右圖座標 (3,3) 位置。

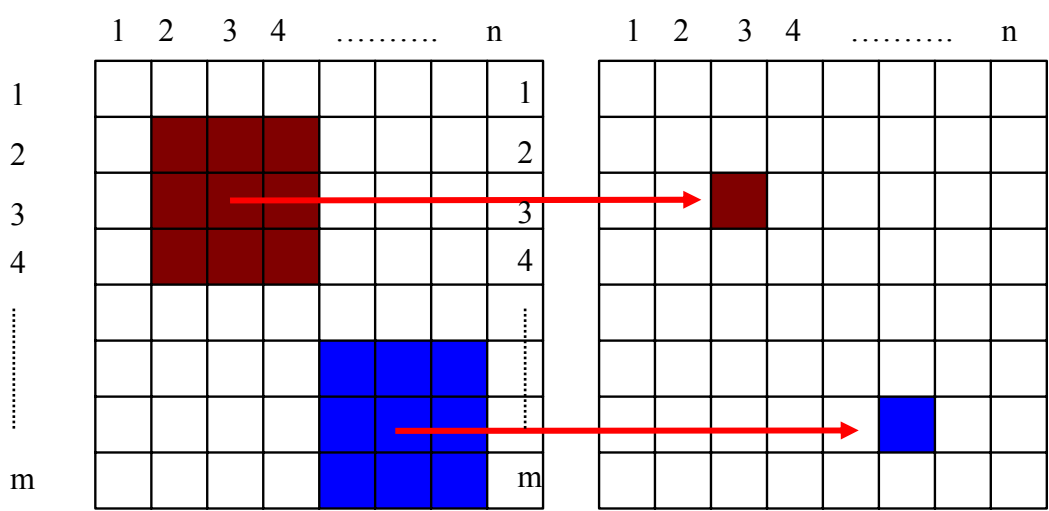


圖 33 Median Filter 運作原理

取中間值的方式，是將 Mask 中的資訊排序後，取出最中間的值，如下圖所示：

135	152	106
83	98	72
56	65	41

原始資料：135, 152, 106, 83, 98, 72, 56, 65, 41

排序資料：41, 56, 65, 72, **83**, 98, 106, 135, 152

取出中間值：83

圖 34 Median Filter 運作範例圖

圖 35 為原始 Lena 圖形與以「D18-10」方法嵌入浮水印後，再以 6×6 的 Mask 進行破壞後之影像的比較，遭破壞後之影像，其邊緣部份被強調了，同時也使影像變模糊。不過我們仍然可以從該影像中偵測出浮水印的存在($\rho = 0.89$)。

(a)

(b)



原始影像

以 6×6 Median Filter 對影像攻擊結果
 $\rho = 0.89$

圖 35 原始 Lena 影像與 Median Filter 處理過影像之對照圖 (a)原始影像 (b)以 6×6 Median Filter 處理過後之影像

實驗結果：

(a) Lena

Mask Size (nxn)		2	3	4	5	6	7	8	9	10	11	12	13
Results from D18-10	ρ	0.97	1	0.96	0.99	0.89	0.90	0.71	0.59	0.39	0.29	0.21	
	PSNR	29.23	34.53	29.66	31.16	28.65	29.09	27.61	27.80	26.78	26.84	26.07	
Results from D18-10(M)	ρ	0.97	1	0.95	0.95	0.86	0.86	0.83	0.80	0.81	0.79	0.78	0.78
	PSNR	29.25	34.60	29.67	31.20	28.65	29.11	27.61	27.81	26.79	26.86	26.08	26.07
Results from D9-7	ρ	0.98	1	0.94	0.98	0.84	0.83	0.64	0.54	0.21			
	PSNR	29.23	34.55	29.65	31.16	28.64	29.09	27.61	27.80	26.78			
Results from D9-7(M)	ρ	0.98	0.98	0.93	0.93	0.86	0.87	0.82	0.83	0.79	0.82	0.79	0.78
	PSNR	29.34	34.74	29.73	31.22	28.68	29.11	27.63	27.81	26.80	26.86	26.09	26.07
Results from D9-7-F	ρ	0.93	1	0.86	0.94	0.71	0.72	0.43	0.34	0.08			
	PSNR	29.26	34.32	29.61	31.05	28.61	29.05	27.60	27.79	26.79			
Results from [1]	ρ	0.38	0.51	0.23									
	PSNR												

(b) Goldhill

Mask Size (nxn)		2	3	4	5	6	7	8	9	10	11	12	13
Results from D18-10	ρ	0.97	1	0.97	0.98	0.84	0.79	0.64	0.50	0.27	0.19		
	PSNR	28.30	31.41	28.48	28.84	27.42	27.35	26.46	26.30	25.70	25.56		
Results from D18-10(M)	ρ	0.97	1	0.92	0.90	0.85	0.82	0.80	0.79	0.75	0.79	0.78	0.73
	PSNR	28.31	31.39	28.47	28.82	27.41	27.35	26.44	26.30	25.70	25.56	25.10	24.96
Results from D9-7	ρ	0.99	1	0.94	0.97	0.79	0.74	0.51	0.34	0.18			
	PSNR	28.28	31.41	28.47	28.84	27.41	27.34	26.45	26.30	25.71			

Results from D9-7(M)	ρ	0.98	0.99	0.93	0.91	0.85	0.82	0.80	0.80	0.78	0.80	0.76	0.77
	PSNR	28.34	31.43	28.49	28.85	27.43	27.36	26.46	26.31	25.71	25.57	25.10	24.96
Results from D9-7-F	ρ	0.92	1	0.85	0.90	0.61	0.61	0.32	0.19				
	PSNR	28.30	31.28	28.42	28.77	27.39	27.33	26.44	26.31				
Results from [1]	ρ	0.38	0.51	0.23									
	PSNR												

(c) Peppers

Mask Size (nxn)		2	3	4	5	6	7	8	9	10	11	12	13
Results from D18-10	ρ	0.97	1	0.96	0.99	0.90	0.88	0.66	0.54	0.32	0.24	0.23	0.18
	PSNR	29.59	34.50	29.94	32.23	29.25	30.47	28.34	29.01	27.44	27.79	26.57	26.74
Results from D18-10(M)	ρ	0.96	1	0.91	0.94	0.87	0.86	0.84	0.83	0.81	0.82	0.77	0.79
	PSNR	29.59	34.47	29.93	32.22	29.24	30.46	28.33	29.01	27.43	27.78	26.56	26.73
Results from D9-7	ρ	0.98	1	0.96	0.96	0.88	0.87	0.64	0.48	0.27	0.26	0.18	
	PSNR	29.56	34.49	29.93	32.23	29.25	30.47	28.35	29.02	27.44	27.78	26.57	
Results from D9-7(M)	ρ	0.98	0.99	0.90	0.91	0.86	0.86	0.82	0.82	0.83	0.81	0.80	0.79
	PSNR	29.62	34.56	29.96	32.26	29.27	30.48	28.35	29.02	27.44	27.79	26.57	26.74
Results from D9-7-F	ρ	0.93	1.00	0.91	0.93	0.75	0.68	0.43	0.29	0.14			
	PSNR	29.57	34.29	29.87	32.11	29.21	30.42	28.33	29.00	27.44			
Results from [1]	ρ	0.38	0.51	0.23									
	PSNR												

表 7 以 Median Filter 對影像進行攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

由表 7 所列之五組實驗結果與參考文獻 [1]之實驗結果相比較，[1]之實驗結果在 Mask大小為5×5時即無法偵測到浮水印，但本論文之五組實驗結果即使當Mask大小增

加到 6×6 時，測得之相關係數值至少都在 0.61 以上，故可證明本論文之方法能夠有效抵抗Median Filter的攻擊。

在改進前、後之演算法效能比較方面，無論是使用 18-10 filter 或是 9-7 filter，在 Mask 大小為 2×2 至 5×5 時，未加入影像前置處理的原始演算法測得的相關係數值稍高，但差距值最多不超過 0.06；當 Mask 大小為 6×6 以上時，加入影像前置處理後之演算法所測得的相關係數值轉而較高，且差距隨 Mask 大小增加而逐漸拉大；當 Mask 大小增加至 12×12 時，原始演算法即完全無法偵測出浮水印，但加入影像前置處理之演算法則仍可測得 0.76 以上之相關係數值，故在 Median Filter 攻擊實驗中，加入影像前置處理後的抵抗效果較佳。

在不同差異化方向演算法效能比較方面，我們比較「D9-7」與「D9-7-F」在三張影像之實驗結果即可發現，隨機強制差異化方向演算法所能承受 Median Filter 攻擊之 Mask 大小介於 8×8 至 9×9 之間，而原始浮水印方法所使用的最適差異化方向演算法所能承受之 Mask 大小則介於 9×9 至 11×11 之間，故可知強制差異化方向之方法在抵抗 Median Filter 攻擊方面之效能，不如原始的最適差異化方向演算法。

5.4 Gaussian Filter and Sharpening 攻擊

本實驗之測試採用StirMark[8][33][34]軟體進行測試，StirMark是一套測試浮水印強韌性之開放軟體，包含了各種形態的影像破壞。其初始版本日期為1997年11月，最新版本則是本實驗所使用之Strimark benchmark 4.0，2003年1月份版本[35]。

Gaussian Filter 及 Sharpening 均是屬於空間域的影像處理方法，其中 Gaussian Filter 能夠使影像達到模糊化的效果，實驗參數之 Mask 值為{1 2 1, 2 4 2, 1 2 1}，而 Scaling Factor 設為 16。

Sharpening 則是與 Gaussian Filter 相反，能夠使影像銳化以加強影像的對比效果，其作法是令影像邊緣及黑白相間部份的訊號增加強度，實驗參數之 Mask 值為{0 -1 0, 1 2 1, 0 -1 0}。



實驗結果：

(a) Gaussian Filter

Image	Lena	Goldhill	Peppers
Results from D18-10	0.72	0.83	0.71
Results from D18-10(M)	0.72	0.77	0.76
Results from D9-7	0.72	0.80	0.71
Results from D9-7(M)	0.77	0.69	0.78
Results from D9-7-F	0.59	0.75	0.67
Results from [1]	0.64	0.56	0.74

(b) Sharpening

Image	Lena	Goldhill	Peppers
Results from D18-10	0.98	0.99	0.99
Results from D18-10(M)	0.99	0.98	0.98
Results from D9-7	0.98	0.99	1.00
Results from D9-7(M)	0.96	0.97	0.98
Results from D9-7-F	0.98	0.98	0.98
Results from [1]	0.46	0.39	0.62

表 8 以 Gaussian Filter 及 Sharpening 攻擊後之實驗結果 (a) Gaussian Filter (b) Sharpening

由實驗結果可以看出，在 Gaussian Filter 攻擊實驗中，本論文所提方法之五組實驗結果於 Goldhill 影像實驗中，偵測出之相關係數值皆高於 [1]；在 Lena 影像實驗中，僅「D9-7-F」所得之相關係數值較 [1] 為低；至於在 Peppers 影像實驗中，「D18-10」、「D9-7」及「D9-7-F」所得之相關係數值均較 [1] 為低。此外在五組實驗數據中，原始演算法與加入影像前置處理後之演算法所測得之相關係數值互有高低，沒有非常明確的優劣比較。而使用隨機強制差異化方向之浮水印演算法，在三張實驗影像中所測得之相關係數值均較原始演算法為低，意即隨機強制差異化方向會使得浮水印抵抗 Gaussian Filter 攻擊的效能下降。

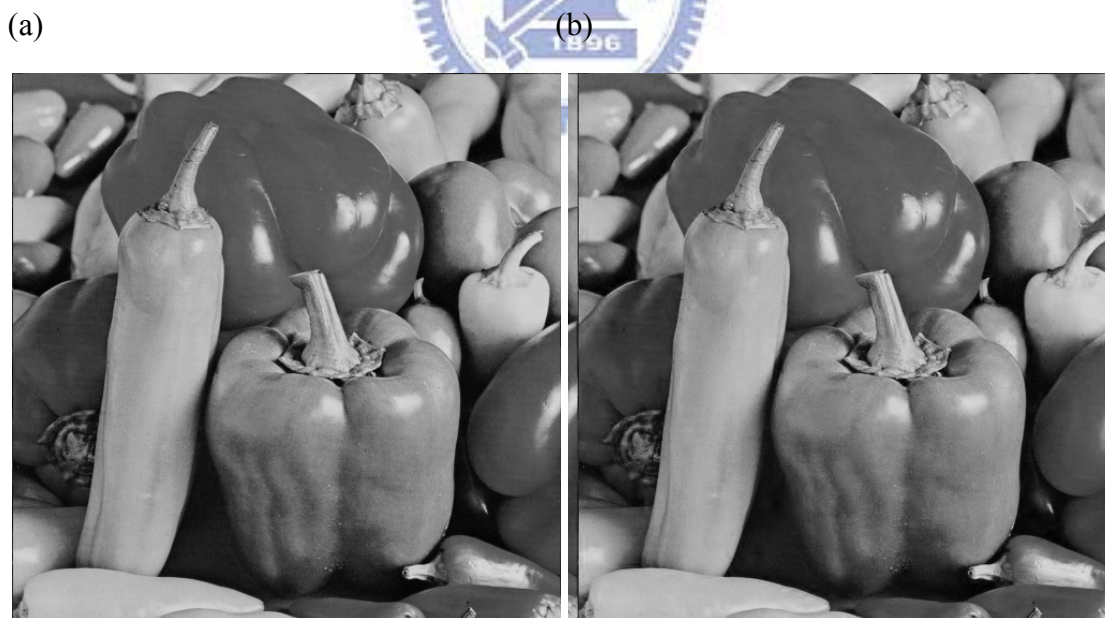
至於在 Sharpening 攻擊實驗中，本論文所提方法之五組實驗數據均較 [1] 為佳，而原始演算法所測得之浮水印相關係數值雖較兩種改進後之演算法為高，但差距值僅 0.02 左右，故沒有非常明確的優劣比較。

5.5 Pixel Shifting 攻擊

Pixel Shifting 屬於一種空間域的影像處理方法，主要是對於影像中每一列(row)或每一行(column)的像素進行橫向或縱向的位移，以達到對影像之所有像素進行移轉的目的。經移轉過後的影像與原始影像就外觀而言，對於人類肉眼上之判斷不會造成太大差異，不過當轉換至頻率域時，兩張影像之係數值卻有極大的變化，故 Pixel Shifting 會因此造成以頻率域係數為基礎的浮水印演算法，在驗證取出之浮水印是否真確時產生偏差。Pixel Shifting 在浮水印攻擊實驗中，可以依位移方向分為兩種類型，分述如下：

一、Type 1：水平位移

令影像中每一列像素均向右位移 n 個位置，並將最右方超出影像邊界的 n 個像素搬移至影像的最左方，形成一個向右位移的循環。如圖 36 所示：



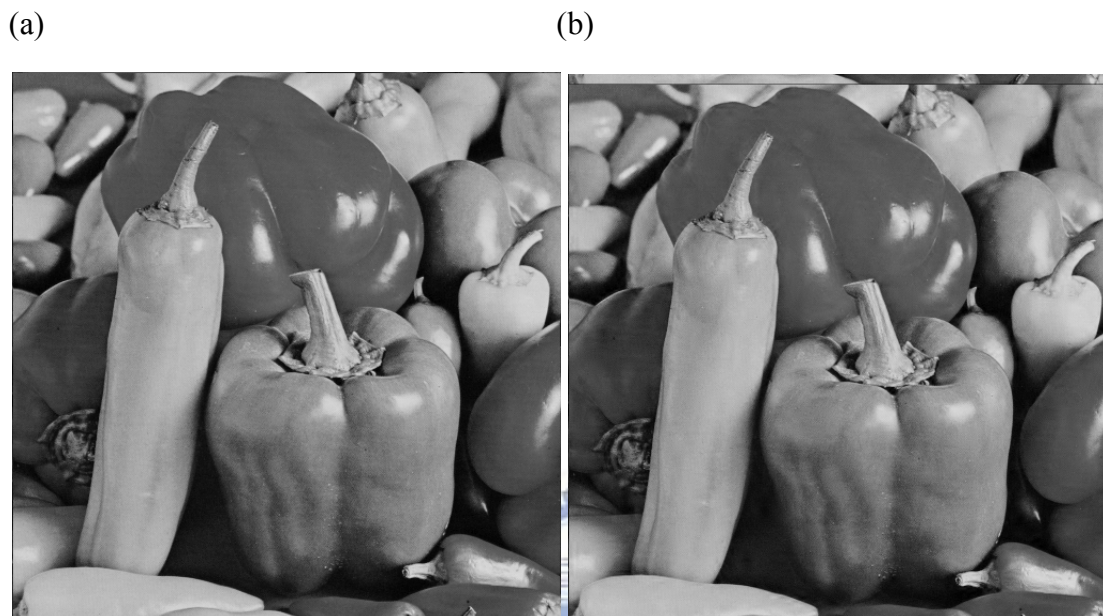
Peppers 原始影像

向右位移 9 個像素後之影像

圖 36 原始 Peppers 影像與向右位移 9 個像素後之影像對照圖

二、Type 2：垂直位移

令影像中每一行像素均向下位移 n 個位置，並將最下方超出影像邊界的 n 個像素搬移至影像的最上方，形成一個向下位移的循環。如圖 37 所示：



Peppers 原始影像

向下位移 9 個像素後之影像

圖 37 原始 Peppers 影像與向下位移 9 個像素後之影像對照圖

實驗結果：

一、Type 1：水平位移

(a) Lena

Shifting Pixels	3	4	5	6	7	8	9	10	11	12
Results from D18-10	0.48	0.32	0.35	0.35	-0.06					
Results from D18-10(M)	0.62	0.57	0.48	0.38	0.34	0.30	0.31	0.28	0.23	0.21
Results from D9-7	0.36	0.38	0.34	0.23	-0.14					
Results from D9-7(M)	0.50	0.48	0.39	0.34	0.31	0.34	0.32	0.26	0.22	

Results from D9-7-F	0.30	0.27	0.40	0.24	0.00						
Results from [1]			0.28	0.34	0.29	0.81	0.26	0.19			

(b) Goldhill

Shifting Pixels	3	4	5	6	7	8	9	10	42	43	56	57
Results from D18-10	0.35	0.29	0.35	0.38	0.00							
Results from D18-10(M)	0.66	0.59	0.56	0.54	0.48	0.48	0.44	0.42	0.30	0.31	0.24	0.22
Results from D9-7	0.44	0.43	0.52	0.29	-0.09							
Results from D9-7(M)	0.65	0.63	0.55	0.53	0.54	0.49	0.42	0.44	0.28	0.22		
Results from D9-7-F	0.25	0.30	0.34	0.13								
Results from [1]			0.36	0.35	0.41	0.84	0.29	0.21				

(c) Peppers

Shifting Pixels	3	4	5	6	7	8	9	10	12	13	14	15
Results from D18-10	0.41	0.31	0.37	0.36	-0.03							
Results from D18-10(M)	0.57	0.51	0.48	0.44	0.43	0.34	0.31	0.27	0.30	0.30	0.27	0.22
Results from D9-7	0.48	0.51	0.51	0.27	-0.10							
Results from D9-7(M)	0.66	0.54	0.51	0.46	0.43	0.41	0.39	0.33	0.27	0.19		
Results from D9-7-F	0.32	0.32	0.44	0.26	0.09							
Results from [1]			0.32	0.34	0.29	0.92	0.29	0.26				

表 9 以 Pixel Shifting Type-1 對影像進行攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

由Type-1 實驗結果可以看出，尚未加入影像切割重組之前置處理的「D18-10」與「D9-7」方法，對於Pixel Shifting Type-1 攻擊僅能抵抗至 6 個像素位移，當像素數量增加至 7 以上時，即無法偵測出浮水印，其效能較可抵抗 9 至 10 個像素位移的 [1]之實驗結果為差；然而當我們加入了影像前置處理方法後，由「D18-10(M)」及「D9-7(M)」方法的實驗數據可看出，浮水印抵抗Pixel Shifting Type-1 攻擊之能力提升許多，即使當像素位移數量增加到 11 個，依然可以順利偵測出浮水印，Goldhill影像甚至可抵抗至 42 個像素位移。故在Type-1 攻擊實驗中，加入影像前置處理確實可大幅提升浮水印的強韌性。

此外由隨機強制差異化方向演算法「D9-7-F」之實驗數據可知，該方法對於 Pixel Shifting Type-1 攻擊僅能抵抗 5 至 6 個像素位移，且相關係數值較「D9-7」之實驗數據為低，故雖然兩種方法能夠抵抗之像素位移數量相差不多，但仍然是原始的最適差異化方向演算法所產生之浮水印強韌性較高。



二、Type 2：垂直位移

(a) Lena

Shifting Pixels	3	4	5	6	7	8	9	10	20	25	26	27
Results from D18-10	0.39	0.33	0.51	0.50	0.04							
Results from D18-10(M)	0.75	0.67	0.62	0.55	0.53	0.52	0.50	0.50	0.32	0.23	0.20	
Results from D9-7	0.39	0.33	0.50	0.31	-0.04							
Results from D9-7(M)	0.66	0.62	0.63	0.56	0.59	0.52	0.47	0.45	0.28	0.23	0.29	0.22
Results from D9-7-F	0.29	0.24	0.41	0.22								
Results from [1]			0.27	0.33	0.27	0.82	0.25	0.17				

(b) Goldhill

Shifting Pixels	3	4	5	6	7	8	9	10	15	20	21	22
Results from D18-10	0.41	0.25	0.46	0.35	-0.03							
Results from D18-10(M)	0.63	0.56	0.56	0.54	0.51	0.48	0.51	0.47	0.36	0.29	0.24	0.21
Results from D9-7	0.38	0.32	0.46	0.25	-0.12							
Results from D9-7(M)	0.61	0.58	0.57	0.53	0.50	0.51	0.47	0.42	0.33	0.25	0.21	
Results from D9-7-F	0.29	0.18	0.29	0.03								
Results from [1]			0.37	0.31	0.43	0.85	0.25	0.20				

(c) Peppers

Shifting Pixels	3	4	5	6	7	8	9	10	16	17	20	21
Results from D18-10	0.38	0.19	0.46	0.41	0.07							
Results from D18-10(M)	0.65	0.64	0.57	0.52	0.49	0.45	0.43	0.44	0.36	0.32	0.27	0.19
Results from D9-7	0.38	0.30	0.47	0.31	-0.06							
Results from D9-7(M)	0.66	0.61	0.59	0.53	0.55	0.46	0.43	0.42	0.25	0.22		
Results from D9-7-F	0.30	0.23	0.39	0.18								
Results from [1]			0.34	0.33	0.31	0.91	0.28	0.27				

表 10 以 Pixel Shifting Type-2 對影像進行攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

由Type-2 實驗結果可以看出，尚未加入影像切割重組之前置處理的「D18-10」與「D9-7」方法，對於Pixel Shifting Type-2 攻擊亦僅能抵抗至 6 個像素位移，當像素數量

增加至 7 以上時，即無法偵測出浮水印，其效能較可抵抗 9 至 10 個像素位移的 [1] 之實驗結果為差；但若當我們加入了影像前置處理方法後，由「D18-10(M)」及「D9-7(M)」方法的實驗數據可看出，浮水印抵抗 Pixel Shifting Type-2 攻擊之能力提升許多，即使當像素位移數量增加到 16 個，依然可以順利偵測出浮水印，Lena 影像甚至可抵抗至 25 個像素位移。故在 Type-2 攻擊實驗中，加入影像前置處理的方法亦能夠大幅提升浮水印的強韌性。

至於隨機強制差異化方向演算法「D9-7-F」對於 Pixel Shifting Type-2 攻擊僅能抵抗 5 個像素位移，而「D9-7」方法則能夠抵抗 6 個像素位移，故在 Type-2 攻擊實驗中，依然是原始的最適差異化方向演算法所產生之浮水印強韌性較高。

綜合 Type-1 及 Type-2 之結論，原始的動態能量差異化浮水印演算法抵抗 Pixel Shifting 攻擊的能力不如 [1]，但較隨機強制差異化方向演算法之抵抗效果為佳；若是加入了影像切割重組之前置處理方法後，其抵抗 Pixel Shifting 攻擊的能力可獲得大幅提升並超越 [1]，進而達到浮水印強韌性的要求。



5.6 Rotation and Scaling 攻擊

本實驗之測試亦採用 StirMark 軟體進行測試，使用版本為 Strimark benchmark 4.0，發表於 2003 年 1 月。

在 Rotation and Scaling 攻擊之中，Rotation 的單位為度(degree)，正值表示影像向左旋轉(逆時鐘方向)，負值則表示影像向右旋轉(順時鐘方向)，如圖 38 所示。此種影像處理之攻擊方式是屬於空間域的幾何攻擊，當影像經過旋轉至某個角度時，StirMark 軟體會將產生的空白處切除，並調整剩餘影像至原本的大小，故影像會因此而具有放大的效果。由實驗結果可以看出，當旋轉角度愈大，影像品質會變的愈差，偵測出的浮水印相關係數也愈低。



圖 38 原始 Goldhill 影像與經過 Rotation and Scaling 攻擊後之影像對照圖 (a) 正值角度表示逆時鐘方向旋轉 (b) 負值角度表示順時鐘方向旋轉

本論文所提方法中，原本的浮水印演算法「D18-10」與「D9-7」，在旋轉角度小於 ± 0.75 度時，其偵測出之浮水印相關係數值皆較 [1] 為高，但當旋轉角度大於 ± 0.75 時，在部份影像中就無法偵測出浮水印，一但旋轉角度大於 ± 1.0 以上，就完全偵測不出浮水印了。故就相關係數值之比較而言，「D18-10」與「D9-7」兩組方法在抵抗Rotation and Scaling攻擊時之效能皆較 [1] 為佳，但當旋轉角度大於 ± 1.0 以上時，則無法抵抗Rotation and Scaling之攻擊。

若是我們在原本的浮水印演算法中加入了影像切割重組之前置處理方法後，抵抗Rotation and Scaling 攻擊的能力可立即獲得大幅提升。我們由「D18-10(M)」與「D9-7(M)」兩組方法之數據中可以看出，當旋轉角度提高至 3.5 度時，偵測出最低之浮水印相關係數值尚在 0.25 以上，且 Goldhill 影像甚至可於旋轉角度為 6.5 度時偵測出浮水印；另外若旋轉角度為 -2.5 度時，最低之相關係數值亦在 0.27 以上，且 Goldhill 影像亦可於旋轉角度為 -6.25 度時偵測出浮水印，其效能高出「D18-10」與「D9-7」甚多。

至於隨機強制差異化方向演算法「D9-7-F」抵抗Rotation and Scaling攻擊之能力則與 [1] 差不多，可容忍之旋轉角度範圍大致為 ± 0.75 度之間，而「D9-7」方法能夠抵抗之正值旋轉角度為 0.75 度，與「D9-7-F」方法相同，但可抵抗之負值旋轉角度則介於 -0.75 至 -1.0 度，較「D9-7-F」方法為佳，故可知在Rotation and Scaling攻擊實驗中，隨機強制差異化方向演算法所產生之浮水印強韌性，低於原始浮水印方法的最適差異化方向演算法。

綜上所述，於Rotation and Scaling 攻擊實驗中，我們可證實加入影像切割重組之前置處理方法後，確實可大幅提升本論文所提之影像浮水印演算法之強韌性，而隨機強制差異化方向演算法則會略微降低浮水印之強韌性。實驗結果可參照表 11 之數據整理如下：

實驗結果：

(a) Lena

Rotation degree		0.25	0.5	0.75	1.0	1.25	2.0	3.0	3.5	3.75	4.0	4.75	5.0
Results from D18-10	ρ	0.87	0.46	0.35	0.25	0.09							
	PSNR	24.07	22.19	20.79	19.75	18.97							
Results from D18-10(M)	ρ	0.95	0.79	0.74	0.63	0.56	0.48	0.36	0.35	0.29	0.28	0.26	0.20
	PSNR	24.09	22.21	20.79	19.76	18.97	17.36	16.10	15.67	15.48	15.31	14.87	14.74
Results from D9-7	ρ	0.82	0.44	0.36	0.21								
	PSNR	24.07	22.18	20.77	19.74								
Results from D9-7(M)	ρ	0.93	0.79	0.67	0.66	0.59	0.40	0.30	0.25	0.20			
	PSNR	24.09	22.21	20.8	19.76	18.97	17.37	16.10	15.66	15.48			
Results from D9-7-F	ρ	0.80	0.35	0.27	0.11								
	PSNR	24.08	22.21	20.80	19.77								
Results from [1]	ρ	0.37	0.29	0.26	0.24								
	PSNR												

Rotation degree		-0.25	-0.5	-0.75	-1.0	-1.25	-1.5	-2.0	-3.0	-3.25	-3.5	-4.0	-4.25
Results from D18-10	ρ	0.86	0.41	0.30	0.20								
	PSNR	23.46	21.27	19.82	18.79								
Results from D18-10(M)	ρ	0.96	0.80	0.68	0.61	0.52	0.47	0.37	0.25	0.23	0.21		
	PSNR	23.47	21.28	19.83	18.8	18.02	17.4	16.49	15.37	15.16	14.94		
Results from D9-7	ρ	0.84	0.45	0.31	0.26	0.20							
	PSNR	23.45	21.25	19.81	18.79	18.01							
Results from D9-7(M)	ρ	0.95	0.73	0.67	0.56	0.48	0.48	0.40	0.28	0.27	0.28	0.25	0.14
	PSNR	23.48	21.29	19.84	18.81	18.03	17.4	16.50	15.38	15.16	14.94	14.51	14.31

Results from D9-7-F	ρ	0.75	0.34	0.25	0.13								
	PSNR	23.47	21.29	19.84	18.81								
Results from [1]	ρ	0.32	0.23	0.24	0.16								
	PSNR												

(b) Goldhill

Rotation degree		0.25	0.5	0.75	1.0	1.25	3.0	5.0	6.5	6.75	7.0	8.25	8.5
Results from D18-10	ρ	0.83	0.43	0.30	0.26	0.16							
	PSNR	23.38	21.56	20.54	19.84	19.32							
Results from D18-10(M)	ρ	0.95	0.76	0.66	0.63	0.57	0.37	0.32	0.29	0.27	0.29	0.23	0.22
	PSNR	23.39	21.58	20.56	19.85	19.33	17.35	16.31	15.78	15.69	15.60	15.20	15.13
Results from D9-7	ρ	0.75	0.40	0.28	0.14								
	PSNR	23.37	21.55	20.53	19.83								
Results from D9-7(M)	ρ	0.92	0.75	0.67	0.57	0.57	0.46	0.29	0.23	0.21			
	PSNR	23.39	21.59	20.56	19.86	19.33	17.35	16.30	15.77	15.69			
Results from D9-7-F	ρ	0.66	0.29	0.23	0.11								
	PSNR	23.39	21.59	20.57	19.87								
Results from [1]	ρ	0.33	0.24	0.21	0.15								
	PSNR												

Rotation degree		-0.25	-0.5	-0.75	-1.0	-2.0	-3.0	-4.0	-5.0	-6.25	-6.5	-8.0	-8.25
Results from D18-10	ρ	0.82	0.42	0.33	0.21								
	PSNR	23.20	21.50	20.50	19.82								
Results from D18-10(M)	ρ	0.91	0.78	0.64	0.63	0.50	0.45	0.41	0.37	0.24	0.21		
	PSNR	23.21	21.51	20.52	19.83	18.35	17.50	16.92	16.44	15.94	15.85		

Results from D9-7	ρ	0.74	0.43	0.26	0.16								
	PSNR	23.18	21.48	20.49	19.81								
Results from D9-7(M)	ρ	0.95	0.75	0.65	0.63	0.49	0.48	0.43	0.34	0.31	0.32	0.23	0.22
	PSNR	23.21	21.52	20.53	19.84	18.35	17.51	16.92	16.44	15.94	15.85	15.34	15.27
Results from D9-7-F	ρ	0.68	0.28	0.20									
	PSNR	23.22	21.54	20.55									
Results from [1]	ρ	0.38	0.27	0.25	0.14								
	PSNR												

(c) Peppers

Rotation degree		0.25	0.5	0.75	1.0	1.25	1.5	2.0	2.5	3.0	3.25	4.25	4.5
Results from D18-10	ρ	0.84	0.48	0.35	0.26	0.13							
	PSNR	23.19	21.38	20.04	19.02	18.20							
Results from D18-10(M)	ρ	0.89	0.73	0.68	0.60	0.57	0.49	0.41	0.34	0.24	0.21		
	PSNR	23.2	21.39	20.05	19.03	18.21	17.55	16.54	15.78	15.20	14.96		
Results from D9-7	ρ	0.82	0.49	0.33	0.20								
	PSNR	23.17	21.35	20.01	19.00								
Results from D9-7(M)	ρ	0.90	0.71	0.70	0.61	0.52	0.52	0.41	0.39	0.27	0.26	0.25	0.20
	PSNR	23.19	21.38	20.04	19.02	18.21	17.55	16.53	15.78	15.20	14.95	14.17	14.01
Results from D9-7-F	ρ	0.76	0.43	0.28	0.12								
	PSNR	23.18	21.38	20.04	19.02								
Results from [1]	ρ	0.41	0.30	0.26	0.17								
	PSNR												

Rotation degree		-0.25	-0.5	-0.75	-1.0	-1.25	-1.5	-2.0	-2.5	-2.75	-3.0	-3.5	-3.75
Results from D18-10	ρ	0.82	0.50	0.35	0.25	0.23	0.13						
	PSNR	23.02	20.93	19.52	18.47	17.63	16.95						
Results from D18-10(M)	ρ	0.87	0.73	0.67	0.59	0.53	0.45	0.44	0.27	0.21			
	PSNR	23.03	20.94	19.53	18.47	17.64	16.96	15.91	15.13	14.81			
Results from D9-7	ρ	0.82	0.49	0.38	0.27	0.22							
	PSNR	23.00	20.91	19.50	18.45	17.62							
Results from D9-7(M)	ρ	0.91	0.71	0.64	0.58	0.51	0.46	0.38	0.28	0.25	0.25	0.23	0.21
	PSNR	23.03	20.94	19.53	18.47	17.63	16.96	15.91	15.13	14.81	14.53	14.05	13.85
Results from D9-7-F	ρ	0.75	0.38	0.23	0.20								
	PSNR	23.03	20.95	19.54	18.48								
Results from [1]	ρ	0.39	0.25	0.25	0.16								
	PSNR												

表 11 以 Rotation and Scaling 對影像攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers



5.7 Multiple Watermarking 攻擊

Multiple Watermarking 攻擊實驗之作法，是對於已嵌入浮水印的影像額外嵌入多個相異之浮水印，並測試後來嵌入之浮水印是否會蓋過先前的浮水印，使原本在影像中之初始浮水印消失。

由於一張影像之版權可能會授予多人使用，且版權亦可能經過多次的買賣，故一張數位影像中含有多個不同的浮水印是正常且必要的功能，也因此良好的浮水印演算法必須能夠承受多個浮水印的嵌入，並可將嵌入的每一個浮水印順利的取出，如此方能確保每一個不同的版權資訊均安全的存在於影像之中。

在加入多個浮水印之後，影像的品質必定會受到影響，其 PSNR 值會因而下降，圖 39 為使用「D18-10」方法額外嵌入 5 個浮水印後的對照圖，雖然影像品質變差了，但我們依然能夠順利的偵測出浮水印之相關係數值。

(a)



Goldhill 原始影像

(b)



額外加入 5 個浮水印後之影像
額外浮水印個數=5，PSNR=30.80， $\rho=1$

圖 39 原始 Goldhill 影像與經過 Multiple Watermarking 攻擊後之影像對照圖

實驗結果：

(a) Lena

Number of Watermarks		1	2	3	4	5	6	7	14	422	471	600	1000
Results from D18-10	ρ	1	1	1	1	1	1	1	1	1	1	1	1
	PSNR	35.37	33.64	32.47	31.54	30.80	30.18	29.65	27.40	22.80	22.71	22.51	22.01
Results from D18-10(M)	ρ	1	1	1	0.99	1	0.99	0.99	0.96	0.30	0.20		
	PSNR	34.98	33.13	31.83	30.75	29.92	29.19	28.58	25.79	12.99	12.70		
Results from D9-7	ρ	1	1	1	1	1	1	1	1	0.98	0.98	0.97	0.96
	PSNR	35.24	33.51	32.27	31.30	30.54	29.92	29.37	26.95	19.01	18.82	18.40	17.62
Results from D9-7(M)	ρ	1	1	1	1	1	0.99	1	0.98	0.21			
	PSNR	35.19	33.38	32.09	31.04	30.23	29.55	28.95	26.18	13.17			
Results from D9-7-F	ρ	0.89	0.82	0.74	0.67	0.64	0.60	0.56	0.22				
	PSNR	34.89	33.17	31.96	31.09	30.34	29.69	29.17	26.82				
Results from [1]	ρ	0.65	0.41	0.27	0.11								
	PSNR	35.50	32.78	29.35	28.05								

(b) Goldhill

Number of Watermarks		1	2	3	4	5	6	7	16	408	422	600	1000
Results from D18-10	ρ	1	1	1	1	1	1	1	1	1	1	1	1
	PSNR	35.77	34.12	33.01	32.12	31.44	30.86	30.35	27.88	23.73	23.70	23.38	22.82
Results from D18-10(M)	ρ	1	1	0.99	0.99	0.99	0.96	0.96	0.93	0.26	0.21		
	PSNR	35.53	33.75	32.37	31.31	30.52	29.86	29.23	25.87	13.49	13.39		
Results from D9-7	ρ	1	1	1	1	1	1	1	1	0.99	0.98	0.98	0.97
	PSNR	35.72	34.02	32.85	31.92	31.15	30.53	30.00	27.28	18.88	18.79	17.97	16.97

Results from D9-7(M)	ρ	1	1	1	1	1	1	1	0.98	0.22			
	PSNR	35.68	33.79	32.49	31.51	30.71	30.00	29.42	26.11	13.71			
Results from D9-7-F	ρ	0.93	0.86	0.80	0.71	0.66	0.64	0.58	0.22				
	PSNR	35.20	33.71	32.44	31.50	30.81	30.24	29.74	26.99				
Results from [1]	ρ	0.79	0.45	0.31	0.18								
	PSNR	35.26	31.50	29.71	28.57								

(c) Peppers

Number of Watermarks		1	2	3	4	5	6	7	17	501	512	600	1000
Results from D18-10	ρ	1	1	1	1	1	1	1	1	1	1	1	1
	PSNR	36.95	35.22	34.09	33.19	32.42	31.84	31.35	28.64	24.21	24.19	24.02	23.40
Results from D18-10(M)	ρ	1	1	1	0.99	0.98	0.97	0.96	0.93	0.22			
	PSNR	36.37	34.47	33.13	32.11	31.26	30.59	30.00	26.20	13.33			
Results from D9-7	ρ	1	1	1	1	1	1	1	1	0.98	0.97	0.97	0.97
	PSNR	36.79	35.04	33.81	32.82	32.05	31.41	30.87	27.74	18.71	18.68	18.39	17.59
Results from D9-7(M)	ρ	1	1	1	1	1	1	1	0.97	0.27	0.22		
	PSNR	36.56	34.61	33.29	32.27	31.41	30.70	30.14	26.66	13.51	13.45		
Results from D9-7-F	ρ	0.94	0.86	0.79	0.73	0.68	0.61	0.57	0.20				
	PSNR	36.17	34.55	33.41	32.37	31.67	31.06	30.52	27.48				
Results from [1]	ρ	0.80	0.53	0.31	0.22								
	PSNR	34.53	31.99	30.19	28.81								

表 12 以 Multiple Watermarking 對影像攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

從實驗結果可以看出，當額外嵌入的浮水印個數增加至 5 個時，本論文所提方法之五組實驗結果均仍然能夠偵測出原始的浮水印，且除了「D9-7-F」方法之相關係數值介

於 0.64 至 0.68 之間外，其餘四組方法之係數值皆趨近於 1，遠高於 [1] 之實驗結果。此外五組實驗方法所產生之影像品質 PSNR 值也都能維持在 30 左右，不致因浮水印數量額外增加 5 個而受到太大的影響。然而若浮水印個數超過 5 個以上時，PSNR 值會逐漸低於 30，即影像品質會愈來愈差。

在改進前、後之浮水印強韌性比較方面，未加入影像前置處理的原始浮水印演算法，即使額外嵌入 1000 個浮水印，其相關係數值仍能維持在 0.96 以上；而加入影像切割重組之前置處理方法後，當額外嵌入之浮水印個數達到 400 至 500 個後，就會發生偵測不出原始浮水印的情形，故在多重浮水印攻擊實驗中，影像前置處理會降低可供嵌入的浮水印個數。不過就實際應用面而言，當加入的浮水印個數過多而導致影像品質太過低劣時，反而會使該張影像失去了商業價值，即使該張影像中仍然可以偵測出每一個嵌入之浮水印的存在，但意義已經不大了。故在探討多重浮水印實驗時，必須將影像品質一併納入考量，以得到較為周全的結論。



至於在不同差異化方向演算法之效能比較方面，隨機強制差異化方向演算法可能會導致影像的小波係數值劇烈的變更，如不同時間嵌入的浮水印位元 W_n 與 $-W_n$ 可能以相同的差異化方向作嵌入，此時前一個嵌入的浮水印位元就會被後者所消除掉。然而最適差異化方向演算法則會選擇小波係數值改變量最少的方向作嵌入，故浮水印位元 W_n 與 $-W_n$ 必定會以不同的差異化方向作嵌入，浮水印資訊不易被消除。由「D9-7」與「D9-7-F」方法之實驗數據可看出，當額外嵌入之浮水印個數達 17 個以上時，「D9-7-F」方法即無法偵測出原始的浮水印，且影像品質均以「D9-7」方法較佳，故可知隨機強制差異化方向之方法在抵抗 Multiple Watermarking 攻擊方面之效能，遠不如原始浮水印方法之最適差異化方向演算法。

5.8 Bitplane Removal 攻擊

Bitplane Removal 攻擊法主要是針對影像中組成小波樹的頻率域係數值加以破壞，其作法是先將影像自空間域轉換至頻率域，再將組成小波樹的頻率域係數以二進位形式表示，此時每個係數值均是由許多的位元所組成，當這些位元組合起來即可視為一位元平面，其正負號位元另外記錄，如同 [1] 在嵌入浮水印的過程中所建立的位元平面一般。當我們移除這些係數的最低位元 (LSB) 時，即表示移除該位元平面的最低一層平面。假設某頻率域係數之最低位元值為 1，則該係數值就會因此而減少 $1 \times 2^0 = 1$ 。當移除的位元平面階層增加，則影像中被移除的資訊就會變多，可能導致係數結構的變化而使浮水印消失。

圖 40 即為使用「D18-10(M)」方法且移除組成小波樹之係數的 5 個最低位元後之影像，與原始影像之對照，5 個最低位元被移除過後之影像，其影像品質因資訊遭移除而降低，然而我們依舊能夠自影像中取出嵌入的浮水印。



(a)

(b)



原始 Lena 影像

移除頻率域係數中 5 個最低位元後的影像，PSNR=32.90， $\rho=0.96$

圖 40 原始 Lena 影像與移除組成小波樹之頻率係數 5 個 LSB 之影像對照圖 (a) 原始影像 (b) 移除頻率域係數中 5 個最低位元後之影像

實驗結果：

(a) Lena

Numbers of LSB		1	2	3	4	5	6	7	8	9	10	11	12
Results from D18-10	ρ	1	1	1	1	0.98	0.29	0.19					
	PSNR	38.50	38.36	37.58	35.54	32.80	29.30	26.31					
Results from D18-10(M)	ρ	1	1	1	0.99	0.96	0.88	0.77	0.72	0.70	0.65	0.64	0.64
	PSNR	38.69	38.71	38.09	36.16	32.90	29.30	26.27	24.11	22.99	22.34	22.32	22.32
Results from D9-7	ρ	1	1	1	1	0.98	0.38	0.21					
	PSNR	38.44	38.40	38.03	36.46	33.65	29.94	26.58					
Results from D9-7(M)	ρ	1	1	1	1	0.98	0.95	0.87	0.85	0.77	0.77	0.77	0.77
	PSNR	38.66	38.76	38.32	36.55	33.39	29.76	26.52	24.07	22.83	22.45	22.45	22.45
Results from D9-7-F	ρ	1	1	1	1	0.88	0.20						
	PSNR	38.39	38.27	37.62	35.84	33.20	29.72						
Results from [1]	ρ	1	1	0.99	0.52	0.11							
	PSNR	36.81	34.72	30.41	24.28	18.47							

(b) Goldhill

Numbers of LSB		1	2	3	4	5	6	7	8	9	10	11	12
Results from D18-10	ρ	1	1	1	1	0.98	0.29	0.05					
	PSNR	38.82	38.57	37.45	34.78	31.47	28.16	25.60					
Results from D18-10(M)	ρ	1	1	1	1	0.93	0.84	0.73	0.69	0.70	0.65	0.64	0.64
	PSNR	38.98	38.79	37.67	34.99	31.46	28.13	25.56	23.91	23.15	22.79	22.75	22.75
Results from D9-7	ρ	1	1	1	1	1	0.40	0.10					
	PSNR	38.87	38.74	37.98	35.59	32.22	28.74	25.90					

Results from D9-7(M)	ρ	1	1	1	1	0.97	0.93	0.84	0.80	0.72	0.73	0.73	0.73
	PSNR	39.00	38.92	38.05	35.53	31.97	28.61	25.81	23.81	22.66	22.49	22.49	22.49
Results from D9-7-F	ρ	1	1	1	1	0.84	0.27	-0.07					
	PSNR	38.80	38.58	37.65	35.17	31.87	28.61	25.87					
Results from [1]	ρ	1	1	0.97	0.38	0.14							
	PSNR	36.07	33.72	28.87	22.72	16.18							

(c) Peppers

Numbers of LSB		1	2	3	4	5	6	7	8	9	10	11	12
Results from D18-10	ρ	1	1	1	1	0.73	0.16						
	PSNR	39.96	39.69	38.72	36.40	33.38	29.85						
Results from D18-10(M)	ρ	1	1	1	0.99	0.94	0.86	0.77	0.76	0.71	0.70	0.70	0.70
	PSNR	40.14	39.98	38.90	36.60	33.42	29.80	26.57	24.26	22.68	22.20	22.15	22.15
Results from D9-7	ρ	1	1	1	1	0.98	0.15						
	PSNR	39.96	39.82	38.98	36.82	33.98	30.42						
Results from D9-7(M)	ρ	1	1	1	0.99	0.96	0.92	0.91	0.79	0.75	0.73	0.73	0.73
	PSNR	40.12	40.07	39.24	37.03	33.79	30.30	26.91	24.21	22.50	21.98	21.98	21.98
Results from D9-7-F	ρ	1	1	1	1	0.49	0.04						
	PSNR	39.86	39.61	38.74	36.65	33.66	30.23						
Results from [1]	ρ	0.99	0.96	0.90	0.64	0.14							
	PSNR	35.97	33.64	28.76	22.71	16.93							

表 13 以 Bitplane Removal 對影像攻擊後之實驗結果 (a)Lena (b)Goldhill (c)Peppers

由實驗結果可以看出，在Bitplane Removal攻擊實驗中，本論文所提方法之五組實驗結果均較 [1]之實驗結果為佳，當移除的LSB位元數達到 5 個時，[1]之方法即無法偵測

出浮水印，而本論文所提之五組方法均仍能偵測出浮水印，所測得之最低相關係數值為 0.49，且多數集中於 0.9 以上，故可知此五組方法確實具有較佳的抵抗 Bitplane Removal 攻擊之效能。

在改進前、後之浮水印方法抵抗 Bitplane Removal 攻擊之效能比較方面，未加入影像前置處理的原始浮水印演算法，在移除之 LSB 位元數達到 6 至 7 個以上時，即無法偵測出浮水印，然而加入影像切割重組之前置處理方法後，即使移除之 LSB 位元數達到 10 個以上，亦皆能偵測出浮水印的存在，故影像前置處理方法確實可增加浮水印抵抗 Bitplane Removal 攻擊之效能。

我們在進行 Bitplane Removal 攻擊實驗時，當移除之係數 LSB 位元數超過 10 以上時，該係數值幾乎已經完全被消除，即係數值趨近於 0，故即使再增加移除之 LSB 位元數，相關係數值與 PSNR 值亦不會再改變。此外在對「D18-10(M)」與「D9-7(M)」方法進行 Bitplane Removal 攻擊時，必須假設攻擊者不知此影像中的浮水印嵌入已經過影像切割重組之前置處理，故在實驗過程中，我們是以未經過前置處理的影像作小波轉換與位元平面移除，也因此當被移除的係數值減為 0 時，實際隱藏浮水印的小波係數仍能維持在經前置處理過的影像資訊中。

至於在不同差異化方向演算法抵抗 Bitplane Removal 攻擊之效能比較方面，「D9-7」與「D9-7-F」方法皆僅能夠抵抗至移除 6 個係數 LSB 位元，效能相差不多，然而就相關係數值的大小差異比較而言，「D9-7」方法之相關係數值略大於「D9-7-F」之方法，故可知隨機強制差異化方向之方法在抵抗 Bitplane Removal 攻擊之效能，不如原始浮水印方法之最適差異化方向演算法。

5.9 實驗結果整理比較

本節為實驗結果之整理比較，將本論文方法之五組實驗結果與 [1]之實驗結果進行對照，下表中符號”○”表示可偵測出浮水印，符號”X”則表示無法偵測出浮水印，由 Lena、Goldhill及Peppers三張影像之實驗結果綜合評定可看出，本論文所提出之方法確實能夠抵抗各種影像處理之攻擊，使浮水印之強韌性增加。

實驗攻擊種類與攻擊程度	D18-10	D18-10 (M)	D9-7	D9-7 (M)	D9-7- F	[1]
JPEG (壓縮係數=40)	○	○	○	○	○	○
JPEG (壓縮係數=30)	○	○	○	○	○	X
JPEG (壓縮係數=6)	○	○	○	○	X	X
JPEG (壓縮係數=3)	X	○	X	○	X	X
SPIHT (Bitrate=0.4)	○	○	○	○	○	○
SPIHT (Bitrate=0.3)	○	○	○	○	○	X
SPIHT (Bitrate=0.1)	○	○	○	○	X	X
Median Filter (Mask=4×4)	○	○	○	○	○	○
Median Filter (Mask=5×5)	○	○	○	○	○	X
Median Filter (Mask=10×10)	○	○	X	○	X	X
Median Filter (Mask=12×12)	X	○	X	○	X	X
Gaussian Filter	○	○	○	○	○	○
Sharpening	○	○	○	○	○	○
Pixel Shifting Type 1、Type 2 (6 像素位移)	○	○	○	○	X	○
Pixel Shifting Type 1、Type 2 (7 像素位移)	X	○	X	○	X	○
Pixel Shifting Type 1、Type 2 (10 像素位移)	X	○	X	○	X	X
Rotation and Scaling (旋轉角度=0.75 度)	○	○	○	○	○	○
Rotation and Scaling (旋轉角度=1.0 度)	○	○	X	○	X	X
Rotation and Scaling (旋轉角度=1.25 度)	X	○	X	○	X	X
Rotation and Scaling (旋轉角度=-0.75 度)	○	○	○	○	○	○
Rotation and Scaling (旋轉角度=-1.0 度)	X	○	○	○	X	X
Rotation and Scaling (旋轉角度=-1.25 度)	X	○	X	○	X	X

實驗攻擊種類與攻擊程度	D18-10	D18-10 (M)	D9-7	D9-7 (M)	D9-7- F	[1]
Multiple Watermarking (浮水印數目=3)	○	○	○	○	○	○
Multiple Watermarking (浮水印數目=4)	○	○	○	○	○	X
Multiple Watermarking (浮水印數目=16)	○	○	○	○	X	X
Multiple Watermarking (浮水印數目=512)	○	X	○	X	X	X
Bitplane Removal (移除的位元數=4)	○	○	○	○	○	○
Bitplane Removal (移除的位元數=5)	○	○	○	○	○	X
Bitplane Removal (移除的位元數=6)	○	○	○	○	X	X
Bitplane Removal (移除的位元數=7)	X	○	X	○	X	X

表 14 實驗結果之整理比較表



5.10 浮水印長度相關效益分析

由於本論文所提出之浮水印方法，是以小波樹為單位進行浮水印位元的嵌入，當原始影像大小為 512×512 像素時，其頻率域小波係數可組成 3072 棵小波樹，故可供嵌入的浮水印長度最大可達 3072 個位元值。

然而，當嵌入的浮水印長度增加時，勢必會因為小波係數改變量增加而導致影像品質的下降，若要提升影像品質，則浮水印的強韌性則可能因此降低，換句話說，浮水印長度、影像品質與浮水印強韌性三者之間無法同時獲得提升，這即是於 2.2.3 節中所述之浮水印特性需求的對立與矛盾。

表 15 即為不同浮水印長度所造成的影像品質降低之比較。當使用 Lena 影像進行浮水印嵌入時，若維持相同的差異化間距與亂數種子設定，僅將浮水印長度由 512 個位元值改為 3072 個位元值，則由表 15 之實驗結果可以看出，即使是不同的浮水印嵌入方法，其嵌入浮水印後之影像 PSNR 值均由約 38 降低至 30 左右，故可知於本論文所提出之浮水印方法中，即使將浮水印長度增加至 3072，在為維持浮水印強韌性而使用相同差異化間距的情況下，影像品質之 PSNR 值依舊可保持在 30 以上。

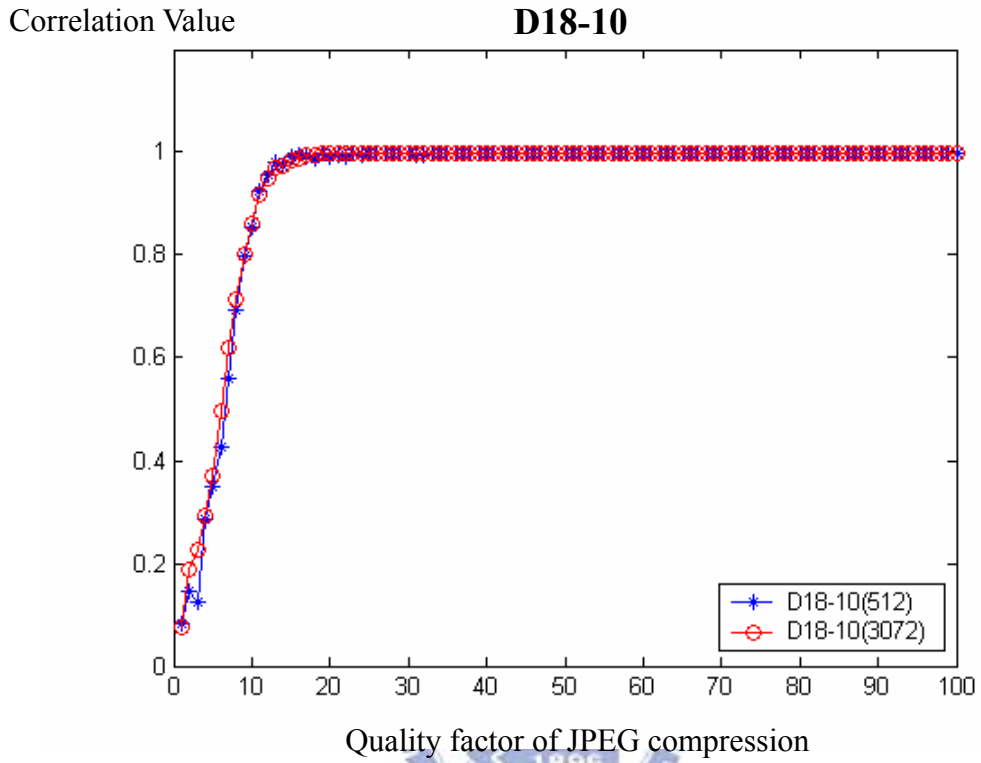
浮水印方法	D18-10		D9-7		D9-7-F	
	512	3072	512	3072	512	3072
影像品質 (PSNR)	38.37	30.51	38.32	30.41	38.33	30.00

表 15 不同浮水印長度所產生之影像品質比較表

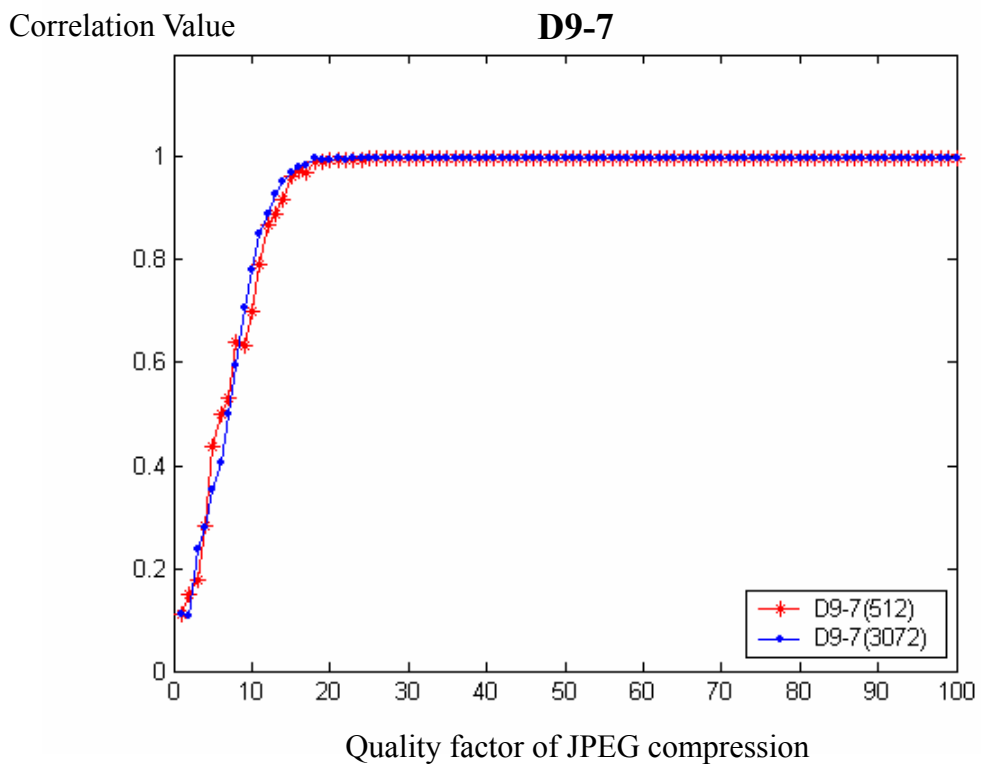
由於在本論文所提出之浮水印演算法中，浮水印的強韌性取決於係數能量的差異化間距大小，故在差異化間距相等的情況下，即使嵌入的浮水印長度不同，浮水印抵抗影像處理攻擊之能力卻相差不多。圖 41 即為使用不同的浮水印方法對 Lena 影像實施不同

長度之浮水印嵌入後，再以 JPEG 壓縮方法對影像進行攻擊實驗之結果。

(a)



(b)



(c)

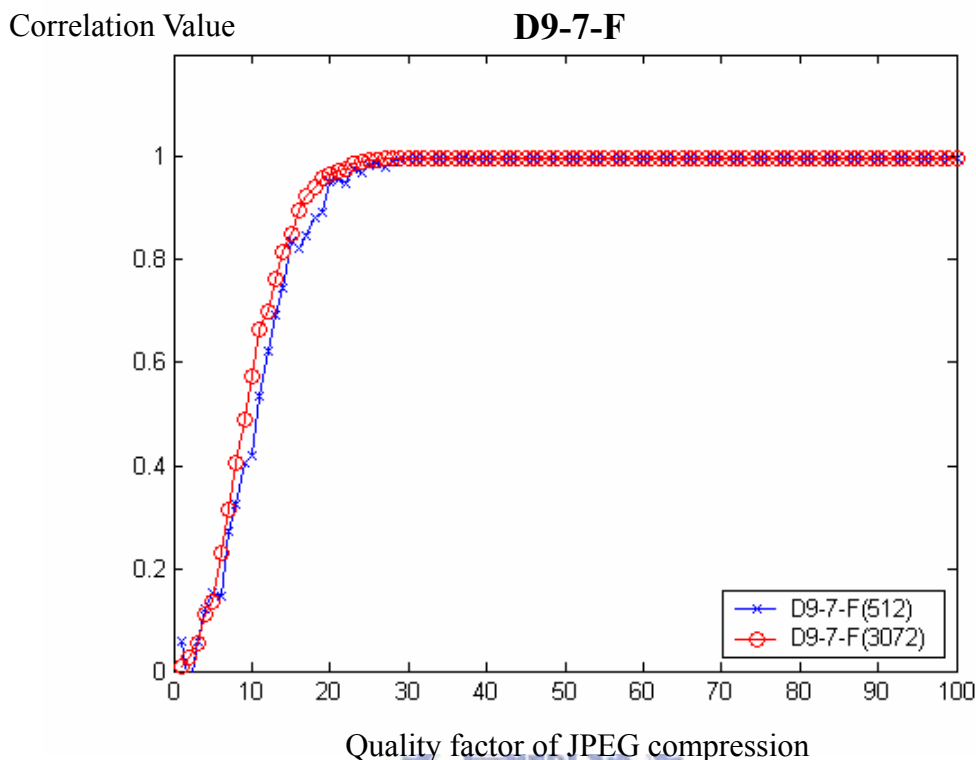


圖 41 對 Lena 影像嵌入不同長度浮水印後之 JPEG 壓縮攻擊結果 (a)D18-10 (b)D9-7 (c)D9-7-F

在圖 41 中，我們針對「D18-10」、「D9-7」與「D9-7-F」等三種浮水印方法進行 JPEG 影像壓縮攻擊實驗，在相同浮水印方法的實驗中，我們分別嵌入長度為 512 及 3072 位元值的浮水印，並維持相同大小的差異化間距。經驗證可得到影像品質會因為浮水印長度增加而降低，其差異即如表 15 所列，至於浮水印抵抗壓縮攻擊之強韌性則是相當接近，幾乎沒有什麼差別。

由上述實驗結果可知，當嵌入的浮水印長度增加時，會造成影像品質的降低，但卻不會影響浮水印的強韌性；浮水印之強韌性是決定於差異化間距的大小，倘若差異化間距增加，則浮水印之強韌性亦會增加，但影像品質自然會因係數改變量增加而下降，此結論在其他類型之影像浮水印攻擊實驗中亦是如此。

綜上所述，在本論文所提出之浮水印方法中，倘若欲嵌入之浮水印長度為最大之 3072 個位元值，則可以動態調降差異化間距，使嵌入浮水印後的影像品質維持在人類視覺可以接受的程度，雖然此舉不免將導致浮水印強韌性降低，但如何取得其中的平衡點，則需視使用者的目的與其較重視之部份而定。表 16 為上述浮水印長度、差異化間距、影像品質與浮水印強韌性之相關性列表。

浮水印長度	差異化間距		影像品質	浮水印強韌性
不變	增加	➡	下降	上升
不變	減少	➡	上升	下降
增加	不變	➡	下降	不變
減少	不變	➡	上升	不變

表 16 浮水印長度、差異化間距、影像品質與浮水印強韌性之相關性列表



第六章 結論與未來展望

6.1 結論

本篇論文是以小波轉換理論為基礎，透過四層小波轉換將影像由空間域轉至頻率域，再利用中頻係數組成之小波樹係數結構的差異，運用能量差異化方式將浮水印位元嵌入其中，並記錄小波樹第二、三階層之各組係數差異化方向及第四階層之 modular 值。當欲自影像中取出浮水印時，則依序讀取小波樹不同階層之各組係數，並藉由嵌入浮水印時所使用之差異化方向與 modular 值，解析各階層係數的能量差異結構，進而判斷出嵌入的浮水印位元值。俟浮水印取出之後，須再經過相關性浮水印偵測運算，取得一相關係數，以驗證浮水印之存在與否。

此外，本論文嘗試對提出的浮水印演算法之強韌性與安全性作進一步提升，其方法之一是運用影像切割重組之前置處理方式，將原始影像轉變為具馬賽克效果之新影像，藉以提高浮水印演算法的安全性與強韌性；方法之二是以隨機強制差異化方向演算法取代原本的最適差異化方向演算法，如此則不需要使用差異化方向記錄檔以記錄每組係數的差異化方向，可達到降低檔案保存與維護的風險。

本論文主要是先藉由說明參考文獻 [1]之演算法則，提出本論文所使用的浮水印產生機制與浮水印驗證程序，並分析該浮水印演算法在安全性方面的缺失。根據參考文獻 [28]運用密碼學角度對 [1]所進行的浮水印攻擊實驗，攻擊者確實可透過小波係數能量差異分析進而估算出係數量化程度，進而運用量化方式消除影像內已嵌入之浮水印。

有鑑於此，我們在第四章提出動態能量差異化之數位影像浮水印演算法，利用小波樹內部係數間的能量差異作為浮水印嵌入的基礎，並針對不同階層之小波係數進行動態能量差異化運算，此外我們並於演算法中加入防止攻擊者使用係數分析攻擊的機制，透

過亂數種子排列之順序，選取特定幾組係數嵌入浮水印位元，並將相反之浮水印位元嵌入另外幾組係數中，使攻擊者在不知亂數種子與差異化方向為何的情形下，無法對小波係數結構進行分析攻擊。

本論文更嘗試利用隨機亂數種子，將原始影像進行切割後再重新排列組合，使攻擊者無法得知進行小波轉換之新影像組成方式，並使對於原始影像造成之局部影像處理攻擊分散至新影像各部份，以提高浮水印之安全性與強韌性。此外，隨機強制差異化方向演算法雖然會造成影像品質的降低，進而影響到原本浮水印的強韌性，但卻提供了一個減少須額外保存之浮水印取出所需資訊量的方法，增加了浮水印的安全性。

由最後的實驗結果可以得知，本論文提供了一個更具安全性與強韌性的數位影像浮水印技術，能夠有效承受各種幾何空間或訊號處理之影像攻擊，達到保護已嵌入浮水印之目的，進而提供更有效的數位影像智財權保護技術，以保障數位影像創作者或合法使用者之權益。



6.2 未來展望

本論文中僅針對 9-7 及 18-10 filter 進行實驗效能測試，然則在小波轉換中，尚有其餘眾多的 filter 會產生不同屬性的小波係數，進而可能會影響到浮水印嵌入流程中的小波樹係數能量結構，故未來可以進行 filter 種類與特性之相關探討，並嘗試尋求更好的效能搭配。

此外，本論文是以四層小波轉換作為小波樹係數的產生機制，未來尚可探討五層以上小波轉換對於整體浮水印演算法所可能產生的影響，並研究是否有更佳的決定差異化方向之演算法。最後在原始影像切割重組方面，亦可尋求不同的子影像組合方式，亦或變更子影像的組合數量，以期得到更安全且更能抵抗影像處理攻擊的方法。

參考文獻

- [1] Shih-Hao Wang, Yuan-Pei Lin, “Wavelet Tree Quantization for Copyright Protection Watermarking”, IEEE Transactions on Image Processing, Vol. 13, No. 2, pp. 154-165, Feb 2004.
- [2] N. Bourbakits, C. Alexopoulos, “Picture Data Encryption Using Scan Patterns”, Pattern Recognition, Vol. 25, No. 6, pp. 567-581, 1992.
- [3] C. J. Kuo, “Novel Image Encryption Technique and Its Application in Progressive Transmission”, Journal of Electronic Imaging, Vol. 2, No. 4, pp. 345-351, 1993.
- [4] 張真誠、黃國峰、陳同孝，電子影像技術，旗標出版股份有限公司，台北，民國九十二年三月。
- [5] Murray, A. H., R. W. Burchfield (eds), The Oxford English dictionary: being a corrected re-issue, Oxford, England: Clarendon Press, 1933.
- [6] Stefan Katzenbeisser, Fabien A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Boston, 2000.
- [7] Jeffrey A. Bloom, Ingemar J. Cox, Ton Kalker, Jean-Paul M.G. Linnartz, Matthew L. Miller, C. Brendan S. Traw, “Copy Protection for DVD Video”, Proceedings of the IEEE, Vol. 87, No. 7, pp. 1267-1276, Jul 1999.
- [8] Jeng-Shyang Pan, Hsiang-Cheh Huang, Lakhmi C. Jain, Intelligent Watermarking Techniques, World Scientific, Singapore, 2004.
- [9] 鄭育仁、殷志揚、吳大鈞、蔡文祥，「數位博物館影像資訊的版權保護與認證」，新世紀數位圖書館與數位博物館趨勢研討會，VIII-1 頁~VIII-11 頁，新竹，民國九十年十一月一日。
- [10] M. S. Kankanhalli, K. R. Ramakrishnan, “Adaptive Visible Watermarking of Images”, IEEE International Conference on Multimedia Computing and System, Vol. 1, 1999.
- [11] S. P. Mohanty, K. R. Ramakrishnan, M. S. Kankanhalli, “A DCT Domain Visible

- Watermarking Technique for Images”, IEEE International Conference on Multimedia and Expo, Vol. 2, pp. 1029-1032, 2000.
- [12] <http://www.trl.ibm.com/projects/RightsManagement/datahiding/dhimg.htm>
- [13] http://www.ndap.org.tw/2_techreport/index.php?pid=228
- [14] 林承龍，「以均勻 Deadzone 為基礎之小波樹量化的數位影像浮水印」，國立交通大學資訊管理研究所，碩士論文，民國九十四年。
- [15] 羅銘耀，「植基於小波轉換之彩色影像浮水印技術」，國立東華大學資訊工程研究所，碩士論文，民國九十三年。
- [16] A. B. Watson, G. Y. Yang, A. Solomon, J. Villasenor, “Visibility of Wavelet Quantization Noise”, IEEE Transactions on Image Processing, Vol. 6, pp. 1164-1175, Aug 1997.
- [17] I. Hontsch, L. J. Karam, R. J. Safranek, “A Perceptually Tuned Embedded Zerotree Image Coder”, Proc. IEEE ICIP, Vol. 1, pp. 41-44, 1997.
- [18] Shih-Hao Wang, Yuan-Pei Lin, “Blind Watermarking Using Wavelet Tree Quantization”, IEEE International Conference on Multimedia and Expo, Vol. 1, pp. 589-592, Aug 2002.
- [19] M. Kutter, F. Jordan, F. Bossen, “Digital Watermarking of Color Images Using Amplitude Modulation”, Journal of Electronic Imaging, Vol. 7, No. 2, pp. 326-332, 1998.
- [20] M. S. Huang, C. C. Chang, K. F. Huang, “A Watermarking Technique Based on One-Way Hash Functions”, IEEE Transactions on Consumer Electronics, Vol. 45, No. 2, pp. 286-294, 1999.
- [21] R. Wolfgang, E. Delp, “A Watermark for Digital Images”, IEEE ICIP96, Vol. 3, Sep 1996.
- [22] G. Voyatzis, I. Pitas, “Chaotic Watermarks for Embedding in the Spatial Digital Image Domain”, IEEE ICIP98, Vol. 2, Oct 1998.
- [23] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoan, “Secure Spread Spectrum Watermarking for Multimedia”, IEEE Transactions on Image Processing, Vol. 6, No. 12, pp. 1673-1687, Dec 1997.

- [24] 單維彰，「凌波初步」，民 88，台北：全華科技圖書股份有限公司。
- [25] Peter J. Burt, Edward H. Adelson, "The Laplacian Pyramid as a Compact Image Code", IEEE Transactions on Communication 31, No. 4, pp. 532-540, 1983.
- [26] Marc Antonini, Michel Barlaud, Pierre Mathieu, Ingrid Daubechies, "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, Vol. 1, No. 2, pp. 205-220, Apr 1992.
- [27] A. Munteanu, J. Cornelis, G. V. D. Auwera, P. Cristea, "Wavelet Image Compression – The Quadtree Coding Approach", IEEE Transactions on Technology in Biomedicine, Vol. 3, No 3, pp. 176-185, Sep 1999.
- [28] Tanmoy Kanti Das, Subhamoy Maitra, "Cryptanalysis of Wavelet Tree Quantization Watermarking Scheme", IWDC 2004, LNCS 3326, pp. 219–230, 2004.
- [29] Min-Jen Tsai, Kuang-Yao Yu, Yi-Zhang Chen, "Joint Wavelet and Spatial Transformation for Digital Watermarking", IEEE Transactions on Consumer Electronics, Vol. 46, Issue 1, pp. 237, Feb 2000.
- [30] Biao-Bing Huang, Shao-Xian Tang, "A Contrast-Sensitive Visible Watermarking Scheme", IEEE Multimedia, Vol. 13, Issue 2, pp. 60-66, Apr-Jun 2006.
- [31] Zhao Dawei, Chen Guanrong, Liu Wenbo, "A Chaos-based Robust Wavelet-domain Watermarking Algorithm", Chaos, Solitons and Fractals 22, pp. 47-54, 2004.
- [32] 陳同孝、張真誠、黃國峰，數位影像處理技術，旗標出版股份有限公司，台北，民國九十二年五月。
- [33] Fabien A. P. Petitcolas, Ross J. Anderson, Markus G. Kuhn, "Attacks on Copyright Marking Systems", in David Aucsmith (Ed), Information Hiding, Second International Workshop, IH'98, Portland, Oregon, U.S.A., Apr 15-17, 1998.
- [34] Fabien A. P. Petitcolas, "Watermarking schemes evaluation", IEEE Signal Processing, Vol. 17, No. 5, pp. 58–64, Sep 2000.
- [35] <http://www.petitcolas.net/fabien/watermarking/stirmark/>

附錄 Filter Banks

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

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