Journal of Computer Science and Educational Technology
2015 Vol.5 No.1

運用複數個最低有效位元設計圖像資訊隱藏技術

Designing Image Information Hiding Techniques Based on Multiple Least Significant Bits

吳家豪　　陳永昇

國立臺北教育大學資訊科學系所

dama.wu@gmail.com　　yschen@tea.ntue.edu.tw

摘 要

隨著網路應用的日益普及，如何提高數位資訊於網路流傳分享時的安全性，變得受到普遍的重視，而圖像資訊隱藏是一個很有用的技術，過去在這方面已有很多相關的技術被提出，但綜觀各種空間域的圖像隱藏資訊技術，發現最低有效位元方法具有龐大的資訊嵌入容量，依舊是最為簡單實用的方式，無論隱藏資訊之嵌入或取出，其概念方法也較為簡捷易懂，同時亦近於數位創作者的操作習慣與模式。所以在此研究中，我們從應用面著手，提出索引位元的定序方法，來強化嵌入資訊的獨立性與完整性，藉以改進最低有效位元方法於資訊隱藏技術的運用外，並同時強化索引位元的排序功能，結合區塊切割，疊加透明遮罩的技術，以及增加最低有效位元值反轉或位移的機制，來增強資訊隱藏的隱密性與安全性，並發掘更便捷、更有效的資訊隱藏方法或技術。根據本研究所提出的方法，可以改善最低有效位元於資訊隱藏技術領域，強健性不足的問題。

關鍵字：資訊隱藏；最低有效位元；浮水印；智慧財產權

Abstract

As the applications of network become popular, the security problem of digital information transmission through networks has attracted a lot of attention from people around the world. Information hiding is useful for enhancing the security of digital image transmission via networks, and many information hiding techniques had been proposed in the literature. In this article, we reviewed various watermarking and image information hiding technologies and found that least significant bit is the most common and practical information hiding research area, not only proceed to improve the method of least significant bit and to enhance index-bit sequencing capabilities, but also combine the cutting block, XOR, shift and superimpose transparent mask technology to strengthen the information hiding privacy and security. Based on the proposed method in this study, that can improve the problem of lack of robustness of the least significant bit in the field of information hiding technology.

Keywords: Information Hiding, Least Significant Bit, Watermarking, Intellectual Property Rights

1. 前言

資訊隱藏(Information Hiding)通常是運用某些特定的模式或演算法，不讓除了接收者以外的任何人，知道傳遞事件之訊息或內容，所使用的一種技術方法。在網路的世界，我們可以利用各種數位媒體為媒介，如影像、文字或可執行檔等，將秘密訊息嵌入其中，讓人們不易察覺，以達到保護機密資訊的目的。

資訊隱藏為何如此被重視？我們可從新聞媒體的報導中，常常聽到「某某明星、偶像的私密照流出」，或者是「某參賽得獎的作品，因被發現是抄襲盜用他人的創作而取消資格」，這再再都顯露了網路的不安全性，而資訊隱藏技術的研究與開發，不外乎是為了保護個人、團體、企業和國家的隱私或機密，不被有心人士窺探或竊取。但也因資訊隱藏技術的研發，導致恐怖攻擊事件接連地發生，造成眾多無辜性命的喪生，較為人所知曉的事件如：於 1998 年東非兩座美國大使館爆炸案和 2001 年 911 雙子星摩天大樓的恐怖攻擊事件中，蓋達組織首腦賓拉登與幹部成員們，便是運用資訊隱藏的技術方法，將恐怖行動的訊息，隱匿在網路聊天室、電子郵件、拍賣或色情網站的照片裡，來進行對美國政府的制裁和報復。所以為了預防並遏止潛在性恐怖攻擊事件的再度發生，許多政府和企業，已開始明文禁止或限制各種加密技術的出口。不過此舉反倒促使各種不同類型的資訊隱藏技術與方法，在網際網路上蓬勃發展、推層出新。

身為數位創作者，既無意剽竊別人的創見，也不樂見自己驕傲的作品，被恣意地盜用。目前網路上最普遍被廣泛應用的自我保護機制，莫過於利用類似浮水印(Watermarking)、特殊符號(Symbol)或自己專屬的簽名(Signature)等等的標誌，如圖 1，嵌入於數位作品的機制；甚至破壞作品的完整性，將其裁邊截圖來留存證明，如圖 2；或者以數位作品壓縮設密碼的方式，來宣示和保護創作者的數位媒體著作權或保障合法使用者的使用權。無論是明顯可見或似有若無，這都是一種資訊隱藏技術的運用；而有關於 1998 年東非兩座美國使館爆炸案和 2001 年 911 恐怖攻擊事件，蓋達組織首腦賓拉登與幹部成員們，傳遞訊息的模式，其使用的方法，也是屬於資訊隱藏技術的範疇。

圖1. 浮水印/符號/簽名

圖2. 裁邊截圖

在電腦網路通訊蓬勃發展和數位設備使用普及化，個人 SOHO 族、工作室崛起的今日，各類作品的數位化，已經是非常的普遍。加上諸如 Facebook、Twitter、Blog 等社交平台，受到人們普遍的使用，及各種具備第三方支付的交易平台應運而生，數位化作品或商品的流通與交易，必定會更加地頻繁。但也因為數位資訊在網路傳播的便利，衍生而來的是創作者的作品，恐被有心人士未經授權的篡改與複製，引發偽造、侵權等法律糾紛。這不僅對創作者的權益造成損失，也間接影響創作者往後創作與分享的意願。所以要如何於作品中，嵌入屬於創作者的個人資訊，

來證明並保障創作者有其作品所有權，以捍衛創作者智慧財產權的方法，便成為近年來非常重要且熱門的研究課題。

在圖像上進行資訊隱藏技術的應用，最早是由 Naor & Shamir[1]於 1995 年在柏林舉行的 Eurocrypt Conference 中，所提出「Visual Cryptography」視覺密碼的資訊保護機制，其目的就是為了保護機密或私密影像，利用人類視覺對圖像辨識解讀的能力，將機密影像分解成數張沒有條理的分享影像(Shares)，於解密時再將這些分享影像進行疊合辨識，其解密過程無需要龐大複雜的運算，是此理論之主要優點。於 1999 年時，Petitcolas[2]等學者也將資訊隱藏技術進行歸類，劃分為隱蔽通道(Covert channels)、隱寫術(Steganography)[6][7][8][9]、匿名(Anonymity)和版權標誌(Copyright marking)[10][11][12][13][14]四大類。但由於為了提昇資訊隱藏技術的隱密性與可靠性，各種方法技術組合的相關研究，漸漸地讓資訊隱藏四大分類的分際，變得不再那麼明確重要了。

資訊隱藏的技術一般而言，就同許多如 DES[3]、RSA[4]或 AES[5]等資料加密的方法一樣，對需要受保護的物件或資料，運用類似保險箱和鎖鑰的概念，使其達到不被竊取或破壞的目的。為了確實保護機密資料的安全，學者們紛紛提出各種不同方法與技術，來強化資訊隱藏技術的強健性。而與視覺密碼相關，圖像上資訊隱藏技術的應用，依照 Petitcolas[2]等學者的分類方式，大致可歸分於隱寫術[6][7][8][9]和版權標誌[10][11][12][13][14]兩個部分內。其中所應用的相關技術，多以最低有效位元(Least Significant Bit, LSB)、向量量化(Vector Quantization, VQ)、擴頻展頻(Spread Spectrum, SS)、離散餘弦轉換(Discrete Cosine Transform, DCT)、離散小波轉換(Discrete Wavelet Transform, DWT)、離散傅立葉轉換(Discrete Fourier Transform, DFT)或霍夫曼編碼(Huffman Coding)等方法為基礎，結合密碼學(Cryptography)資料加密的方法，進行資訊隱藏技術的改進與提升。

為了讓數位作品的智慧財產權受到保障，也希望資訊隱藏的技術，能普遍應用於日常生活中，創作者若能自行對其數位作品，進行資訊隱藏技術的保護與認證，這對那些網路分享的數位作品，恣意篡改、複製的盜用份子，肯定能產生遏阻的作用，於引發偽造、侵權等法律糾紛時，也能提出自己是作品所有權人的證明機制。在視覺密碼的領域，於空間域(Spatial Domain)中嵌入機密資訊，具有快速運算和較大的資料嵌入量的優點，是最簡單且容易上手的做法，僅需使用圖像編輯軟體和試算表，人人皆可完成資訊隱藏的目的，不須龐大複雜的公式或演算法的運算，但其最大的缺點在於其隱藏資訊的強健度，很難抵抗幾何攻擊及一般訊號處理造成的破壞，如下表 1 之比較。

表1. 各種圖像資訊隱藏技術之優劣表

	最低有效位元	向量量化編碼	展頻擴頻	離散餘弦轉換	離散小波轉換	離散傅立葉轉換
不可察覺性	佳					
不可偵測性						
不可移除性						
強健性	弱	弱	弱	強	強	強
明確性	尚可					
容量	大	次小	小	小	小	小
是否需要額外來源	不一定	需要	需要	不需要	不需要	不需要
有效率						

在本文中將以最低有效位元(Least Significant Bit, LSB)為基礎技術，針對空間域資訊隱藏強健性(Robustness)的不足加以改進，在盡量不影響或改變到數位媒體的原貌下，提出不同的資訊隱藏技術模型與研究方向，來強化並增進對數位化作品之智慧財產權的實質應用性。之所以會採用 LSB 最低有效位元方法於圖像上資訊隱藏之研究，最主要的原因在於其龐大的資訊嵌入容量，以 1024×768=786432 個像素點的彩色圖像為例，就有包含 RGB 三色共 786432×3=2359296 bytes 的容量空間可供資訊隱藏之使用，其可運用彈性空間之大，非其他資訊隱藏技術或方法所能相比，若以 ASCII Code 來完成資訊隱藏的編碼，光是單單使用最低有效位元，也都還有 2359296/8=294912 個字元(Character)數的容量。就如本論文之英文摘要，含關鍵字共 157 個字，不計算空白間隔則有 930 字元數，若連同空白間隔一併計算，總共也才 1085 字元數而已。

要實現其實務上應用的可行性與便利性，綜觀各種圖像隱藏資訊，發現最低有效位元方法，於資訊隱藏空間域的運用，依舊最為普遍，無論隱藏資訊之嵌入或取出，其概念方法也較為簡捷易懂，同時亦近於數位創作者的操作習慣與模式。所以在本論文中，我們將從應用面著手，提出索引位元的定序方法，來強化嵌入資訊的獨立性與完整性，藉以改進最低有效位元方法於資訊隱藏技術的運用外，同時強化索引位元的排序功能，結合區塊切割，疊加透明遮罩的技術，並增加最低有效位元值反轉或位移的機制，來增強資訊隱藏的隱密性與安全性，以發掘更便捷、更有效的資訊隱藏方法或技術，來維護創作者之數位作品的著作權，並提高資訊隱藏技術於實務上應用的便利性。

2. 相關基礎技術與原理

2.1. 彩色透明圖像疊合之 RGB 運算公式

透過圖像的透明化，我們才能利用肉眼去看到色彩疊合時的變化。所以當我們於基底背景圖上，疊加一張具有透明度的圖像，我們藉由其越高的透明度，即越低的 α 值，才能越清楚看見背景圖的樣貌。在此完成圖以符號 *C* 表示，背景圖為 *B*，遮罩圖是 *A*，透明度 $T = (1-\alpha)$。假若透明度是 *70%*的話，我們所能看見之完成圖 *C* 的值，就是 *70%*未被遮蔽之背景圖 *B* 值，加上遮罩圖 *A* 所殘存 *30%*值的加總, 則其疊加前後 RGB 值與 *α* 值間之關係如下式(1)。

$$C_{rgb} = B_{rgb} \times (1-\alpha) + A_{rgb} \times \alpha \tag{1}$$

但是計算機對乘法與除法的運算速度是慢的，尤其是在對多像素點同時計算時，就會顯得相當費時且沒有效率，這裡將對現有式(1)稍作調整，以減少乘法的運算次數。但是色彩疊合的特性是平移漸進的，於疊合後之 RGB 值，也會落在兩相互疊合像素點 RGB 值的中間，但會產生加減的判斷問題。如下式(2)，當 $B_{rgb} < A_{rgb}$ 時，採用加法，$B_{rgb} > A_{rgb}$ 時，則採用減法；而下式(3)的加減計算剛好相反，當 $A_{rgb} < B_{rgb}$ 時，採用加法，$A_{rgb} > B_{rgb}$ 時，才採用減法。雖然式(2)、式(3)都存在有值大值小的判斷問題，但相較於式(1)，面對龐大像素點 RGB 值之運算時，其計算機運算的速度便相對快多了。

$$C_{rgb} = B_{rgb} \pm \left(\left| B_{rgb} - A_{rgb} \right| \times \alpha \right) \quad (2)$$

$$C_{rgb} = A_{rgb} \pm \left(\left| B_{rgb} - A_{rgb} \right| \times \left(1 - \alpha \right) \right) \quad (3)$$

在資訊隱藏技術的操作，也會有兩透明圖像或多圖相疊合的運用，其色彩 RGB 值的計算，可藉由上面的式子來推導。我們已知 α 值是透明度的相反值，當經過兩次不同 *T* 值的遮罩圖時，其 *T* 值會以乘積方式降低，這就好比透過兩副太陽眼鏡看世界，透明度變差變暗了。在此完成圖以符號 *F* 表示，以背景圖 *B* 為基底圖：疊加二遮罩圖 *A*1、*A*2 而成之疊合圖以符號 *C*12 表示，其透明度 *T* 值為 *T*1、*T*2，分別對應 *α12*、*α1* 和 *α2*，公式推導如下：

因為 $T = T1 \times T2, \quad T1 = (1-\alpha 1), \quad T2 = (1-\alpha 2)$

所以 $\alpha 12 = 1 - T = 1 - (1-\alpha 1) \times (1-\alpha 2) = \alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2$ (4)

代入已知式(1)，進行兩次疊加，得

$$F_{rgb} = \left(B_{rgb} \times \left(1 - \alpha 1 \right) + A1_{rgb} \times \alpha 1 \right) \times \left(1 - \alpha 2 \right) + A2_{rgb} \times \alpha 2 \quad (5)$$

利用二疊加遮罩圖之 α 值公式(4)，代入式(1)，得

$$F_{rgb} = B_{rgb} \times \left(1 - \left(\alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2 \right) \right) + C12_{rgb} \times \left(\alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2 \right) \quad (6)$$

合併式(5)、式(6)運算，

$$\left(B_{rgb} \times \left(1 - \alpha 1 \right) + A1_{rgb} \times \alpha 1 \right) \times \left(1 - \alpha 2 \right) + A2_{rgb} \times \alpha 2$$
$$= B_{rgb} \times \left(1 - \left(\alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2 \right) \right) + C12_{rgb} \times \left(\alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2 \right)$$

整理，得 *A*1、*A*2 二透明遮罩圖之疊加公式(7)

$$C12_{rgb} = \frac{A1_{rgb} \times \alpha 1 \times \left(1 - \alpha 2 \right) + A2 \times \alpha 2}{\alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2} \quad (7)$$

2.2. 各像素點間 RGB 值等比例變化之特性

雖然色彩間的作用有許多不同的運作方式，不過其基本原理是相通的，無論是單張或多張透明圖的疊加方式，均可於圖像之資訊隱藏時靈活運用，因為每個像素點的 RGB 值都有 0~255 的色彩變化。除了利用如上列公式之色彩疊合計算公式進行運算外，在研究過程中，看似雖沒阿拉伯數字同等累加的特性，但從實驗過程中所產生的數值發現證實，在其圖像色彩經疊合前後，其像素點 RGB 值的顏色變化，彼此間確實存在著等比例的關係。同時發現到甚至是於不同的像素點間，其 RGB 值於色彩疊加前後，似乎也有著同樣等比例變化的特性。由公式(1)我們清楚知道，

疊加透明圖的色彩變化，是介於兩圖之間，端視透明度來決定其變化之配比。但圖像於相疊加後，其每個像素點的 RGB 值，是否都是依同一比例來進行顏色的改變？

我們就以 Pablo Picasso 的名畫為樣本，以透明度 *30%*的樣本圖和透明度 *70%*之遮罩圖，二透明圖疊加的方式來進行驗證。為了使實驗過程和說明更為清楚明瞭，我們將這張名畫縮小為像素點 40×50 如圖 3 的圖片，以方便針對圖像特定部位進行取樣，在圖上隨意找二像素點(8, 2)和(9, 35)如圖 4，來供做此次實驗的數據分析。

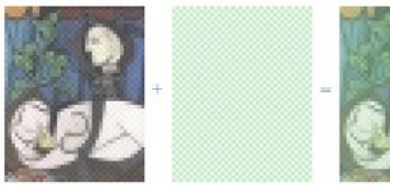


圖3. 40×50像素之疊合圖像

圖4. 二取樣之像素點放大圖

圖像上像素點(8, 2)和(9, 35)疊合前後之 RGB 值，可以運用上述公式(7)進行計算，或者使用圖像編輯軟體色彩工具直接查詢，其 RGB 數值都是完全相同的。其表示之符號分別為樣本圖 *P*，疊加之透明遮罩圖 *A*，疊合完成圖 *C*；而二圖間各 RGB 數值的差距值，則分別以|*P*－*A*|、|*P*－*C*|和|*A*－*C*|表示，數值記錄和比較如下表 2 與表 3。

表2. 二像素點疊和前後RGB數值

	樣本圖 *P*	透明遮罩圖 *A*	疊合完成圖 *C*
(8,2)	P_{rgb} (47, 86, 115)	A_{rgb} (169, 248, 173)	C_{rgb} (93, 148, 137)
(9,35)	P_{rgb} (28, 19, 22)		C_{rgb} (82, 106, 79)

表3. 二圖間像素點RGB數值差距比較

	(8,2)			(9,35)		
	R	G	B	R	G	B
|*P*－*A*|	122	162	58	141	229	151
|*P*－*C*|	46	62	22	54	87	57
|*A*－*C*|	76	100	36	87	142	94

由觀察表 3 數值之變化結果，排除計算時所產生之些微誤差，可歸納推算出：無論是像素點(8, 2)或像素點(9, 35)，於疊加透明圖前後，RGB 色彩值的偏移量，是以相同的比例來調整。有其同步和等比例的特性。也就是說圖像上任何一像素點，在疊加透明圖像後，其 RGB 值都會產生同等比例的變化，是同步一致的。所以根據此一特性，也可運用於與色彩有關的各種資訊隱藏技術，並可結合最低有效位元方法的運用，以發掘找尋色彩圖像所要傳達的訊息與意義。

2.3. 最低有效位元(Least Significant Bit, LSB)之運用

最低有效位元是屬於在空間域資訊隱藏的一種技術，也是一個資訊隱藏簡單概念的運用。其將像素值以二進制的方式表示，並配合美國資訊交換標準碼(American Standard Code for Information Interchange, ASCII)，把秘密資訊分解成最小單位 Bit，然後將這些 Bit 置入取代每個像素值最後一個 Bit 的方法，如下圖 5。圖像在藏入秘密資訊後，每個像素點的數值改變範圍僅有正負 1 的差別，以人類肉眼是很難感受有任何明顯變化，因此能夠確保一定的影像品質。

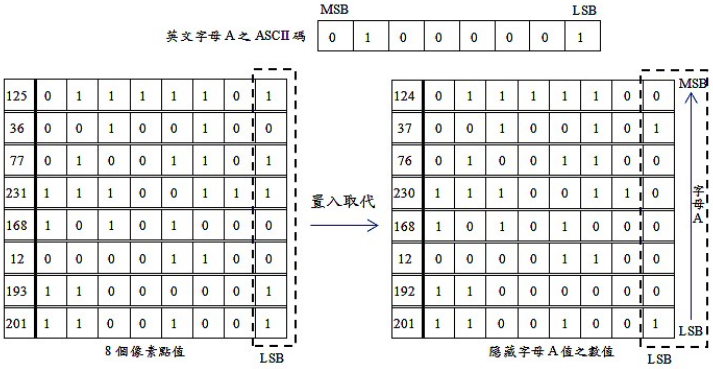


圖5. 隱藏A字母訊息於8個像素點內之表示圖

彩色點陣圖像是由眾多個像素點所構成，而每個像素點分別有 RGB 三個像素值，每個像素值的範圍是從 0~255，共佔 8 位元空間，並將其複數個最低有效位元定義為資訊位元和索引位元。於最低有效位元處嵌入機密資訊的位置，所指的就是資訊位元；再於更高有效位元處置入索引資訊的位置，也就是指索引位元，如圖 6 所示。我們將 8 位元空間的最低 3 個位元空間，分別定義為資訊位元和索引位元。在圖像每個像素點 RGB 的 3 個像素值，可分別嵌入 3 個字元資訊，資訊位元可位於每個字元中的 1 個或 2 個最低有效位元，而索引位元是用來表示 0~7 之索引值，為高於資訊位元的 1 個或 2 個有效位元。

在實際應用上，這 3 個最低有效位元空間，是索引位元與資訊位元的對應組合，其功用是於資訊隱藏操作時，可做有效之配置。例如：當嵌入資訊量較小時，可僅利用 1 個資訊位元和 1 個索引位元來完成；反之，當嵌入資訊量較大時，則可運用 1 個資訊位元和 2 個索引位元或 2 個資訊位元和 1 個索引位元，藉此以增加最低有效位元 LSB 方法於實際應用時的彈性。

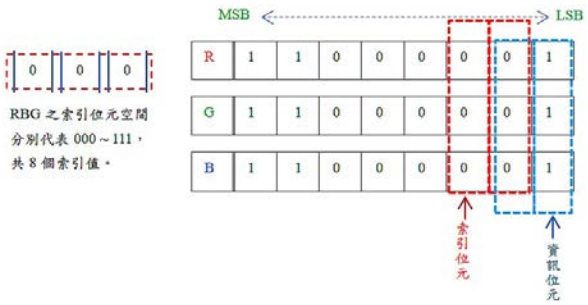


圖6. 資訊位元和索引位元之配置

於最低有效位元技術中增加索引位元的目的，主要是為了方便快速還原嵌入資訊，同時也可藉此索引訊息，來確認嵌入資訊還原時的完整性，並強化嵌入資訊還原的獨立性。通常最低有效位元於每個 8 位元數值中，最多只會運用到最低的 3 個位元空間，來置入隱藏資訊，因為越高位元的修改，對原圖像的破壞也會越高，為避免原圖失真，這裡也以不超過最低的 3 個位元空間為研究方向，完成資訊隱藏技術的改進。

彩色圖像的像素點分別有 RGB 三個像素值，單僅使用最低有效位元空間，每 8 個像素點就有 3×8 = 24 bits，就可做為 3 個字元訊息的隱藏空間。所以以 1024×768 像素的圖像為例，有 786432 個像素點，便可藏入 294912 個字元的資訊量；若改使用最低 2 個位元空間，則有 589824 個字元的資訊量。加入索引訊息來避免如此龐大的資訊量，因圖像遭到編輯剪裁而導致隱藏資訊流失的風險，也是其用意之一。因為索引位元在每個像素點可以有 8 個索引值，分別對應字元的 8 個位元。在索引值不重複的前提下，以 1 個索引位元進行資訊隱藏來計算，可完成 3 個字元的資訊隱藏，需要 8 個像素點空間。若是運用 2 個索引位元，則其可藏入高達 3×8 = 24 個字元數，需要 8×8 = 64 個像素點空間。

在此為了解說方便，我們運用 1 個索引位元和 1 個資訊位元，以 8 個像素點隱藏「God」為例，來進行分析說明。「God」的 ASCII 碼值分別為 G (71) = 01000111, o (111) =01101111, d (100) = 01100100，因為橫式之一般習慣是由左到右，所以我們也將 8 Bits 之 MSB 至 LSB 依序對應索引位元 000~111，其運作之方法模型，可以參考圖 7，索引值 0~索引值 7 之欄位，分別表示被嵌入資訊之 8 個像素點 RGB 值的 2 個最低有效位元值，其分別為 1 個資訊位元值和 1 個索引位元值，藉由索引位元值 0~7 的順序，依序將資訊位元值由 MSB 到 LSB 排序組合，來完成共計 3 個字元訊息的資訊隱藏。

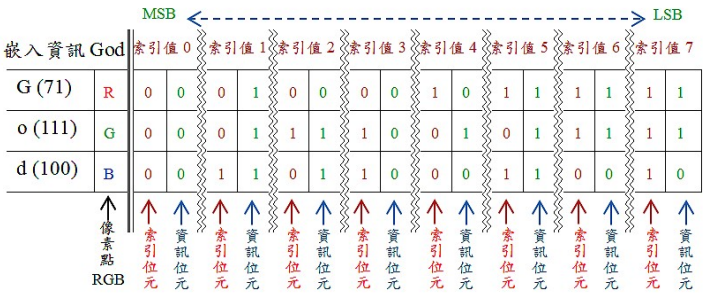


圖7. 以8個像素點隱藏「God」資訊為例之索引型LSB方法

圖 7 是以隱藏文字訊息為例，若要隱藏圖像資訊或不可視浮水印(Imperceptible Watermarking or Invisible Watermarking)，也可運用此操作原理和方法。以單一索引位元和資訊位元空間，實現嵌入彩色浮水印，則每個浮水印圖的像素點，需佔用隱藏資訊圖 8 個像素點空間；而灰階浮水印，則僅需佔用隱藏資訊圖 8/3 個像素點空間。單單 1024×768 像素的圖像，就有 786432 個像素點，可隱藏資訊之大；其空間切割配置的使用方式，或同時嵌入文字和圖像資訊，均可依需求編排，非常多元，於後會針對與索引相關之方法做進一步說明，在此便不多贅述。

2.4. 視覺密碼(Visual Cryptography)概念衍生之應用

關於如何於圖像中隱藏所要傳遞的資料訊息,從視覺密碼(Visual Cryptography)直接使用人類肉眼直觀式解讀辨識的概念出發，以及來自於圖片編輯軟體運用的經驗，所應運而生的想法。既然我們都能運用各種計算方式，對像素點進行疊加、柔光、屏幕、濾色等模式，來改變像素點原始的顏色，那想必以像素點的 RGB 值做為隱藏資訊的密碼，並直接進行色彩 RGB 值的編排是可行的。這與 Naor & Shamir[1]所提出之「視覺密碼」概念，有不盡相同之處，他們是將機密影像分解成多張看似雜亂無章的分享影像(Shares)，以達到保護資訊的機制，特別是對私密圖像的保護。但其「看似雜亂無章的分享影像」有時反更啟人疑竇，讓人對這些影像心生懷疑。假使這些雜亂無章的分享影像，是一張張再普通不過的生活照或風景照，那是不是更能提高隱藏資訊的安全性，及降低被他人窺探解密的風險呢？當然其隱藏資訊之傳遞的安全性與便利性，與圖片疊合的張數和方式，以及圖片像素的大小之複雜度有著密切的關係。以下是我們的方法與實驗。

其設計概念採用的方式，大致可分為：(a)素底素字 (b)素底花字 (c)花底素字等數種。在此會運用到 Alpha 值之疊合運算公式，以素底素字的方式進行一個簡單實驗，來試驗其可行性，圖示說明如下：

圖8. 數張風景圖或生活照+Key圖

我們就以圖 8 之圖檔為實驗樣本，運用六張普通的風景圖片與一張 Key 圖，也就是編排隱藏資料訊息的關鍵解碼圖，當然也可以透過多張 Key 圖，來加強隱藏資訊的安全性。在此採用素底素字的編輯方式，配合使用圖像編輯軟體和試算表，來試作疊加合成解析被隱藏之資訊，產生所要傳遞的秘密圖檔訊息。為了讓圖像編輯軟體和試算表的運用較為簡捷快速，在不影響實驗結果的情形下，將圖檔全部調整為解析度 40×30 Pixel 之樣本，並以 *70%*的透明度(α 值=0.3)進行編輯，以方便觀察其圖像疊合時顏色的變化情形。

另外也必須利用到試算表的運算，設計出一張能將疊合之圖，全部變為同色素面的 Key 圖。我們可運用公式(1)的反轉運算，將 Key 圖的 RGB 值計算出來，例如下面表 4 中的公式列 *fx*=(125-(Ps_RGB!E11*0.7))/0.3，就是為了要將所有圖於疊合後，其 RGB 值均變為 rgb (125, 125, 125)之灰色圖而計算;公式列中之 Ps_RGB 是為取自六圖疊合圖 RGB 試算表儲存格之值(如表 5)。

因為實驗樣本為解析度 40×30 Pixel 之圖像，其中以表 4 和表 5 中的 E 之 2G 行之 E11 儲存格為例，E11 儲存格值即是圖像中第二行第十列像素點 RGB 值中之 G 的數值。當然運用人力之簡單計算，無法找到最佳之 RGB 數值，所以在反轉推算的結果，可能會產生負值或大於 255 的值，這時就要將這些 RGB 值進行調整，其結果就會如圖 9 所示，無法達到完全一模一樣的顏色。

表4. 運用公式(1)反轉運算之Key圖RGB數值

E11 =(125-(Ps_RGB!E11*0.7))\0.3

	A	B	C	D	E	F	G	H	I	J	K	L
1	1R	1G	1B	2R	2G	2B	3R	3G	3B	4R	4G	4B
2	174	132	29	165	125	27	155	127	29	125	106	20
3	181	139	39	165	130	32	153	123	32	125	109	25
4	167	134	34	167	137	41	127	106	20	120	109	32
5	139	111	18	141	116	27	120	102	20	106	97	20
6	125	104	22	118	97	18	118	102	22	116	104	32
7	92	76	4	102	85	15	109	95	22	90	83	15
8	95	83	20	92	78	20	60	53	-6	113	113	57
9	104	90	36	109	97	46	81	76	25	85	85	34
10	90	81	29	120	111	64	116	113	62	69	69	20
11	69	60	8	99	92	43	95	92	46	90	90	46
12	104	90	41	88	78	29	92	95	53	125	130	90
13	174	165	113	172	165	123	176	183	148	120	132	106
14	204	190	155	188	183	153	186	195	176	158	174	162
15	195	186	158	209	209	188	193	204	190	113	134	130
16	202	195	176	193	190	176	160	172	160	118	132	130
17	223	221	207	221	223	207	155	167	160	137	151	146
18	190	195	181	167	176	162	130	148	139	127	144	137
19	167	181	169	158	172	165	153	176	169	116	137	130
20	183	207	209	186	207	211	158	179	181	132	151	148
21	155	179	188	188	209	218	214	237	246	216	232	239

表5. 六圖疊合圖Ps_RGB試算表之RGB數值

E11 139

	A	B	C	D	E	F	G	H	I	J	K	L
1	1R	1G	1B	2R	2G	2B	3R	3G	3B	4R	4G	4B
2	104	122	166	108	125	167	112	124	166	125	133	170
3	101	119	162	108	123	165	113	126	165	125	132	168
4	107	121	164	107	120	161	124	133	170	127	132	165
5	119	131	171	118	129	167	127	135	170	133	137	170
6	125	134	169	128	137	171	128	135	169	129	134	165
7	139	146	177	135	142	172	132	138	169	140	143	172
8	138	143	170	139	145	170	153	156	181	130	130	154
9	134	140	163	132	137	159	144	146	168	142	142	164
10	140	144	166	127	131	151	129	130	152	149	149	170
11	149	153	175	136	139	160	138	139	159	140	140	159
12	134	140	161	141	145	166	139	138	156	125	123	140
13	104	108	130	105	108	126	103	100	115	127	122	133
14	91	97	112	98	100	113	99	95	103	111	104	109
15	95	99	111	89	89	98	96	91	97	130	121	123
16	92	95	103	96	97	103	110	105	110	128	122	123
17	83	84	90	84	83	90	112	107	110	120	114	116
18	97	95	101	107	103	109	123	115	119	124	117	120
19	107	101	106	111	105	108	113	103	106	129	120	123
20	100	90	89	99	90	88	111	102	101	122	114	115
21	112	102	98	98	89	85	87	77	73	86	79	76

我們可以從圖 9 之流程中，清楚的看出其圖像疊加的結果。A 圖顯示疊合一張未隱藏任何訊息之 Key 圖，其每一像素點的 RGB 值已幾近相同，若要使其值完全一樣，僅需再多疊加一張 Key 圖便可完成。而 B 圖則是與一張加入藏有「S」訊息之 Key 圖，所疊合後的結果。此概念最大的關鍵就在於 Key 圖上像素點的色彩編排，也是這方法唯一較有困難度的地方，若要完美精準呈現隱藏之訊息，仍然必需仰賴計算機方程式和演算法的運算，或使用高階的圖像編輯工具，才較為容易順利完成設計。這也是直觀式視覺資訊解讀，亟需繼續努力研究的方向。

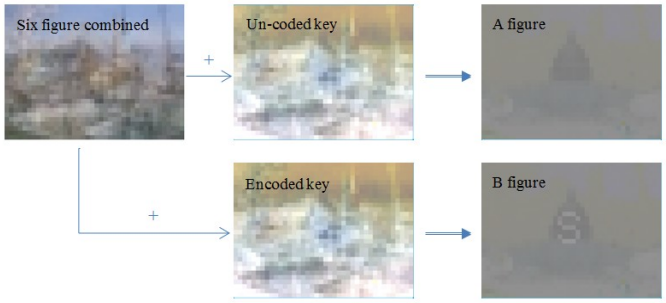


圖9. 六圖疊合圖+Key圖=解讀訊息之示範流程圖

3. 運用最低有效位元進行資訊隱藏

將索引功能做為資訊隱藏技術輔助的方法，無論在空間域或頻率域的應用，都有眾多學者發表各種不同的理論與研究。這裡我以數位創作者的角度出發，提出幾項便於他們使用的方法與技術，其思維概念雖然簡單，卻極具實用性與便利性，且提出之各種概念方法，均富有靈活組合之應用性。通常只要是懂得如何使用圖像編輯軟體的創作者，就可應用於對自己數位作品的保護，其技術原理之應用分述如下。

3.1. 區塊密集之最低有效位元索引方法

此法運用 2.3 節所介紹的原理，以 1024×768 像素的圖像並以 2 個索引位元，可隱藏 24 個字元空間為例，有 786432 個像素點，最多可切割成 786432/64 = 12288 個 8×8 的空間區塊；也就是說以這 24 個字元密碼，可在 1024×768 像素的圖像中重複嵌入 12288 次。

其操作模式有二，一是連續區塊密集嵌入方式，所以只要在圖像上某個大於 8×8 像素的區塊，就能還原秘密資訊。另一則是以相同之單一索引區塊密集的嵌入方式，嵌入之空間區塊或密集度可大可小，若以 8×8 的像素空間單位為例，64 個索引值可於 12288 個 8×8 的空間區塊內，重複嵌入 192 次。將 RGB 值轉以二進制方式表示，就能清楚搜尋找出各個索引的位置與順序，便可得知隱藏之 24 字元。因圖像的資訊隱藏空間夠大，無須擔心圖像因裁剪，而造成機密資訊的流失，其空間配置方式如圖 10 說明。

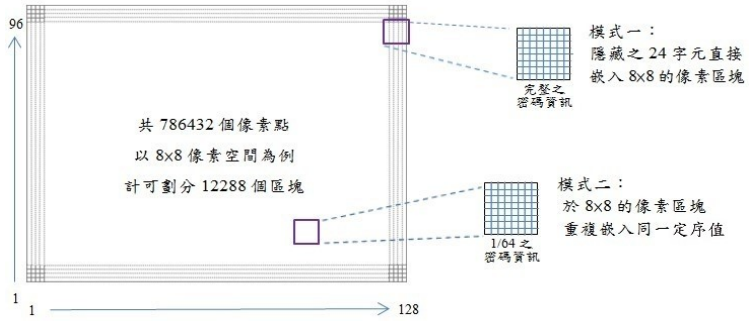


圖10. 1024 × 768像素圖像編碼區塊配置示意圖

3.2. 分散不規則之最低有效位元索引方法

即以隨機方式嵌入，以上法為例，同樣最多可重複嵌入 12288 次，也可僅於圖像之重點部位，嵌入隱藏資訊或密碼，如圖 11。但其為不規則分布，可使外人較無法從其排列發覺隱藏之機密資訊，可提高機密資訊的安全性。但其缺點與區塊密集之最低有效位元索引方法一樣，均是運用 2 個索引位元，共 64 個索引值排序，僅可隱藏 24 個字元空間。當隱藏資訊量大於 24 個字元時，就必須結合其他方法技術，加以輔助。

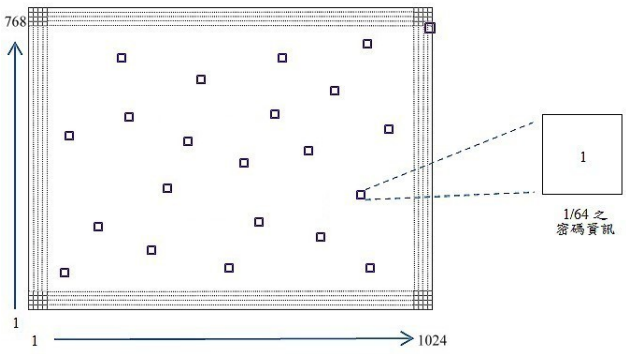


圖11. 分散不規則之最低有效位元索引方法編碼空間配置示意圖

3.3. 將索引值分組排序，以增加索引字元數

此方法是為了彌補區塊密集和分散不規則之最低有效位元索引方法，索引值排序數量的不足而設計，如圖 12，以 1 個索引位元 0 ~ 7 切分二組排序為例，有 (0123), (1234), (2345), (3456), (4567), (5670), (6701), (7012) 共 8 種不同的排列組合，而各單位索引值均出現 4 次，所以便可增加 4 倍的字元數；而且此方法亦可結合索引值原本依序的模式，加以組合運用。簡而言之，例如：二索引位元結合分組排序等等，當然其實際增加之字元數，也會因索引值分組或組合方式的複雜度而有所不同。

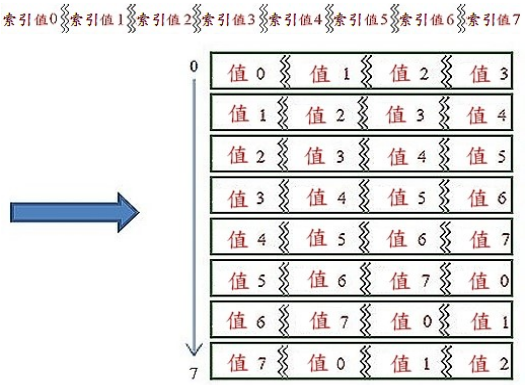


圖12. 索引值分組排序示意圖

3.4. 延伸索引表法(Extended Index Table)

主要是在 3.1 節和 3.2 節的方法，用於龐大訊息之隱藏，當二索引位元 64 個索引值不敷使用時。其亦有多種使用的技巧與方法，可將所有嵌入資訊像素點之對應索引值，或者選擇其關鍵索引值，直接另建存於一個查詢對照表，以供訊息還原使用，如圖 13。

當若採用把所有嵌入資訊像素點之對應索引值，直接建錄於查詢對照表內，則可將 RGB 像素點 8bits 中的最低三位元，全改為資訊位元，可全部運用於隱藏資訊之用途，以提高藏入資料量。而所謂關鍵索引值就是，無論是採用 3.1 節或 3.2 節之方法，均可將其作較大區塊的切分，將這些切分的位置資訊建於查詢對照表內，其查詢對照表的資料容量，會比全部建錄於查詢對照表的資料量小，查詢時也會較為快速方便。

所以只要查表就能清楚知道機密資訊隱藏的位置，亦方便機密資訊的提取與還原。但於浮水印標誌資訊的運用，只要圖像經過變形處理，索引值所對應之位置，就需重新進行調整，其所隱藏之資訊值也會因變形而有所不同，無法對抗外力的破壞，這是所有點陣圖像共通的問題。不過若是在隱藏資訊的傳遞上，就不是個非得解決的問題了，需端視其使用之用途而定論。

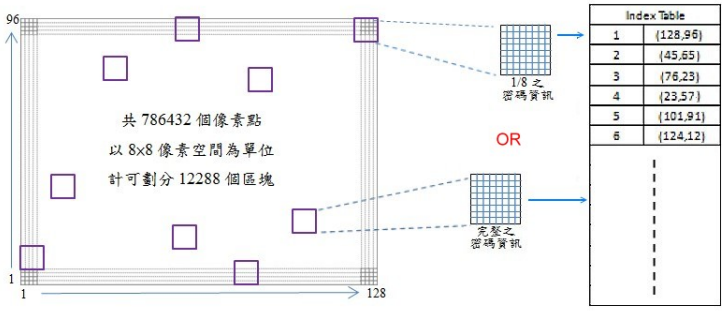
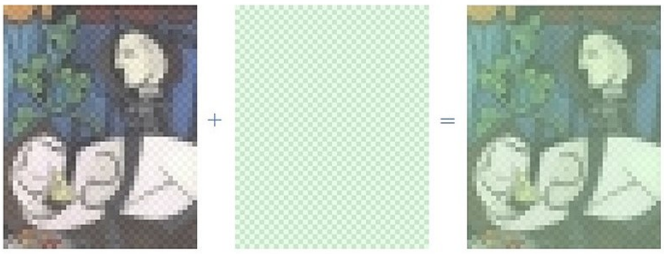


圖13. 延伸索引表法示意圖

3.5. 增加透明遮罩圖或反轉、移位的機制

為了不讓有心人士輕易發覺，於圖像中有嵌入任何之機密資訊，要強化機密資訊的隱密性，可運用 2.1 節介紹的技術原理，於隱藏機密資訊的圖像上，在不影響圖像品質的情形下，再疊加一層似有若無的透明遮罩，或者運用邏輯運算子 XOR (Exclusive OR)反轉的原理，或移位(Shift)機制，將其被嵌入資訊的最低有效位元值進行反轉或移位的動作。其主要目的是在調整其每個像素點的資訊位元值，藉以打亂原有的規律性或隱藏其資訊內容，讓有心人士難以窺視圖像內所隱藏之機密資訊。以增加透明遮罩圖為例，如下圖 14。



$$C_{rgb} = B_{rgb} \times (1-\alpha) + A_{rgb} \times \alpha \qquad C12_{rgb} = \frac{A1_{rgb} \times \alpha 1 \times (1-\alpha 2) + A2 \times \alpha 2}{\alpha 1 + \alpha 2 - \alpha 1 \times \alpha 2}$$

圖14. 運用疊合公式增加透明遮罩圖

以上方法，無論是採用文字或圖像資訊進行隱藏，都是屬於肉眼不可見的機制，可用於機密訊息的傳遞。但若是要遏止分享之數位作品，為有心人士恣意盜用，可適度結合高透明度，也就是 α 值較低的可視浮水印(Visible Watermarking)，於較不影響數位作品的空間，明白宣示作品的智慧財產權。

當然在資訊隱藏技術和方法的應用，其使用隱密性和安全性，絕對是首要考量的關鍵，但也不能因此忽視其實用的便利性，而一味地追求如何提升資料強健性之技術。圖像上隱藏資訊之傳

遞的安全性與便利性，與資訊隱藏所運用之相關技巧方法，以及圖片疊合的張數和方式，與圖片像素的大小和解碼關鍵圖的編碼複雜度，存在著絕對密切的關係。

4. 實驗與分析

如何讓資訊隱藏的技術，成為任何人生活中一項便利的工具，為本研究之首重。所以將會運用一般常見之軟體工具，來進行上述方法之實驗；就個人較常使用例如 Microsoft 系列、Adobe 或 Corel 系列的軟體工具。在此實驗過程中將會使用到 Excel、Photoshop、Fireworks、PhotoImpact 等工具軟體，以及於資訊隱藏最低有效位元索引方法應用時，也會運用到美國訊息交換標準碼 ASCII Code，來做為隱藏資訊英文字符的轉換。

4.1. 資訊隱藏嵌入實驗

以下之各種實驗將以畢卡索於在紐約佳士得拍賣行，以 1.064 億美元的天價拍出的「裸體、綠葉和半身像」畫像，和「Development of Image Information Hiding Techniques Based on Multiple Least Significant Bits」這句文字，作為資訊隱藏實驗之本體與內文。此句文字的二進制和十進制 ASCII Code 以及畢卡索圖像，分別顯示於圖 15 和圖 16。

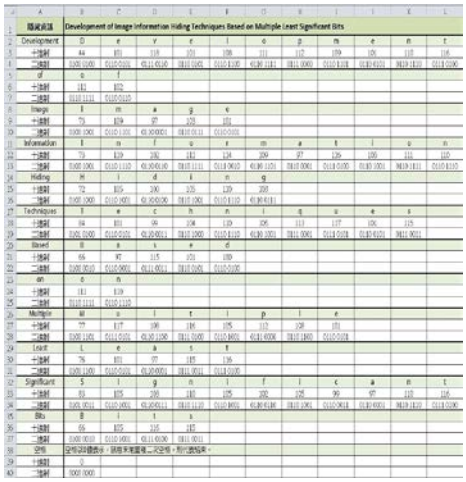


圖15. 二進制/十進制之ASCII Code對照

圖16. 200×200像素紅框之半身頭像

為讓整個實驗的比對更為清楚方便，以下各實驗操作，均會以像素值為 566×719 畢卡索圖像中，所標示 200×200 像素大小紅框之主要圖像部分為實驗樣本，如上圖 16 之右上半身頭像。

4.1.1. 實驗一：運用區塊密集之最低有效位元索引方法

本方法之實驗試運用連續區塊密集嵌入方式，以 8×8 的像素空間單位圖像，並以 2 個索引位元，共可隱藏 24 個字元空間為例。於圖 16 紅框之左上 8×8 的像素區塊，進行「Least Significant Bits」資訊之隱藏，包括 20 個英文字母、2 個空白字符和 2 個空白表結數字符，剛好共計 24 個字元。以每 3 個字元嵌入 8 Bits 的 RGB 值中，分別為「Lea」、「st□」、「Sig」、「nif」、「ica」、「nt□」、「Bit」和「s□□」等共 8 組，如表 7 所示。其中 1 個值為 0 的空格「□」，表英文字句的間隔；連續相連的 0 值空格「□□」，則代表隱藏資訊的結束。

現在就來觀察這 24 字元資訊於嵌入前後，對我們的肉眼在影像視覺上，會造成什麼樣的差別或影響。如圖 17 為嵌入資訊前後之 8×8 像素區塊，所截取圖像的放大圖。雖然每個像素點的 RGB 值均已被更動，就以人類的視覺而言，其實其差異並不那麼明顯。更遑論於 1024×768 或解析度更高的圖像中，實在難以被發掘有被竄改或其中藏有秘密的訊息。

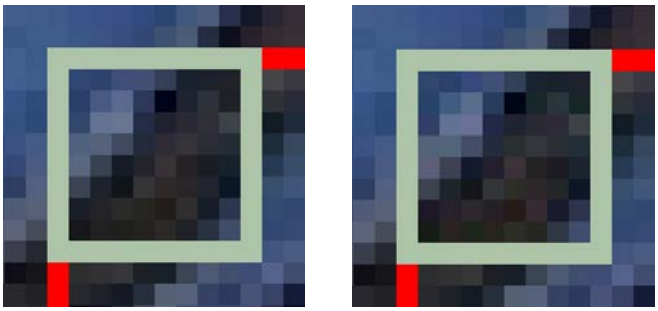


圖17. 左為原圖，右為嵌入資訊之8×8像素區塊圖

最主要是 8 Bits 中僅針對最低的 3 個有效位元，進行資料的嵌入，雖然 3 個位元的更動，會與原值產生≦7 的差距，但也並非每個像素點的 RGB 值，都會有如此大的差距。若以均值的機率概論如表 6 所示，與原數值會產生 7 之差距的機率，僅 2/64；差距在 5 以上且包含 5 的機率，也僅 6/64，連十分之一都不到。所以在眾多圖像的研究與實務應用上，都證明對此最低的 3 個位元數值的改變，對人類的肉眼視覺而言，其辨識的敏感度確實是不高的，且保有一定的圖像品質，並無損及圖像之原樣貌。尤其是在解析度極高的圖像，也就是指在大圖上進行資訊的隱藏，其對些微的改變，更是人類肉眼視覺所無法分辨。

表6. 最低的3個有效位元RGB差值機率

	000	001	010	011	100	101	110	111
000	0	1	2	3	4	5	6	7
001	1	0	1	2	3	4	5	6
010	2	1	0	1	2	3	4	5
011	3	2	1	0	1	2	3	4
100	4	3	2	1	0	1	2	3
101	5	4	3	2	1	0	1	2
110	6	5	4	3	2	1	0	1
111	7	6	5	4	3	2	1	0

話雖說如此，不過此機率的分布畢竟是在均值的情況下，並非那麼絕對；如果有某個資訊隱藏的設計，於嵌入前後的資料數值，剛好都等於 7 值，那真無法想像，那嵌入資訊後圖像的變化，會到達何等驚人的程度。我們也可由實驗所載入的數據中驗證，發現其中的差值剛好為 7 的，竟達有 7 個之多，真是個好的實驗樣本，如表 7。這也讓我們能清楚觀察圖 17 中，左圖與右圖之間的差異性。

表7 區塊密集嵌入資訊前後RGB數值

		嵌入資訊前						嵌入資訊後					
		R		G		B		R		G		B	
	Lea	48	110000	72	1001000	118	1110110	48	110000	72	1001000	112	1110000
		46	101110	67	1000011	110	1101110	41	101001	65	1000001	107	1101011
		62	111110	79	1001111	122	1111010	56	111000	73	1001001	125	1111101
		66	1000010	80	1010000	117	1110101	64	1000000	80	1010000	118	1110110
		52	110100	64	1000000	88	1011000	49	110001	66	1000010	88	1011000
		26	11010	33	100001	49	110001	25	11001	35	100011	50	110010
		29	11101	33	100001	44	101100	24	11000	34	100010	44	101100
		34	100010	35	100011	40	101000	32	100000	35	100011	47	101111
	st□	41	101001	64	1000000	106	1101010	40	101000	68	1000100	104	1101000
		71	1000111	90	1011010	130	10000010	65	1000001	93	1011101	130	10000010
		90	1011010	106	1101010	142	10001110	89	1011001	109	1101101	140	10001100
		46	101110	57	111001	89	1011001	41	101001	61	111101	94	1011110
		0	0	3	11	25	11001	0	0	6	110	24	11000
		31	11111	35	100011	47	101111	24	11000	39	100111	42	101010
		47	101111	50	110010	59	111011	41	101001	54	110110	60	111100
		31	11111	34	100010	39	100111	25	11001	38	100110	38	100110
	Sig	51	110011	71	1000111	108	1101100	50	110010	64	1000000	104	1101000
		99	1100011	115	1110011	148	10010100	99	1100011	113	1110001	147	10010011
		104	1101000	118	1110110	147	10010011	106	1101010	113	1110001	149	10010101
		37	100101	44	101100	70	1000110	35	100011	40	101000	70	1000110
		32	100000	34	100010	49	110001	34	100010	35	100011	48	110000
		43	101011	41	101001	52	110100	42	101010	42	101010	51	110011
		47	101111	47	101111	55	110111	43	101011	42	101010	55	110101
		62	111110	67	1000011	71	1000111	59	111011	67	1000011	71	1000111
	nif	80	1010000	97	1100001	127	1111111	82	1010010	100	1100100	120	1111000
		84	1010100	98	1100010	124	1111100	83	1010011	101	1100101	123	1111011
		64	1000000	75	1001011	97	1100001	67	1000011	77	1001101	101	1100101
		36	100100	42	101010	58	111010	34	100010	44	101100	62	111110
		47	101111	44	101100	55	110111	43	101011	47	101111	48	110000
		54	110110	51	110011	58	111010	51	110011	54	110110	59	111011
		44	101100	44	101100	52	110100	43	101011	46	101110	53	110101
		37	100101	44	101100	52	110100	34	100010	47	101111	54	110110
	ica	91	1011011	103	1100111	127	1111111	92	1011100	96	1100000	120	1111000
		57	111001	66	1000010	83	1010011	61	111101	65	1000001	83	1010011
		22	10110	28	11100	40	101000	21	10101	25	11001	45	101101
		36	100100	36	100100	44	101100	36	100100	32	100000	46	101110
		56	111000	51	110011	57	111001	61	111101	50	110010	56	111000
		38	100110	33	100001	37	100101	36	100100	34	100010	34	100010
		26	11010	29	11101	36	100100	28	11100	27	11011	36	100100
		24	11000	34	100010	44	101100	29	11101	35	100011	47	101111
	nt□	60	111100	67	1000011	83	1010011	60	111100	68	1000100	80	1010000
		30	11110	34	100010	45	101101	29	11101	37	100101	42	101010
		18	10010	21	10101	26	11010	21	10101	21	10101	28	11100
		53	110101	53	110101	55	110111	52	110100	53	110101	54	110110
		47	101111	41	101001	41	101001	45	101101	46	101110	40	101000
		42	101010	37	100101	41	101001	45	101101	39	100111	42	101010
		34	100010	37	100101	46	101110	37	100101	38	100110	44	101100
		11	1011	23	10111	39	100111	12	1100	22	10110	38	100110
	Bit	27	11011	31	11111	40	101000	30	11110	24	11000	40	101000
		35	100011	36	100100	40	101000	39	100111	33	100001	43	101011
		43	101011	43	101011	43	101011	46	101110	41	101001	45	101101
		39	100111	38	100110	34	100010	38	100110	32	100000	39	100111
		35	100011	30	11110	27	11011	38	100110	27	11011	24	11000
		48	110000	46	101110	49	110001	54	110110	42	101010	51	110011
		22	10110	28	11100	40	101000	23	10111	26	11010	44	101100
		43	101011	55	110111	79	1001111	46	101110	51	110011	78	1001110
	s□□	23	10111	24	11000	28	11100	22	10110	28	11100	24	11000
		33	100001	33	100001	33	100001	39	100111	36	100100	34	100010
		31	11111	32	100000	27	11011	31	11111	36	100100	28	11100
		36	100100	35	100011	31	11111	39	100111	36	100100	30	11110
		43	101011	42	101010	40	101000	46	101110	46	101110	40	101000
		33	100001	32	100000	38	100110	38	100110	38	100110	34	100010
		43	101011	50	110010	68	1000100	47	101111	54	110110	68	1000100
		63	111111	77	1001101	104	1101000	63	111111	78	1001110	110	1101110

4.1.2. 實驗二：分散不規則之索引方法結合索引表之運用

在實驗一以 8×8 的像素空間，運用 2 個索引位元，進行連續區塊密集的嵌入方式中，確定其嵌入資訊的可行性，並證實最低 3 位元的資料嵌入，對於人類肉眼而言，是不易被察覺的，且無損原圖的品質。所以接下來之此實驗，同樣為方便觀察，僅將於實驗一 8×8 的像素空間旁，接續擴大 3 倍的像素空間，來進行分散不規則之最低有效位元索引方法，和延伸索引表法的同時運用。

雖然所要嵌入的資訊為「Information Hiding」，以 64 索引值就可分辨決定其排序；在此結合索引表的運用，純粹為了同時進行此法的演繹。同樣以每 3 個字元嵌入 8 Bits 的 RGB 值中，分別為「Inf」、「orm」、「ati」、「on□」、「Hid」、「ing」和「□□」等 7 組，同樣 1 個值為 0 的空格「□」，表英文字句的間隔；連續相連的 0 值空格「□□」，則代表隱藏資訊的結束。除原有索引值的運用外，同時利用索引表，將這 8×7=56 個像素點資料，分散嵌入於 8×8×3 的像素空間，此區塊空間像素點之 RGB 數值如下列表 8。

表8. 分散嵌入8×8×3像素空間原圖之RGB數值

	R		G		B		R		G		B		R		G		B	
2	47	101111	45	101101	48	110000	70	1000110	93	1011101	145	10010001	24	11000	32	100000	53	110101
3	71	1000111	69	1000101	72	1001000	69	1000101	92	1011100	144	10010000	10	1010	14	1110	26	11010
4	58	111010	57	111001	65	1000001	70	1000110	94	1011110	142	10001110	19	10011	19	10011	29	11101
5	28	11100	29	11101	47	101111	76	1001100	100	1100100	146	10010010	24	11000	23	10111	29	11101
6	30	11110	37	100101	65	1000001	71	1000111	93	1011101	140	10001100	15	1111	13	1101	18	10010
7	48	110000	62	111110	99	1100011	58	111010	79	1001111	122	11110010	20	10100	18	10010	23	10111
8	63	111111	81	1010001	127	1111111	73	1001001	89	1011001	125	11111001	26	11010	25	11001	31	11111
9	68	1000100	90	1011010	140	10001100	79	1001111	93	1011101	122	11110010	27	11011	26	11010	32	100000
10	34	100010	35	100011	39	100111	71	1000111	94	1011110	146	10010010	10	1010	17	10001	36	100100
11	48	110000	51	110011	56	111000	68	1000100	94	1011110	143	10001111	15	1111	19	10011	31	11111
12	40	101000	44	101100	56	111000	62	111110	89	1011001	136	10001000	28	11100	28	11100	36	100100
13	26	11010	32	100000	54	110110	59	111011	86	1010110	131	10000011	26	11010	25	11001	31	11111
14	65	1000001	76	1001100	108	1101100	62	111110	84	1010100	131	10000011	20	10100	18	10010	23	10111
15	67	1000011	82	1010010	125	1111101	62	111110	83	1010011	126	11111110	25	11001	23	10111	28	11100
16	61	111101	79	1001111	129	10000001	77	1001101	93	1011101	129	10000001	29	11101	28	11100	34	100010
17	62	111110	83	1010011	136	10001000	67	1000011	78	1001110	106	11010010	28	11100	27	11011	33	100001
18	53	110101	60	111100	66	1000010	75	1001011	98	1100010	150	10010110	13	1101	20	10100	39	100111
19	18	10010	28	11100	37	100101	73	1001001	100	1100100	147	10010011	14	1110	16	10000	29	11101
20	31	11111	43	101011	59	111011	71	1000111	100	1100100	144	10010000	25	11001	25	11001	33	100001
21	77	1001101	91	1011011	117	1110101	59	111011	86	1010110	129	10000001	31	11111	30	11110	35	100011
22	79	1001111	93	1011101	130	10000010	66	1000010	89	1011001	133	10000101	27	11011	25	11001	30	11110
23	72	1001000	89	1011001	135	10000111	60	111100	79	1001111	121	11111001	31	11111	29	11101	34	100010
24	63	111111	80	1010000	132	10000100	76	1001100	90	1011010	125	11111101	33	100001	32	100000	38	100110
25	68	1000100	89	1011001	142	10001110	70	1000110	81	1010001	109	11011101	29	11101	30	11110	35	100011
26	27	11011	40	101000	49	110001	81	1010001	103	1100111	150	10010110	16	10000	21	10101	40	101000
27	29	11101	45	101101	60	111100	75	1001011	99	1100011	143	10001111	16	10000	18	10010	30	11110
28	54	110110	72	1001000	94	1011110	77	1001101	106	1101010	146	10010010	21	10101	22	10110	27	11011
29	61	111101	78	1001110	108	1101100	64	1000000	92	1011100	131	10000011	25	11001	24	11000	29	11101
30	49	110001	67	1000011	107	1101011	66	1000010	89	1011001	131	10000011	21	10101	19	10011	24	11000
31	60	111100	77	1001101	123	1111011	60	111100	77	1001101	120	11110000	24	11000	23	10111	28	11100
32	70	1000110	86	1010110	137	10001001	79	1001111	93	1011101	128	10000000	27	11011	28	11100	33	100001
33	85	1010101	103	1100111	153	10011001	73	1001001	83	1010011	110	11011110	25	11001	26	11010	31	11111
34	23	10111	39	100111	55	110111	81	1010001	99	1100011	139	10001011	9	1001	14	1110	33	100001
35	38	100110	58	111010	82	1010010	62	111110	84	1010100	123	11110111	24	11000	26	11010	38	100110
36	56	111000	77	1001101	106	1101010	71	1000111	97	1100001	132	10000100	26	11010	27	11011	32	100000
37	49	110001	69	1000101	104	1101000	78	1001110	104	1101000	141	10001101	15	1111	14	1110	19	10011
38	55	110111	73	1001001	113	1110001	59	111011	81	1010001	122	11110010	15	1111	14	1110	19	10011
39	64	1000000	79	1001111	122	1111010	66	1000010	81	1010001	122	11110010	18	10010	19	10011	23	10111
40	70	1000110	83	1010011	127	1111111	84	1010100	97	1100001	131	10000011	23	10111	24	11000	29	11101
41	88	1011000	101	1100101	145	10010001	64	1000000	74	1001010	101	11001010	23	10111	24	11000	29	11101
42	19	10011	36	100100	62	111110	42	101010	55	110111	87	10101111	6	110	9	1001	26	11010
43	56	111000	79	1001111	110	1101110	18	10010	33	100001	66	11000010	27	11011	30	11110	39	100111
44	64	1000000	86	1010110	123	1111011	36	100100	56	111000	89	10110001	31	11111	30	11110	36	100100
45	51	110011	73	1001001	110	1101110	68	1000100	88	1011000	123	11110111	16	10000	15	1111	20	10100
46	74	1001010	90	1011010	126	1111110	58	111010	75	1001011	118	11101110	22	10110	21	10101	26	11010
47	56	111000	69	1000101	104	1101000	63	111111	78	1001110	119	11101111	24	11000	23	10111	28	11100
48	34	100010	42	101010	78	1001110	80	1010000	93	1011101	127	11111111	26	11010	27	11011	32	100000
49	47	101111	56	111000	89	1011001	58	111010	65	1000001	91	10110011	27	11011	28	11100	33	100001
50	72	1001000	91	1011011	124	1111100	9	1001	15	1111	37	100101	9	1001	12	1100	29	11101
51	52	110100	74	1001010	113	1110001	1	1	11	1011	36	100100	23	10111	26	11010	35	100011
52	50	110010	72	1001000	113	1110001	48	110000	62	111110	91	10110011	23	10111	22	10110	28	11100
53	65	1000001	85	1010101	122	1111010	57	111001	71	1000111	106	11010010	24	11000	24	11000	26	11010
54	79	1001111	94	1011110	125	1111101	55	110111	70	1000110	111	11011111	16	10000	16	10000	18	10010
55	44	101100	54	110110	81	1010001	66	1000010	80	1010000	119	11110111	24	11000	23	10111	28	11100
56	21	10101	27	11011	49	110001	86	1010110	97	1100001	129	10000001	32	100000	33	100001	38	100110
57	24	11000	29	11101	49	110001	53	110101	60	111100	86	10101110	29	11101	30	11110	35	100011
58	65	1000001	85	1010101	122	1111010	43	101011	47	101111	59	11101011	10	1010	13	1101	28	11100
59	50	110010	71	1000111	114	1110010	12	1100	17	10001	36	100100	27	11011	30	11110	37	100101
60	55	110111	74	1001010	116	1110100	53	110101	60	111100	88	10110000	25	11001	24	11000	29	11101
61	94	1011110	110	1101110	146	10010010	75	1001011	86	1010110	118	11101110	20	10100	20	10100	22	10110
62	51	110011	63	111111	89	1011001	49	110001	63	111111	102	11001110	12	1100	12	1100	14	1110
63	28	11100	35	100011	53	110101	62	111110	76	1001100	115	11100011	24	11000	24	11000	26	11010
64	35	100011	37	100101	49	110001	81	1010001	92	1011100	124	11111100	27	11011	28	11100	33	100001
65	63	111111	66	1000010	75	1001011	47	101111	55	110111	78	1001110	26	11010	27	11011	32	100000

分散嵌入之位置索引，在此以試算表上列的英文字母 A (B、C、D、E、F)、字母 G (H、I、J、K、L)、字母 M (N、O、P、Q、R)，和左行的數字來表示。以表 3.1.2.1 左上第一個 rgb (47, 45, 48)為例，將以 A2 為其位置索引。建構之分散嵌入前後 RGB 值和位置索引如表 9 所示。

表9. 分散嵌入資訊前後RGB數值與位置索引表

		嵌入資訊前						嵌入資訊後						
		R		G		B		R		G		B		位置索引(Index)
3	Inf	73	1001001	89	1011001	125	1111101	72	1001000	88	1011000	120	1111000	G8
4		23	10111	22	10110	28	11100	17	10001	17	10001	27	11011	M52
5		29	11101	45	101101	60	111100	24	11000	41	101001	61	111101	A27
6		26	11010	27	11011	32	100000	24	11000	24	11000	38	100110	M48
7		64	1000000	86	1010110	123	1111011	65	1000001	83	1010011	120	1111000	A44
8		67	1000011	82	1010010	125	1111101	64	1000000	83	1010011	123	1111011	A15
9		59	111011	81	1010001	122	1111010	56	111000	83	1010011	125	1111101	G38
10		29	11101	30	11110	35	100011	25	11001	26	11010	38	100110	M25
11	orm	47	101111	45	101101	48	110000	40	101000	44	101100	48	110000	A2
12		77	1001101	106	1101010	146	10010010	73	1001001	109	1101101	147	10010011	G28
13		51	110011	63	111111	89	1011001	49	110001	61	111101	93	1011101	A62
14		24	11000	24	11000	26	11010	24	11000	29	11101	30	11110	M63
15		63	111111	78	1001110	119	1110111	57	111001	78	1001110	113	1110001	G47
16		56	111000	77	1001101	106	1101010	57	111001	78	1001110	107	1101011	A36
17		69	1000101	92	1011100	144	10010000	65	1000001	95	1011111	148	10010100	G3
18		28	11100	28	11100	36	100100	25	11001	30	11110	39	100111	M12
19	ati	10	1010	13	1101	28	11100	10	1010	8	1000	24	11000	M58
20		26	11010	32	100000	54	110110	27	11011	33	100001	51	110011	A13
21		16	10000	18	10010	30	11110	19	10011	17	10001	29	11101	M27
22		68	1000100	94	1011110	143	10001111	66	1000010	89	1011001	142	10001110	G11
23		44	101100	54	110110	81	1010001	42	101010	50	110010	81	1010001	A55
24		81	1010001	99	1100011	139	10001011	82	1010010	99	1100011	138	10001010	G34
25		75	1001011	86	1010110	118	1110110	74	1001010	82	1010010	116	1110100	G61
26		13	1101	20	10100	39	100111	11	1011	18	10010	39	100111	M18
27	on□	19	10011	19	10011	29	11101	18	10010	20	10100	24	11000	M4
28		34	100010	35	100011	39	100111	35	100011	37	100101	34	100010	A10
29		26	11010	25	11001	31	11111	27	11011	29	11101	28	11100	M13
30		68	1000100	88	1011000	123	1111011	66	1000010	92	1011100	126	1111110	G45
31		71	1000111	97	1100001	132	10000100	67	1000011	103	1100111	128	10000000	G36
32		49	110001	67	1000011	107	1101011	51	110011	71	1000111	106	1101010	A30
33		25	11001	25	11001	33	100001	27	11011	31	11111	36	100100	M20
34		12	1100	12	1100	14	1110	11	1011	14	1110	14	1110	M62
35	Hid	1	1	11	1011	36	100100	4	100	8	1000	32	100000	G51
36		52	110100	74	1001010	113	1110001	53	110101	73	1001001	115	1110011	A51
37		15	1111	13	1101	18	10010	12	1100	9	1001	21	10101	M6
38		23	10111	24	11000	29	11101	20	10100	24	11000	30	11110	M40
39		29	11101	28	11100	34	100010	29	11101	27	11011	32	100000	M16
40		38	100110	33	100001	37	100101	36	100100	58	111010	91	1011011	G44
41		78	1001110	104	1101000	141	10001101	76	1001100	106	1101010	140	10001100	G37
42		79	1001111	93	1011101	130	10000010	76	1001100	91	1011011	134	10000110	A22
43	ing	72	1001000	91	1011011	124	1111100	76	1001100	92	1011100	120	1111000	A50
44		76	1001100	90	1011010	125	1111101	77	1001101	93	1011101	123	1111011	G24
45		18	10010	19	10011	23	10111	21	10101	21	10101	21	10101	M39
46		67	1000011	78	1001110	106	1101010	68	1000100	76	1001100	110	1101110	G17
47		53	110101	60	111100	86	1010110	53	110101	63	111111	80	1010000	G57
48		18	10010	28	11100	37	100101	20	10100	31	11111	35	100011	A19
49		68	1000100	90	1011010	140	10001100	68	1000100	95	1011111	141	10001101	A9
50		26	11010	27	11011	32	100000	29	11101	30	11110	39	100111	M36
51	□□	27	11011	28	11100	33	100001	30	11110	24	11000	33	100001	M32
52		60	111100	79	1001111	121	1111001	62	111110	72	1001000	123	1111011	G23
53		26	11010	25	11001	31	11111	30	11110	24	11000	29	11101	M8
54		85	1010101	103	1100111	153	10011001	86	1010110	96	1100000	159	10011111	A33
55		10	1010	17	10001	36	100100	14	1110	18	10010	32	100000	M10
56		47	101111	55	110111	78	1001110	46	101110	50	110010	74	1001010	G65
57		56	111000	69	1000101	104	1101000	62	111110	66	1000010	108	1101100	A47
58		19	10011	36	100100	62	111110	22	10110	34	100010	62	111110	A42

從實驗結果之資訊嵌入前後圖觀察，發覺分散不規則嵌入的方式，確實比實驗一之區塊密集嵌入方式，更不會影響到圖像的品質，其嵌入資訊前後，幾乎沒有差別，如下面圖 18。其實位置索引的作用，主要是在嵌入字元數量龐大，且非重複字元的情況下，與索引方法結合運用，才能發揮其作用。所以在此實驗中較不具重要性，且與實驗結果的產出無關，只是為了說明與展現位置索引的使用方法與步驟。

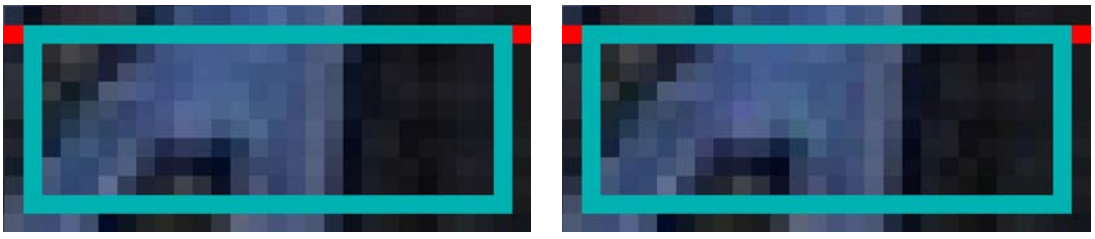


圖18. 左為原圖，右為嵌入資訊之8×8×3像素區塊圖

從以上二實驗結果，最低有效位元的應用，確實可運用於資訊隱藏的領域；在學術的研究上，也許學者們多以頻率域的編碼、轉碼為鑽研的目標，因有其隱藏資訊容量不大，以及實務上應用的困難度與高門檻等問題，仍有待改善解決，故無法使頻率領域的資訊隱藏技術，廣被使用。在

實務上仍以空間領域之點陣圖隱藏資訊的應用，最為普遍、簡捷，也是大多數的數位工作者，以直觀的方式，就能操作使用。

4.2. 峰值訊噪比(Peak Signal to Noise Ratio, PSNR)之分析驗證

數位影像在經過嵌入資訊或壓縮之後,所輸出之影像通常都會與原始影像產成某種程度的差異，而 PSNR 值得評估是最普遍、最廣泛使用的評鑑畫質的客觀量測法，因此我們以 PSNR 值來分析評估，經過嵌入資訊程序後影像品質的效能。

PSNR 值定義如下[15]：

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right), \qquad MSE = \frac{\sum_{n=1}^{FrameSize}(I_n - P_n)^2}{FrameSize}$$

其中 PSNR 的 Peak 就是指 8 bits 表示法的最大值 255;而 MSE 指是指均方誤差值(Mean Square Error)，*In* 指原始影像第 *n* 個 pixel 值，*Pn* 指經處理後的影像第 *n* 個 pixel 值。

均方誤差值(MSE)為一般影像評估常用的方式之一，求出來的 MSE 值越小，表示輸入影像與複製影像之間的差異少，也就代表品質較好，一般肉眼較無法辨識其差異。均方誤差值計算方式，是將輸入影像以及複製影像的每一個像素點相減，再將其差值的平方加總起來取平均值。

峰值訊噪比(PSNR)經常用作圖像壓縮等領域中信號重建質量的測量方法，主要是利用影像信號的最大值與影像中雜訊的比值作為評估的標準。對於每點有的 RGB 三個值的彩色圖像來說峰值訊噪比的定義類似,只是均方差是所有方差之和除以圖像尺寸再除以 3。PSNR 的單位為 dB，PSNR 值越大，就代表失真越少，這是一個客觀的評比數據，但有時候並不能完全代表人的主觀感受，通常 PSNR 值越高，則表示越無法用一般肉眼辨識其與原圖間的差異。

而 PSNR 值的高低對於影像品質是否失真的評估[16]，就一般而言，當所計算出的 PSNR 值介於 40 dB~50 dB 之間時，便代表其有良好的圖像品質；而當 PSNR 值大約於 30 dB 左右時，則代表其具有中等的圖像品質，因此大部分 PSNR 之檢測值，皆被要求至少要大於 30 dB 以上，其已嵌入資訊之影像才會被接受。但有時會因為受到嵌入資訊圖像之位置，以及嵌入容量的大小而有所影響；所以 PSNR 值的高低，並不絕對代表影像品質與原圖差異性的大小，有時候還是必須靠人的肉眼輔助來判斷影像的品質才較為正確。

4.2.1. 資訊嵌入實驗驗證一

前面 4.1 節所述之實驗一所截取之畫面圖像為 14×14=196 個像素點，包含 RGB 三原色共計有 588 個位元組，即 4704 個位元；而嵌入資訊為 8×8=64 個像素點，共計有 64×3=192 個資訊位元；也就是嵌入約 4.08%的資訊量，達到 PSNR 值為 47.05 dB 的影像品質；而實驗二所截取之畫面圖像為 28×12=336 個像素點，嵌入資訊量一樣為 8×8=64 個像素點 192 個資訊位元，其嵌入資訊量約為 2.38%，PSNR 值則約為 49.16 dB 的影像品質。

對照 Shie [17]與 Lai [18]所提的資訊隱藏技術，其在 512×512=262144 個像素點之灰階圖像中，分別嵌入 48 Kbits、64 Kbits 和 80 Kbits 之資訊，其 PSNR 分析數據，如下列表 10。

表10. 文獻[17]與[18]之PSNR值

嵌入資訊量(bit)	48K		64K		80K	
	Shie [17]	Lai [18]	Shie [17]	Lai [18]	Shie [17]	Lai [18]
Lena	29.84	30.57	27.57	28.80	23.76	26.85
F16	31.51	32.89	29.57	31.64	26.79	30.05
Baboon	24.51	26.04	22.02	24.86	20.49	23.33
Pepper	30.07	30.64	28.42	29.02	26.81	27.18
Barbara	24.27	24.88	22.59	23.79	19.86	22.37

以上表中嵌入 48Kbits、64Kbits 和 80Kbits 的資料量來分析，在 512×512 之灰階圖像 262144×3×8 bit =6291456bit 的空間容量中(也就是等於在 6144Kbit)，其嵌入資訊量僅分別約為 0.78%、1.04%和 1.30%。而所本研究所進行的兩個實驗的資訊嵌入量約為 4.08%與 2.38%，嵌入的資訊量比較多，但 PSNR 分別為 47.05dB 與 49.16 dB，顯示影像品質較佳，因此證明所提出的方法的優勢。

2009 年時 Shie 等學者提出向量量化的改進方法[19]，其實驗之 PSNR 值雖比其之前 2006 年所發表之數據為高，尤其是在 80Kbits 的資訊嵌入量有特別明顯的改善，但也僅達 32.6 dB [19].

4.2.2. 資訊嵌入實驗驗證二

接下來我們引用[17]和[18]實驗中所使用之相同的五張 512×512 像素大小實驗樣圖，以 2 個索引位元和 1 個資訊位元之最低有效位元方法，於各組同樣分別嵌入 48 Kbits、64 Kbits 和 80 Kbits 資料量，來進行資訊嵌入實驗的對照。因為本方法分別於每個像素點的 RGB 值中所嵌入的資訊，扣除 2 個索引位元不予計算，真正存入的資訊內容僅占 1bit 之最低有效位元空間，也就是 1 個像素點僅可嵌入於 R、G、B 共 3bits 容量的資訊，所以要於圖像中嵌入 48 Kbits (49152 bits)、64 Kbits (65536 bits)和 80 Kbits (81920 bits)的資料量，則至少需要依序分別為 16384、21845 和 27307 個像素點的空間容量。

為實驗設計方便，我們選用至少大於 16384、21845 和 27307 個像素點空間容量的圖像方塊，來模擬所嵌入的資訊內容，其依序分別相當於 128×128、148×148 和 166×166 像素大小的圖像方塊。當然要進行資訊的嵌入，不一定非得使用圖像方塊，只要相當於 48 Kbits、64 Kbits 和 80 Kbits 資料量之任何型態的嵌入樣本，都可運用於此資訊隱藏嵌入的實驗，本實驗用之圖像樣本如下面圖 19。

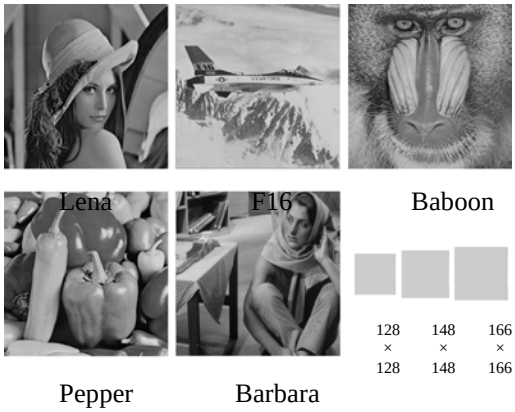


圖19. 峰值訊噪比(PSNR)實驗之圖像樣本

將上圖 Lena、F16、Baboon、Pepper 和 Barbara 之五張實驗用圖像，分別嵌入相當於 48 Kbits、64 Kbits 和 80 Kbits 資料量的圖像方塊，嵌入的技巧在於運用透明遮罩透明度 α 值原理，使每一像素點之 RGB 值控制於正負 7 之間，也就是盡量保持 2 個索引位元和 1 個資訊位元，共 3 個最低有效位元的資料變動。同時，在此實驗中我們也會運用 PSNR Calculators 軟體，來計算 Lena、F16、Baboon、Pepper 和 Barbara 五張圖像，於嵌入資訊後圖像之 PSNR 值；其嵌入結果圖像之 PSNR 值如下列表 11 所示。

表11. 最低有效位元索引方法與Shie et al.、Lai之PSNR值比較

Capacity (in bits)	48K			64K			80K		
	Shie et al.	Lai	The Proposed	Shie et al.	Lai	The Proposed	Shie et al.	Lai	The Proposed
Lena	29.84	30.57	52.41	27.57	28.80	51.19	23.76	26.85	50.28
F16	31.51	32.89	57.37	29.57	31.64	56.21	26.79	30.05	55.24
Baboon	24.51	26.04	57.36	22.02	24.86	56.35	20.49	23.33	55.44
Pepper	30.07	30.64	55.93	28.42	29.02	54.35	26.81	27.18	53.36
Barbara	24.27	24.88	54.92	22.59	23.79	52.81	19.86	22.37	51.61

透過上面表 11 之 PSNR 值的數據比較，得知最低有效位元索引方法，對於原圖在進行隱藏資訊後，並未造成圖像上明顯的破壞。在此我們以 PSNR 數值較差的 Lena 和 Barbara 二組圖像，來做為人類肉眼觀察上的視覺比較，如下面圖 20 與圖 21。

Lena 原圖

嵌入資料量=48Kbits
PSNR=52.41dB

嵌入資料量=64Kbits
PSNR=51.19dB

嵌入資料量=80Kbits
PSNR=50.28dB

圖20. Lena原圖與嵌入資訊圖像之視覺比較

Barbara 原圖

嵌入資料量=48Kbits
PSNR=54.92dB

嵌入資料量=64Kbits
PSNR=52.81dB

嵌入資料量=80Kbits
PSNR=51.61dB

圖21. Barbara原圖與嵌入資訊圖像之視覺比較

最低有效位元索引方法於此 PSNR 值的實驗測量上，雖然分別嵌入 48 Kbits、64 Kbits 和 80 Kbits 之資料量，在表象上僅運用 1 個最低有效位元之資訊位元；但實際上在這 512×512 像素大小的圖像中，所嵌入之資訊容量，尚另包括 2 個索引有效位元，所以真正嵌入的資訊容量大小，其實是為 48 Kbits、64 Kbits 和 80 Kbits 的 3 倍資料量。雖然如此，其 PSNR 值在實驗結果數據上的表現，並未因 3 倍的嵌入量而受影響，證實了所提之最低有效位元索引方法於資訊隱藏技術的價值。

5. 結論

本文最主要的目的，就是研究發掘一些符合資訊隱藏機制要求，在實務上確實能運用的方法和技術；其除了具備基本的安全性與隱蔽性外，透過增加索引位元的定序方法，來強化嵌入資訊的獨立性與完整性，藉以改進最低有效位元方法於資訊隱藏技術的運用，同時強化索引位元的排序功能，結合區塊切割，疊加透明遮罩的技術，並增加最低有效位元值反轉或位移的機制，來增強資訊隱藏的隱密性與強健性。且能夠以最簡捷的方式，完成資訊隱藏在實務上的應用，提供大家更多關於資訊隱藏技術的選擇。

由以上研究探討和實驗結果證實，點陣圖格式之圖像，其資訊隱藏量之大，已是無庸置疑的；運用周遭的軟體工具和圖像上的各種計算公式，便可對像素點直接進行色彩 RGB 值的編排，以完成疊加、柔光、屏幕、濾色等模式的操作，來改變像素點原始的顏色，或者嵌入隱藏資訊。而圖像上資訊隱藏的安全性與便利性，其實與疊合圖片的大小、張數和編碼方法，有絕對密切的關係。無論是文字密碼型態的資訊隱藏，或者是隱藏圖像的表現，如浮水印。其技術原理是相通的，且都可以靈活組合運用，以提高安全性與實用性。雖然資訊隱藏的領域，在 1999 年時 Petitcolas[2] 等學者，對資訊隱藏技術有較嚴格的劃分為四大類，但由於在各種技術和方法的交互併用下，使得資訊隱藏技術的共通性提高，分類的界線也因此趨於模糊。它可以是隱寫術[6][7][8][9]，也可以是不可視浮水印或圖像裡的數位簽章，姑且不論其為何；在資訊隱藏技術和方法的應用，隱密性和安全性絕對是首要考量。不過最主要目的還是要回歸於實務需求上的應用。

本文所提之各種資訊隱藏技術的組合技巧和方法，除了可以有效改善最低有效位元於資訊隱藏技術領域，強健性不足的問題外，並提出許多增強嵌入資訊隱密性的方法，對於均可對數位化作品的智慧財產權和有心人士，達到某種程度之保護與遏止的作用。並建議另外結合 α 值較低的浮水印或標記，直接明白宣示作品的著作權，以嚇阻作品被恣意盜用。點陣圖的運用在目前依舊是數位圖像的主流，雖然在於圖像編輯運用上，有其方便性；但其圖像本身較不具強健性，隱藏於內之資訊，亦較容易遭受修改與破壞。而較常使用的圖像格式除了點陣圖外，向量圖也是許多數位設計者常用之圖檔，且其具有圖像的高強健性，是點陣圖無法相提並論的；雖然目前圖像上的資訊隱藏，在實務應用上仍多偏重於點陣圖的研究，也期待未來對於向量圖檔在資訊隱藏領域的研究，能獲得進一步的發展，開拓出資訊隱藏領域的新格局。

雖然關於資訊隱藏研究，在各種領域上的技術，看似蓬勃發展；但在實務應用層面上，仍稍有不足，無法使人人享用到資訊隱藏技術，所帶來的安全性與便利性。尤其是在本論文所介紹之資訊隱藏相關技術中，於未來的研究，無論是空間域或是頻率域的方法，均可試將其理論技術加以整合，並導入於實務應用上；提供更多在實務上，能加以應用的方法或技術，讓資訊隱藏的技巧，更加多元與便利，以提高個人作品保護機制的獨特性，這對於許多的數位創作者來說，實更為之重要。個人隱私與秘密資訊被竊取，或數位作品遭恣意盜用，已是無可避免的事實。在技術實務層面，除了不斷地提昇各種方法和技術，來達到保護數位作品與嚇阻惡意破壞行為的作用外；在道德層面，也希望各國政府都能從教育著手，加強世界公民道德涵養的教化與宣導，讓個人隱私權和著作權能受到大家彼此的尊重，這何嘗不也是解決此問題的根本之道。

6. 參考文獻

[1] Noar, M., & Shamir, A. (1995). Visual Cryptography. *Advances in Cryptology:Eurpocrypt'94*, Berlin, Springer-Verlag, pp. 1-12.

[2] Petitcolas, F. A. P., Anderson, R. J., & Kuhn, M. G. (1999). Information hiding—A Survey. *Proc. IEEE Special Issue on Identification and Protection of Multimedia Information*, vol. 87, pp. 1062-1078.

[3] Smid, M. E., & Branstad, D. K. (1977). The Data Encryption Standard. *Federal Information Processing Standards Publication*, vol. 46, pp. 43-64., National Institute of Standards & Technology.

[4] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Ppublic-Key Cryptosystems. *Communications of the ACM*, vol. 21, No. 2, pp. 120-130.

[5] Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., & Roback, E. (2000). Report on the Development of the Advanced Encryption Standard (AES). National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce.

[6] Anderson, R. J., & Petitcolas, F. A. P. (1998). On the Limits of Steganography. *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474-481.

[7] Lie, W. N., & Chang, L. C. (1999). Data Hiding in Images with Adaptive Numbers of Least Significant Bits Based on the Human Visual System. *IEEE International Conference on Image Processing*, pp.286-290.

[8] Zhang, X., & Wang, S. (2006). Efficient Steganographic Embedding by Exploiting Modification Direction. *IEEE Communications Letters*, vol. 10, no. 11 , pp. 781-783.

[9] Luo, W., Huang, F., & Huang, J. (2010). Edge Adaptive Image Steganography Based on LSB Matching Revisited. *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 201-214.

[10] Swanson, M. D., Kobayashi, M., & Tewfik, A. H. (1998). Multimedia Data-Embedding and Watermarking Technologies. *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064-1087.

[11] Langelaar, G. C., & Lagendijk, R. L. (2001). Optimal Differential Energy Watermarking of DCT Encoded Images and Video. *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 148-158.

[12] Chan, C. K., & Cheng, L. M. (2004). Hiding data in images by simple LSB substitution. *Pattern Recognition*, vol. 37, no. 3, pp. 469-474.

[13] Celik, M. U., Sharma, G., Tekalp, A. M., & Saber, E. (2005). Lossless Generalized-LSB Data Embedding. *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253-266.

[14] Alghoniemy, M., & Tewfik, A. H. (2006). Progressive Quantized Projection Approach to Data Hiding. *IEEE Transactions on Image Processing*, vol. 15, issue 2, pp. 459-472.

[15] Wang, R. Z., Lin, C. F., & Lin, J. C. (2001). Image hiding by optimal LSB substitution and genetic algorithm. *Pattern recognition*, Elsevier, vol. 34, no. 3, pp. 671-683.

[16] Sheikh, Hamid. R., & Bovik, A. C. (2006). Image Information and Visual Quality. *IEEE Transactions on Image Processing*. vol. 15, no. 2, pp. 430-444.

[17] Shie, S. C., Lin, S. D. & Fang, C. M. (2006). Adaptive data hiding based on SMVQ prediction. *IEICE Transactions on Information and Systems*, vol. 89, no.1, pp. 358-362

[18] Lai, S. H. (2008). A Predictive SMVQ Steganographic Method Using Multiple Classification Codebooks. Master's Thesis, Chaoyang University of Technology, Taichung, Taiwan.

[19] Shie, S. C., Lin, S. D. (2009). Data hiding based on compressed VQ indices of images. *Computer Standards & Interfaces*, Elsevier, vol. 31, no. 6, pp. 1143-1149.