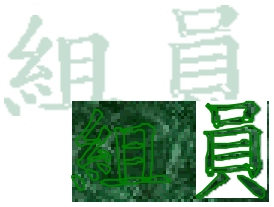
組員： 蔡易達 a8628002　李孟潔 a8628045　曾俊傑 a8628054

目錄



蔡易達 a8628002
李孟潔 a8628045
曾俊傑 a8628054

壹、 摘要

網際網路的蓬勃發展，徹底顛覆了人類資訊取得之方式。加上各種資料的數位化趨勢，使用者可以大量且毫無限制地複製、重製資料，使好資料擁有者的著作權受到很大、的威脅，如何解決這項嚴肅的課題，使用資訊隱藏技術(information hiding)的數位浮水印，(digital watermarks)則被寄予厚望。不過，在本文中，我們將不討論數位浮水印的相關技術，而把焦點鎖定在資訊隱藏技術之探討。

貳、 前言

資訊隱藏是一門非常古老的技術。在西方，最早可追溯至希臘時代。早期有許多傳統的隱藏技術，都是將資訊隱藏於文章中。例如，利用文章中每個句子的奇偶數來代表所隱藏的資訊為 1 或 0;或者從文章中依特定規則抽取字母組成句子。在二次大戰時期，德國間諜就曾用下列文章傳遞訊息。

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suet and vegetable oils.

將上述文章中每個字的第二個字母抽取出來組合，即透露出以下的訊:

Pershing sails form NY June 1.

由於傳統資訊隱藏技術受限於無法隱藏大量資訊與隱藏不易的缺憾，例如某些隱藏技術在嵌入資訊時需要相當程度的語彙或語意等素養。所以在使用上通常都局限於國家級機密情報的秘密通訊(cover communication)上，也正是因為這個原因 資訊隱藏這門技術便較少受到一般人的關注。

近年來，由於電腦技術大幅進步，傳統秘密通訊方式也發生了重大變革，

影像(image)、音訊(audio)、視訊(video)等數位媒體的資料量大，透過電腦的處理，我們可以輕易的將人類感官系統所無法察覺的細微變化分析出來，克服了傳統資訊隱藏技術所面臨無法傳遞大量資料的困境。許多將資訊隱藏於各種數位媒體的相關軟體被陸續開發出來，藉由電腦的幫助，使用者可以輕易地將資訊大量隱藏於各種數位媒體之中。從此，使用者不再局限於特定之情報或軍事單位，一般的商業或個人機密之通訊都可利用這項技術來加強對通訊資料的保護。

資訊隱藏的應用，除了前面所提及的秘密通訊及數位浮水印之外，還可應用在追蹤洩密來源(traitor tracing , fingerprint)、驗證(authentication)、偽品偵測(fraud detection)、影像(image)的說明或影片(video)中的字幕或相關資訊也可以直接嵌入在媒體之中(caption embedding)，而不用另外儲存，增加管理及儲存的負擔，可預期的是更多相關的應用，將如雨後春筍般地，被陸續開發出來，多元化的應用也使得資訊隱藏技術受到前所未有的注意。

所謂的資訊隱藏，就是將資訊隱藏於另一份媒體之中，通常我們稱這個媒體為掩護媒體(cover-media)，隱藏的動作稱為嵌入(embedding)，掩護媒體經嵌入資訊後變成一份偽裝媒體(stego-media)。所以資訊隱藏就是將資訊本身的存在性(existence)隱藏起來，讓人察覺不到;這和密碼學隱藏資訊本身的意，但仍然可以感受到資訊是存在的;資訊隱藏則是從另一個角度切入，將資訊隱藏起來，不讓人感受到。然而發展資訊隱藏技術之目的，並不是取代加解密技術，而是對通訊提供另一層次的保護。所以我們建議資訊在嵌入到掩護媒體之前，最好先經過加密。這樣做的好處是，除了可以保護機密資訊外，更重要的是可以增加嵌入行為的不可偵測性(indefectibility)，這也是這門技術的一項主要需求(requirements)。

參、 系統設計與需求考量

一個成功的資訊隱藏系統，在設計時，必須依其不同的應用，考量不同的的需求。在此，我們將概略地探討一些一般性的需求：

(1)隱蔽性(imperceptibility, virtually undectability)

所謂的隱蔽性是指嵌入的資訊不能降低原來掩護媒體的品質，也就是說偽裝媒體和掩護媒體在人類感覺系統下是不可分辨的，也就是說在偽裝後是我們肉眼無法看出來有何差異性的，這是所有的資訊隱藏系統最基本的需求。

(2)不可偵測性

秘密通訊的行為如果曝光，最直接的後果就是通訊者的身份馬上跟著曝光。因此，利用資訊隱藏技術進行秘密通訊，即使嵌入的資訊由人類的感覺系統無法察覺到，並不代表電腦也分析不出來。舉例來說，媒體本身具有一些與媒體內容相關的性質，當嵌入的行為更改到這些特性時，即有可能在通訊過程中，曝露出秘密通訊的行為。所以在不可偵測性上是要求偽裝媒體是不可被電腦查出差異。

(3)安全性(security)

嵌入資訊後，必需擁有某種程度的相關祕密資訊，才能取出原來嵌入之資訊。一般來說，嵌入演算法要保密並不容易，所以嵌入系統的安全性，就必須建構在通常代表嵌入位置的嵌入金鑰(steog-key)上。利用嵌入金鑰當成是亂數產生器(random number generator)的種籽(seed)，產生一連串的隨機亂數，再配合嵌入演算法來嵌入資訊。因此，從偽裝媒體中萃取資料時，也必需擁有相同的嵌入金鑰才能取出資訊。

(4)容量(Capacity)

容重就走掩護媒體所能嵌入的最大資訊量，一般來說，嵌入的資訊量越多，隱蔽性相對較差，嵌入行為曝光的風險也較大。容量和嵌入技術、系統需求及掩護媒體本身的規格都有很高的相關性。

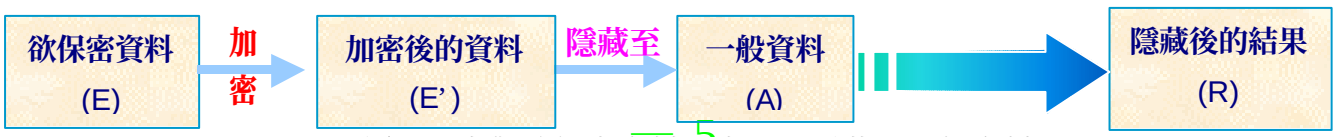
(5)強韌性(Robustness)

強韌性是指資訊在嵌入掩護媒體後，經過壓縮或其它的媒體處理後，是否仍然可以取出原先嵌入的資訊。強韌性是數位浮水印系統所必需滿足的一項基本需求。至於在和密通訊上的應用，通道(channel)管理者，也可以將從通道上通過的媒體另外處理過後再逆出 用以避免或限制通道上可能發生的秘密通訊行為;此時傳送才說必須使用具強韌性的嵌入技術，來確保接收才能收到嵌入的資訊。為對抗不同程度的媒體處理程序，便需考量不同的嵌入技術，一般來說，強韌性質等級較高的嵌入技術，其嵌入量便相對較小，這是不得不付出的代價。

肆、 資訊隱藏的流程

資訊隱藏，故名思義是一種加密的方法，把重要的資訊隱藏到一般資訊中。不同於純粹對重要資訊直接加密，資訊隱藏是藉由一個不會令人起疑的媒介，將機密資料隱藏其中(見圖一)。相對於直接加密，資訊隱藏的方式更具有保密的效果，如果再塔配既有的一些加密方法，將欲保密的資料先作一層加密的動作，再隱藏至一般資料中，則保密效果就更好了，等於有了雙重的保密(見圖二)。

圖一　資訊隱藏的流程



圖二　先對欲保密資料加密之再隱藏至一般資料

伍、 嵌入技術介紹 影像二元樹

由於嵌入技術和掩護媒體的儲存格式有很大的關連性，加上數位媒體的儲存格式又非常繁多，因此在接下來的文章中，我們僅僅介紹是關於將掩護媒體鎖定在數位影像，針對其中的一種嵌入技術進行介紹。

一、前言

隨著網路時代的來臨，資訊的流通越來越便利，地球村不再是遙不可及的夢想。藉由網路，可以隨時隨地掌握全世界各地的資訊:藉由網路，人與人之間的連絡更為方便。但隨之而來的問題，如資訊倫理的問題，網路安全的問題等等，卻不得不讓我們靜下心來思考，尋求它們的解決方案。這篇論文所要探討的，便是屬於網路安全這個領域。

資訊的加解密可以提高資訊在網路上傳送的安全性，但以往我們所使用的加解密方法都有一個共通的缺點，就是新編出來的密文都是一堆看不懂的亂碼。對於有經驗的網路駭客來說，當他們竊取到這一類的檔案，便會知道這是一個加密過的檔案，於是千方百計的想破解:就算是再好的一套加密法，也終會有被破解的一天，這只是時間上的問題罷了。因此，近來有人開始研究另一種資訊安全方法—隱藏資訊，就是把資訊藏在一張看來極為普通的圖片中。以藏圖為例，我們將預藏之圖稱為真圖，而用來傳送、隱藏真圖的圖稱為偽圖。由於偽圖是一張普通的影像，所以不容易被懷疑，因此真圖被截取破壞的機會也就小很多了。

在數位化資料上作隱藏(steganography)之研究雖歷史不久，但已引起大家的注意，而資料隱藏(data hiding)或是資訊隱藏(information hiding)是其中常用的方法，Bender 等人[1]對資料隱藏這方面的技術作了一個整理，並以三種應用，也是資料隱藏技術三種不同的考量角度:智財權保護(copyright

protection)、資料篡改證明(temper-proofing)、與增加隱藏資料(argumentation data embedding)之適用性來評估這些技術。在智慧財產權保護應用中，是將商標或是識別資料與展現資料一起呈現(如浮水印)或是隱藏起來，強調資料的所有權，故隱藏資料或是識別資料要能免於因展現資料修改而有所變化或是不能辨識的情形發生。資料篡改證明是要能證明展現資料是否被別人更改過，例如:在文章中增加或刪除字、或是在照片中增加或刪除人物等，故隱藏資料要能持有展現資料的特性。而增加隱藏資料是在展現資料的特定位置上，增加記載一些我們對展現資料感興趣特徵的描述資料:故只要此特徵存在，增加的隱藏資料亦仍存在，而且可隨特徵因位置或大小改變，而隨之調整。

二、影像二元樹

影像二元樹的作法，是針對一張影像，連續的對其作垂直水平交錯的二等分切割而成。在二元樹中，二等分切割約兩個子區塊，以兩個二元樹節點來表示，並將其放到對應的父節點之下，整棵樹的根節點所表示的是整張影像。切割的順序是先垂直分割，再作水平分割。每一次分割所得的子區塊相子節點對應的方式是這樣的:當垂直分割時，左邊的區塊對應在子節點，右邊的區塊對應右子節點;當水平分割時，下面的區塊對應到左子節點，上面的區塊對應右子節點。這樣由根部一直往下建二元樹，只要同一區塊申 pixel 的色階不是單一的。就持續不斷的進行垂直水平交錯切割的動作，直到同一區塊中 pixel 的色階是單一約為止。在這樣一個二元樹中，包含兩種節點:內部節點和外部節點;內部節點所代表區塊中的 pixel，包含兩種以上的色階:外部節點所代表區塊中的 pixel，僅有一種色階。在講完二元樹的編碼後，將以一張 4*4 的的灰階影像圖為例，如下頁圖(a)，建構出其對應的影像區塊圖，如下頁圖(b)，及影像二元樹，如下頁圖(c)。下頁圖(a)中的數字代表色階值，下頁圖(b)中之英文字母代表切割後的各區塊。

將建構好的二元樹，每一個外部節點以一個二元碼來表示，稱為 bincode。因此編碼後，一張影像就可以用一個 bincode 的有序串列來表示。由於在 bincode 中隱含了一個單一顏色影像區塊的位置、大小及顏色，所以能夠簡化相關的運算及減輕執行時的負擔。

假設影像大小為 2*2，色階數為 M，座標系統(i,j)之原點座標(0,0)設在影像最左下角 pixel，i 代表水平軸座標，j 代表垂直軸座標。給定一個影像二元樹的外部節點，已知此節點對應區塊之最左下角 pixel 座標為(i,j)(即代表此區塊座標)，節點的階層數為 l，色階值為 c，此節點之 bincode 可依照下列步驟獲得:

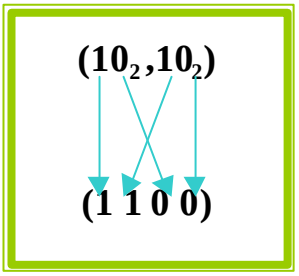
❖ 步驟一：將代表此區塊的(i,j)、階層數l 及色階值c 轉換成二進位表示法。

舉例：以下頁圖(c)中的 I 區塊為例。
I 區塊為：座標(2,2) ， 階層數 2， 色階值 231
$(2_{10},2_{10})\rightarrow(10_2,10_2)$ [取 4 bits，即i 取 2bits，j 取 2bits]
$2_{10}\rightarrow 010_2$ [取 3 bits]
$231_{10}\rightarrow 11100111_2$ [取 8 bits]

❖ 步驟二：將(i,j)之位元交錯排列組合成基點座標(location code)。

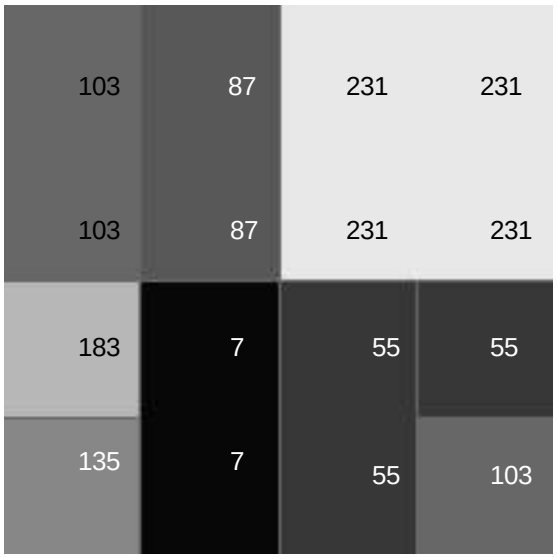
即$(i_{N-1}j_{N-1}\,i_{N-2}j_{N-2}\,\ldots\ldots\ldots\ldots\,i_1j_1\,i_0j_0)$。

舉例：以下頁圖(c)中的 I 區塊為例。
I 區塊為：座標(2,2) ， 改為二進位後為$(10_2,10_2)$
$(10_2,10_2)\rightarrow i_1=1,\ i_0=0,\ j_1=1,\ j_0=0$
➔➔ (1100) <<可參考右圖>>
(1100)則為基點座標

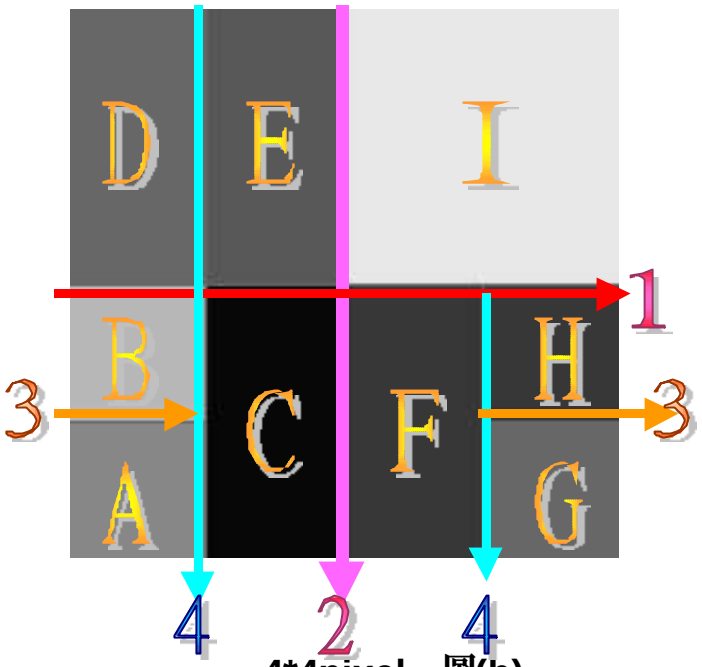


❖ 步驟三：將基點座標配上代表影像區塊的階層數和色階之二進位碼即可得此節點的 bincode。

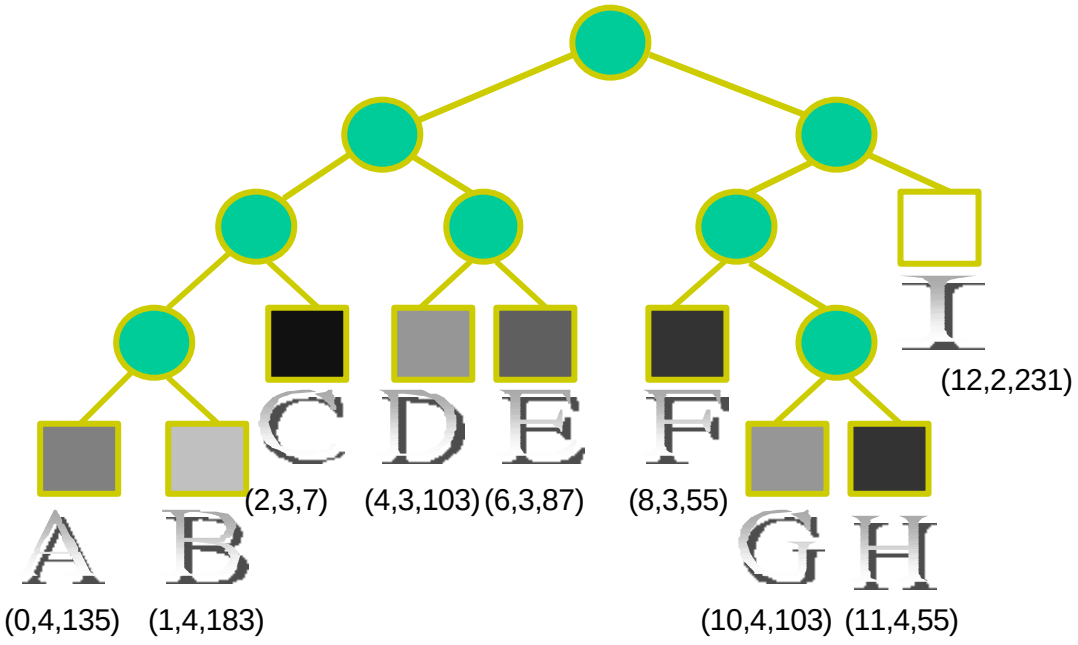
舉例：以下頁圖(c)中的 I 區塊為例。
I 區塊的 bincode 為(1100 010 11100111)



4*4pixel 圖(a)



4*4pixel 圖(b)



影像二元樹 圖(C)

舉例來說：如在上頁圖(b)中我們可以求出代表 C 區塊的座標為(1,0)，階層數=3，色階值=7。依照步驟一轉換成二進位碼為(01,00)，階層數 011，色階值 00000111，再依步驟二求得基點座標(0010)，依步驟三求得代表 B 區塊的 bincode 為(001001100000111)。在此後的文章中，為了描述上的方便，將以區塊的基點座標、階層數與色階結合為數對(基點座標(i,j)，階層數(l)，色階值(c))來表示區塊的編碼。

上頁圖(a)是一張原始的影像圖，大小為 4*4. 各 pixel 中加入數字來表示其色階值。在經過二元分割後，所得到的分割區塊最終如上頁圖(b)所示，我們也在圖上以 1.2.3.4 不同顏色的線條之方式表示切割的順序。上頁圖(c)便是建構好的影像二元樹，圓形代表的是內部節點，即色階並非單一的區塊點，方形的是外部節點，即色階是單一的最終分割區塊。整棵二元樹依深先或是廣先走訪的順序列出外部節點串列，結果是一樣的。以英文字母將其循序編號，一一對應到上頁圖(b)的影像區塊中，並在方形中加上顏色，對應影像區塊的顏色。編號後的三組數字表示其 bincode 之(基點座標，階層數，色階)，以十進位數表示。

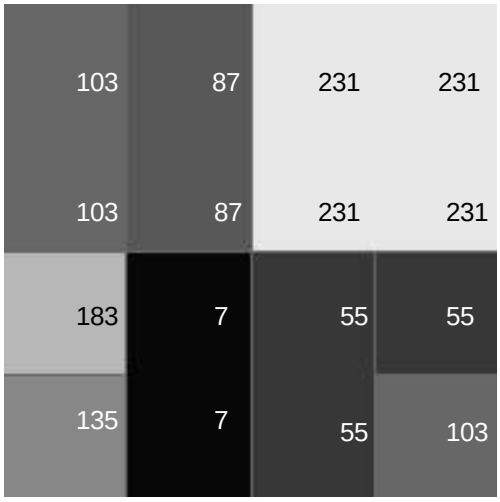
而在這裡我們值得注意的是在於對影像區塊的編號，並不是一定得要如此編排 A~I，共要編出其 bincode 即可，在這裡我們先採用此一編號方式。

我們整理上面所述的內容，可以將整個影像二元樹的建立重點描述於下：

- 是針對真圖來進行，即是卻保密資料 E。
- 切割方式是先垂直切割，再水平切割。
- 影像二元樹的建立是依據切割結果而來，且最後的影像區塊的色階值是單一的，並對每一個影像區塊給予編號。
- bincode 的編製是由(基點座標，階層數，色階)三組值構成的。

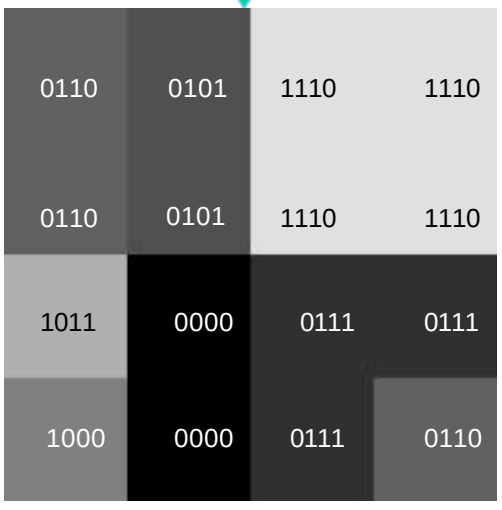
三、影像平滑

如果真圖的顏色複雜度極高，使得我們將真圖建構成影像二元樹後，其外部節點數會相當的多，有時甚至其外部節點數僅略小於整張圖的 pixel 總數，這樣便失去利用影像二元樹編碼的意義。因此，為了避免上述的問題，必需讓色階數目變少。 改變影像中各 pixel 的色階值，讓影像看起來較柔和，稱為影像的平滑。當我們作平滑時，若色階改變的程度不大，通常影像改變前與改變後，是很難用肉眼分辨出來的。因此，可以利用平滑的技巧將真圖色階改變，讓真圖影像二元樹的外部節點數大量減少，如此需要隱藏的資訊量也隨之變少了。



4*4pixel 圖(a)

取各色階二進位表示法中比較高的 4 個 bit



4*4pixel 圖(d)

以 256 色階影像作 16 色階平滑為例，原圖上任一 pixel 的色階值若介於 0-15 之間則將其全部改為 7(取 0-15 的中間值),16-31 則改為 23(取 16-31 的中間值)，依此類推，將整張圖利用 16 色階來平滑重建。如此一來，原圖上任一 pixel 的色階值，不論其低 4 位元的值是什麼，一律會被改成 0111。也就是其低 4 位元的值變得不重要了，其色階的判斷完全由其高 4 位元的值來決定。因此在儲存色階值時，就只需要儲存高 4 位元就夠了，因而減少了資料儲存的長度。我們以圖(a)為例進行影像平滑

處理得到圖(d)，而我們幾乎是看不出圖(a)與圖(d)之間有何差異。

我們將真圖或偽圖作一色階平滑的動作，因此平滑後之影像品質與原圖差距有多少，需有一套衡量的標準。我們採用訊號雜訊比(Pink Signal to Noise Ratio，簡稱 PSNR，也可以稱為忠誠度)，令 f 表示原訊號，f 表示平滑後的訊號，N 表示訊號的長度，則：

$$PSNR = 10\log_{10}\left[\frac{\sum_{i-1}^{N} 256^2}{\sum_{i=1}^{N}(f_i - f_i)^2}\right]$$

由此公式可知，PSNR 值越小表示平滑影像之品質越差，失真情況越嚴重:PSNR 值越大，表示平滑影像之品質越好，失真情況越不嚴重。一般而言，PSNR 值大於 30 的話，影像失真的狀態人的肉眼就不易分辨出來:故希望能讓所有平滑後的影像，其 PSNR 之值大於 30。

以圖(a)➔圖(b)為例來計算其 PSNR 值：

	圖(a) 影像區塊色階值	圖(b) 影像區塊色階值
A	10000111	10000000
B	10110111	10110000
C	00000111	00000000
D	01100111	01100000
E	01010111	01010000
F	00110111	00110000
G	01100111	01100000
H	00110111	00110000
I	11100111	11100000

$PSNR = 10\ \log_{10}[9*256^2/9*7^2]$

$= 10\ \log_{10}[65536/49]$

$= 10\ \log_{10}[1337.47]$

$= 10 * 3.16$

$= 31.6 > 30$

四、資料隱藏偽裝

我們將真圖利用影像二元樹加以編碼，使需要隱藏的單位由 pixel 變成影像區塊;而偽圖的 pixel 便可以對應到此影像區塊，減少所需要用到的偽圖 pixel 數，避免偽圖色階不足的問題。但真圖的色階可能非常的複雜，使得我們切割出來的影像區塊只比影像總 pixel 數少一點，這樣便失去利用影像二元樹編碼的意義，所以要將真圖作平滑，以減少色階的變化，使切割出來的影像區塊變少，如此需要隱藏的資訊量也隨之變少了，而在 codebook 的設計上，我們將 codebook 中的每一個 entry 來配合偽圖上的每一個 pixel，記錄其對應的影像區塊二元樹編碼，以及色階的差值。因為資料隱藏的方法不同以及隱藏資料量的多寡不同，在 codebook 的設計上就有所不同。因此我們設計了四種不同的作法。

❖ 第一種作法

將真圖先作影像平滑處理後，利用影像二元樹的作法將它切割成影像區塊，並在偽圖上，利用連續的 pixel 來對應這些影像區塊。而 codebook 中，依序的用每一個 entry，來記錄這些連續 pixel 對應的影像區塊二元樹碼，以及真圖影像區塊和偽圖 pixel 的色階差值。

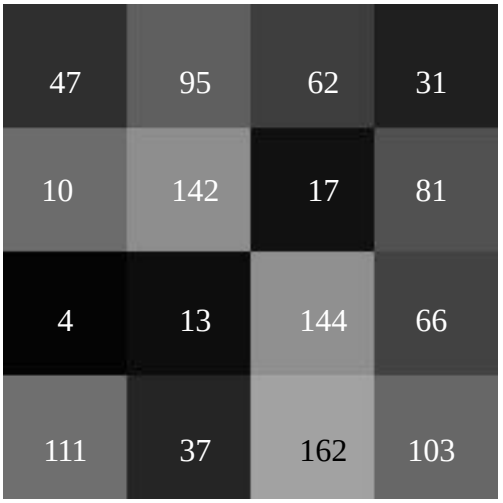
(一)編碼過程：

- (1) 將真圖平滑：用色階平滑的技巧將真圖重建，為求其不嚴重失真，我們採用 16 色階的色階平滑(圖 d)。
- (2) 建構影像二元樹：平滑後的真圖，將其建構成影像二元樹，此時外部節點數會比原圖的外部節點數少很多。
- (3) 建立 codebook：選定一張偽圖，只要其總 pixel 數比步

驟一建立好的影像二元樹的外部節點總數多就好。接下來建立一個 codebook，用來記錄真圖的影像區塊與偽圖的每一個 pixel 間的對照關係。Codebook 為一表格，由偽圖最左上角的 pixel 開始，由左到右，由上至下，依序的用每一列來記錄一個 pixel 所對應 真圖影像區塊的資訊。每一列都有三個欄位，第一欄存放對應的真圖影像區塊的基點座標值，第二欄存放對應的真圖影像區塊，其在影像二元樹中的階層數，由此階層數就可以知道該影像區塊的大小，第三個欄位則存放色階的差值，色階的差值等於 pixel 的色階值減掉對應影像區塊的色階值，如下表一。

- (4) 傳送資料：分開傳送偽圖及 codebook 給接收者。

4*4pixel的真圖 圖(a)



4*4pixel 偽圖 圖(e)

偽圖 pixel 的編號	偽 圖 pixel 之色階值	對應真圖的影像區塊編號	基 點 座 標 (*)	階 層 數 (*)	真圖影像區塊色階值	色 階 的 差 值 (*)
1	47	A	0000 (0)	100 (4)	135	-88
2	95	B	0001 (1)	100 (4)	183	-88
3	62	C	0010 (2)	011 (3)	7	55
4	31	D	0010 (4)	011 (3)	103	-72
5	108	E	0100 (6)	011 (3)	87	21
6	142	F	1000 (8)	011 (3)	55	87
7	17	G	1010 (10)	100 (4)	103	-89
8	81	H	1011 (11)	100 (4)	55	26
9	4	I	1100 (12)	010 (2)	231	-227

表 一 真 圖 與 偽 圖 關 係 對 照 表
(為方便對照，所以在基點座標和階層數以二進位表示)

- (二)解碼過程:

解碼時，利用偽圖及 codebook 這兩樣資訊，將真圖還原，由偽圖最左上角的 pixel 開始，依下列步驟還原。

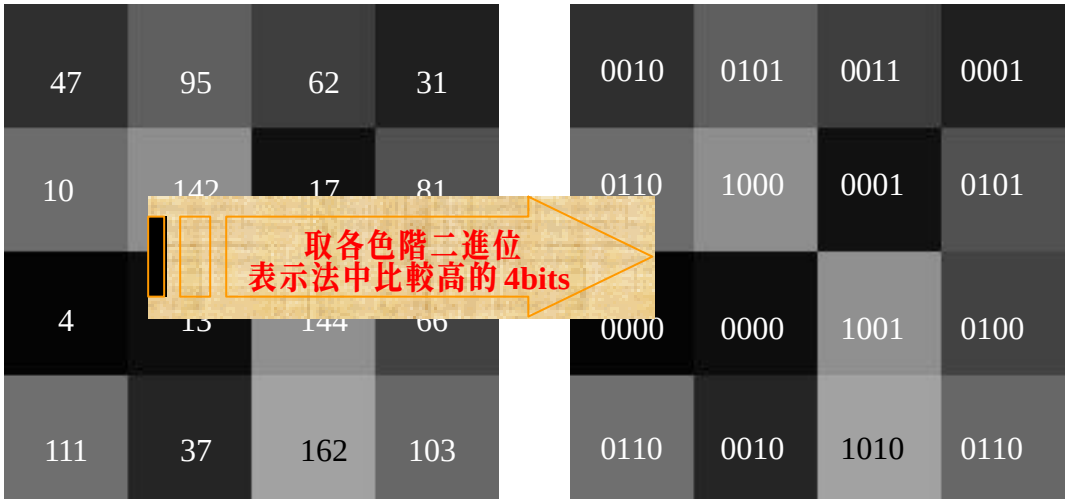
- (l) 讀取偽圖 pixel 的色階值。
- (2) 計算真圖影像區塊的色階值：將步驟(1)計算出來的色階值，減掉在 codebook 中該 pixel 那一列第三欄的色階差值，即可得到對應影像區塊真正的色階值.在 codebook 中加入第四欄，存放這個真圖影像區塊的色階值。
- (3) 重複步驟(1)、(2)。依序處理每一個 pixel，直到整個 codebook 的每一列都處理完為止。
- (4) 還原真圖：用 codebook 中的第一、二、四欄的資訊，即可逐步還原出真圖。

此種作法的優點為:由於偽圖完全不失真，所以即使被攔截到此張圖，也不容易被懷疑，且偽圖及 codebook 分開存放，被破解還原成真圖的機會變很小了，以下以一個簡單的例子來說明這種作法:我們將上頁圖(a)視為平滑後之真圖，而偽圖如上頁圖(e)所示，大小亦為 4*4，在每個 pixel 上標示其色階值。於是，依照上述編碼的步驟. ，我們可以求得一張真圖與偽圖關係對照表，如表一所示。

而我們所要的 codebook，並不要那麼多的欄位，因為偽圖的 pixel 編號和真圖的區塊編號都是循序的，所以並不用寫在 codebook 中，而偽圖顏色之色階值我們可直接從偽圖中讀出，也不需記錄，真圖區塊顏色的色階值則是利用讀出之偽圖 pixel 顏色之色階值再配合顏色色階差值作運算即可。因此，真正需要傳送之 codebook 內容便只是表一欄位名稱中標示有(*)號的欄位。當接收者拿到偽圖及 codebook 這兩份資訊，便可依上述之解碼方法，將真圖還原。

❖ 2.第二種作法

當我們在作 16 色階的色階平滑時，人眼很難分辨其真偽;既然如此，偽圖各 pixel 色階值的後 4 個 bit 就顯得不那麼重要了，即使將其更改過，由於失真不嚴重，也不容易讓人起疑心。因此，在本作法裡，我們直接將 pixel 對應的真圖影像區塊顏色存放在 pixel 色階值的後 4 個 bit 中，codebook 也就不用放色階的差值了，可以大大減少 codebook 的大小。



原始偽圖各 pixel 的色階值　　偽圖各 pixel 的較高的 4 個 bit

圖 f　偽圖處理過程

(一)編碼過程：

(1) 處理真圖及偽圖：先將真圖和偽圖作色階平滑的處理，但是此時偽圖的色階平滑，不需取其該組色階的代表值，只要將每一個 pixel 的較高位元 4bit 保留就好(圖 f)。作完色階平滑的動作以後，對於真圖，還需要將其建成影像二元樹，而對於偽圖則不需要。

- (2) 儲存真圖影像區塊色階的資訊：將真圖的影像區塊顏色存入偽圖的較低 4bit 中。儲存的資料是影像區塊顏色所在的色階組別數；也就是色階值較高 4bit 的值。儲存的 pixel 順序亦同作法一。
- (3) 建立 codebook：建立 codebook，由於真圖的區塊顏色已經存入偽圖的 pixel 較低 4bit 中，所以便不需在 codebook 中存顏色的差值。因此，在 codebook 中，僅需記錄對應的真圖影像區塊的基點座標及階層數這兩欄的資訊。
- (4) 傳送資料：分開傳送偽圖及 codebook 給接收者。

(二)解碼過程：

- 依序在偽圖中，取出每一個 pixel 之色階值中較低的 4 個 bit，由此可以得到真圖中影像區塊的色階組別。從 codebook 中找出對應的影像區塊基點座標及階層數，便可逐步還原整張真圖。

此種作法的優點有：由於 codebook 少了顏色差值這一欄，所以整個 codebook 就會變得比較小。而且真圖影像區塊的顏色可直接由偽圖的較低 4bit 取得，不需要經過額外的運算，可以節省不少時間，以下以一個簡單的例子來說明這種作法:為了方便讀者比較，偽圖仍延用作法一的例子，而真圖則作些許

00101111	01011111	00111110	00011111
01101100	10001110	00010001	01010001
00000100	00001101	10010000	01000010
01101111	00100101	10100010	01100111

偽圖原色階值，較低 4bit 以低線標示

00101000 (A)	01011011 (B)	00110000 (C)	00010110 (D)
01100101 (E)	10000011 (F)	00010110 (G)	01010011 (H)
00001110 (I)	0000****	1001****	0100****
0110****	0010****	1010****	0110****

圖 g 偽圖各個 pixel 色階之二進位值，較低 4bit 已由真圖圖(d)的 4bit 取代，隱藏了真圖區塊顏笆的資訊，對應真圖影像區塊為後面括號中英文字母。又因為對應的只有 A~I 九個值，所以之後的以*表示的資料視為 don't care

修改，但其平滑後所切割成的影像區塊，則仍延用作法一。各個 pixel 不繪出顏色、直接將其 256 色階的色階值寫出。

依步驟(1)所示，分別對真圖及偽圖作處理。當真圖偽圖都處理完後，便可以做步驟(2)所示，將真圖的影像區塊顏色組別，依序加入偽圖的 pixel 較低 4bit 中，圖 g 表示隱藏真圖的影像區塊後，偽圖各 pixel 的色階值(以二進位表示)，後面幾個 pixel 自於沒有對應的區塊，因此後 4bit 可以不必處理。

❖ 第三種作法

延續修改偽圖顏色的構想，我們既然可以將真圖色階藏於偽圖 pixel 的後 4bit 中，當然也可以利用這來藏文字。一個字元的 ASCII 碼有 8 個 bit，可藏於兩個 pixel 中。因此，我們可以將欲隱藏的文字資料藏在偽圖的後 4 個 bit 中。將這張藏有文字資料的偽圖視為是新的偽圖，然後就依照作法一，利用色階的差值，將真圖藏到偽圖中。利用此原理，在同一張圖中可以藏圖也可以同時藏文字。

(一)編碼過程：

(1) 處理真圖：真圖的處理與作法一的步驟(1)完全相同。

	0	1	2	3	4	5	6	7	8	9
0	\000	\001	\002	\003	\004	\005	\006	\a	\b	\t
10	\n	\v	\f	\r	\016	\017	\020	\021	\022	\023
20	\024	\025	\026	\027	\030	\031	\032	\033	\034	\035
30	\036	\037	Space	!	“	#	$	%	&	‘
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w

ASCII code 表二

的色階值，建立一個三欄位的 codebook，內容同作法一的 codebook。不過還要多加一項資訊。就是欲傳文字的總字元數。

- (4) 傳送資料：分開傳送偽圖及 codebook 給接收者。

(二)解碼過程：

- (1) 真圖的解碼過程：同作法一的解碼過程。
- (2)文字的解碼過程：從偽圖最左上角的 pixel 起，連續兩個偽圖 pixel 為一組，取出兩個 pixel 的色階較低四位元(4bit)，合成為 8bit，對照 ASCII 表可得此 8bit 所代表的文字：重覆此作法，直到所得到字元等於 codebook 中所告知的文字總字元數為止。

與前兩種作法比較，此種作法的優點優是多了隱藏文字的功能。

以下以一個簡單的例子來說明這種作法：我們所使用的真圖與偽圖仍沿用第二種作法，將真圖與偽圖的進行處理，我們以藏 GOOD 為例說明，先將 GOOD 透過上頁表二轉換成其所代表的十進位 ASCII 碼得到(71,79,79,68)，再轉成二進位的 ASCII 碼得到(01000111, 01001111,01001111,01000100)，將此四個 8bit 折成 8 個 4bit 得到(0100,0111,0100,1111,0100,1111,0100, 0100)，依序將這 8 個二進位碼存人處理好的偽圖較低位元的 4bit 中，結果如圖 h 所示，而 codebook 的內容則包含表三中標示有(*)號的欄位。

偽圖 pixel 對應偽圖之對應真圖的影基點座標	階層數	真圖影像區色階的差值	
00100100 (A)	01010111 (B)	00110100 (C)	00011111 (D)
01100100 (E)	10001111 (F)	00010100 (G)	01010100 (H)
00000100 (I)	0000****	1001****	0100****
0110****	0010****	1010****	0110****

圖 h 欲傳給偽圖各個 pixel 色階之二進位值，值低 4bit 加底線表示，其隱含了欲隱藏文字的資訊。其中*號代表 don't care。

的編號	pixel的色階值	像區塊編號	(*)	(*)	塊色階值	(*)
1	36	A	0000 (0)	100 (4)	135	-99
2	87	B	0001 (1)	100 (4)	183	-96
3	82	C	0010 (2)	011 (3)	7	45
4	31	D	0010 (4)	011 (3)	103	-72
5	100	E	0100 (6)	011 (3)	87	13
6	143	F	1000 (8)	011 (3)	55	88
7	20	G	1010 (10)	100 (4)	103	-83
8	84	H	1011 (11)	100 (4)	55	29
9	4	I	1100 (12)	010 (2)	231	-227

表三 真圖與偽圖關係對照表

❖ **4.第四種作法**

通常我們需要隱藏的文字資料都不會太多，因此我們可以在偽圖的不同的部份，分別儲存文字資料和影像資料。不僅可以使得偽圖的每一個 pixel 被充份的利用，而且由於真圖的區塊顏色已經存入偽圖的 pixel 較低 4bit 中，所以便不需在 codebook 中存顏色的差值，減少了 codebook 的大小。

- (一)編碼過程：
  - (1) 處理真圖及偽圖：與作法二的步驟(1)完全相同。
  - (2) 儲存真圖影像區塊色階的資訊：與作法二的步驟(2)完全相同。
  - (3) 將字元的 ASCII 碼存入偽圖 pixel 的後 4bit 中：與作法三的步驟(2)完全相同。
  - (4) 建立 codebook:在 codebook 中，除了需記錄對應的真圖影像區塊的基點座標及階層數這兩欄的資訊以外，還需要記錄欲傳送文字資料的數目、文字資料與影像資料在偽圖內的起始位置。
  - (5) 傳送資料：分開傳送偽圖及 codebook 給接收者。圖 i 偽圖各個 pixel 色階之二進位值，較低 4 位元加底線表示，其隱含了真圖區塊顏色的資訊，對應的真圖影像區塊為後面括號

中的英文字母。後面 6 個 pixel 則儲存了 GOD 這個英文字的 ASCII 碼。其中*號代表 don't care。

00101000 (A)	01011011 (B)	00110000 (C)	00010110 (D)
01100101 (E)	10000011 (F)	00010110 (G)	01010011 (H)
00001110 (I)	00000100	10010111	01000100
01101111	00100100	10100100	0110****

圖 i 偽圖各個 pixel 色階之二進位值，較低的 4bit 加底線表示，其隱藏了真圖區塊顏色的資訊，對應的是真圖影像區塊為面括中的英文字母。後面 6 個 pixel 則儲存了 GOD 這個英文字的 ASCII 碼，其中*代表 don’ t care

五、四種作法的比較

我們看了四種作法後，我們將四種作法的特色一一列出：

- 第一種作法：真偽圖皆不失真。
- 第二種作法： 真圖隱藏在偽圖的色階值之中。
- 第三種作法： 文字隱藏在偽圖的色階值之中。
- 第四種作法： 將真圖與文字的資訊隱藏在偽圖的色階值之中。

從上述以及表四我們可以知道第四種作法是一個比較好的方式，雖然第四種作法與第三種作法之間的差別只在 codebook 所需傳送的資料值，第三種作法需要傳送顏色色階的差值，而第四種作法需要傳送的是文字資料，但實際上文字資料比色階差處理上來說是較省時，且資料量是較小的，因此比較之後是第四種作法是較好的。

	影像平滑	隱藏文字	Codebook 大小	結果排序
第一種作法	╳	╳	3 欄	4
第二種作法	○	╳	2 欄	3
第三種作法	○	○	3 欄	2
第四種作法	○	○	2 欄+文字資料	1

表四　四種作法的比較

我們從整個方法上來看，我們採用影像二元樹的技後進行資訊隱藏可以有以下優點：

- 優點：
  - 1. 真圖與偽圖的色階分布並沒有任何限制，不需要色階分布相似。

- 2. 真圖的大小並沒有限制，只要編碼後的影像區塊夠少，可以用相當小的偽圖來隱藏真圖，所以小圖帶大圖是可行的。
- 3. 作業方式比較容易，並不需要作色階的平移。(而所謂的色階平移技術目的在解決色階不足的問題，因此要作色階調整會增加處理時間。)
- 4. codebook 對應偽圖的 pixel 是連續的，因此可以省去 search 的時間。

此外我們在這理探討的是灰階的影像，當我們面對的是 RGB 彩色圖片時，並不代表不能進行該技術，我們可以透過將 bincode 中的色階值 c 改成 R、G、B 三個值，且每一個也都是用 4bit 來表示，也就是說將色階值原 4bit 增加為色彩值 12bit 來紀錄 RGB 三個數值，不過相對的資料量會變大，處理時間也會花較多，不過我們仍可以依這裡所介紹的技術來進行資訊隱藏。

陸、 結論

在資訊隱藏這個領域中，有許許多多的技術，我們今又介紹的僅是其中的一種，對於其他的技術並沒有深入的了解，因此無法比較今又所介紹的與其他技術之比較，但經由了解這一個影像二元樹的資訊隱藏技術後，使我們真正踏入資訊隱藏這個領域，了解其中的奧秘，並且能夠真正的來進行這一個技術，這是我們在這一次報告中獲得的最大的收獲了。

柒、 參考資料

資訊安全通訊 五卷第四期 88.09

數位影像之資訊隱藏技術探討

李遠坤陳玲慧

交通大學資訊科學系自動化資訊處理實驗室

http://debut. cis. nctu. edu. tw

技術學刊 第十五卷 第三期 民國八十九年

Journey /of Technology, Vol. 15, No.3, pp . 36372(2000)

資料隱藏偽裝技術之研究

侯永昌 趙元甫

國立中央大學資訊管理系

陳培敏

東吳大學資訊科學系

中央研究院計算中心通訊 第十六卷 第二十二期 89/10/23

資訊隱藏與隱像術

莊樹諄
