長榮大學資訊工程學系專案實作成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　※
※　　　　　　以向量量化法實作影像資料隱藏　　　　　※
※　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

專 案 編 號 ：CJCU－CSIE－PRJ－2009－02

執 行 期 間 ：98 年 02 月 01 日至 99 年 01 月 09 日

專 案 成 員 ：蔡昇倫、李瑜晨、房彤諺、鄭鶴文

指 導 老 師 ：吳永基

中 華 民 國　98　年　11　月　07　日

長榮大學資訊工程學系專案實作

以向量量化法實作影像資料隱藏

The Implementation of Image Data Hiding by Vector Quantization

專 案 編 號 ：CJCU-CSIE- PRJ-2009-02
執 行 期 間 ：98 年 02 月 01 日至 99 年 01 月 09 日
參 與 人 員 ：蔡昇倫、李瑜晨、房彤諺、鄭鶴文
指 導 老 師 ：吳永基

中文摘要

隨著電腦與網路的進步，數位資料可以快速的藉由網路傳輸到各個地方，隨之而來的就是安全的問題，例如：智慧財產權。數位的年代，由於資料的數位化的趨勢以及網際網路的發展，資訊的傳播變得更容易與快速，資訊的取得也更加方便。然而，傳播中的資訊容易遭到複製與竄改，其中就牽扯到是否侵犯智慧財產權的問題。

加入「數位浮水印」的方法有很多種，每種加入「數位浮水印」的方法都會影響到加密後的影像品質。本論文將會使用的方法為利用向量量化(Vector Quantization)的方法來加入「數位浮水印」達到資料隱藏的目的，此種加入「數位浮水印」後的數位浮水印是屬於「看不見的數位浮水印」。

本論文在實作影像資料隱藏時會用到向量量化法，普通的向量量化法執行的時間較久，而我們使用向量平均法和 Partial Distortion Searching (PDS)來加速向量量化法的速度，本論文中對 512x 512 的 LENA 圖執行 VQ,原本需要約 1 分鐘的執行時間，而現在剩下約 1.6 秒。浮水印實驗的對象我們採用黑白、灰階、彩色和文字等浮水印嵌入影像，而強韌性的測試我們使用「PHOTOSHOP」程式來對嵌入後的影像進行銳利化、挖空以及添加字等破壞，之後取出浮水印看是否可以辨識出該浮水印，實驗的結果顯示我們嵌入後的影像對以上三種破壞的強韌性不錯。

關鍵詞：向量量化法、資料隱藏、PDS。

Abstract

With the growing of computer and network, digital data can be spread to anywhere in the word quickly. In addition, digital data can also be copied or tampered easily so that the security becomes an important topic in the protection of digital data. Digital watermark is a method to protect the ownership of digital data. Embedding the watermark will influence the quality certainly. In this project, Vector Quantization (VQ) is used to compress and

embed the watermark to fulfill data hiding. This kind of watermarking is invisible which means the users will not conscious the existing of embedded watermark. However, because VQ needs a lot of computation burden so that we develop a fast VQ encoding scheme by the average and partial distortion searching (PDS) to speed up the encoding. The watermarks we hide to the image include gray, bi-level and color images. Texts are also can be regarded as watermark to embed to the image. In order to test the robustness of the system, we adopt Photoshop to fulfill sharpen, cropping and altering to check if the extracted watermark is still recognizable. Experimental results demonstrate that the proposed system can resist the three kind of tampering in general cases.

Keyword ： Vector Quantization 、 Data Hiding、PDS

一、簡介

1.1 背景與動機

四千多年來，數不盡的保密技術，此消彼長。過去這些技術大多只侷限在國防、軍事用途和外交等方面之應用。近年來，由於工商界蓬勃的發展，急需使用電腦做為傳送郵件之主要工具。因此，電子資訊傳輸將是未來資訊工業發展的重要方向。由於敏感的資料，無論是儲存在電腦記憶體中，或者正透過通訊設備傳送的過程裡，都極有可能會遭遇到不法者的盜竊、拷貝、偷聽等，而從中獲取不法之利益。要去除這種威脅，一個逐漸被採用的做法是利用密碼轉換的方法將這種敏感的資料加以「改頭換面」，藉以掩飾其原來的面目，使得縱然這些資料在儲存或傳送過程中，一旦落入他人手中，不法者亦無法了解其原意，更無法因此而造成傷害。

而當前社會由於網路的快速發展，不論任何資料都可以自由的在網路上傳輸，然而伴隨而來的就是安全的問題，如何安全又保密的保護要傳輸的資料是目前最重要的課題之一。無論是小至個人的照片或者和他人交談的消息，大到關係國家機密的軍事機密，都是必須要保密而不被他人所知道，正因為保護數位資料是很重要的，所以我們選了「以向量量化法實做影像資料隱藏」這個題目來針對影像資料的隱藏來做深入的了解以及測試。

1.2 向量量化法 (Vector Quantization)

VQ (Vector Quantization)是一種非常基本的失真影像壓縮法。改良 VQ 的壓縮結果將會對其他的相關影像壓縮技術有所幫助，故 VQ 是學術界最廣被用來研究影像壓縮的重要格式。傳統 VQ 的基本作法首先將壓縮的影像分割成許多大小相同的小方格。例如一張 512 × 512 點的影像，我們通常會將它分割成 128 × 128 個 4 × 4 點的小方格。按著查詢事先完成的編碼簿(Codebook)，找出跟每一個影像方格最接近(即最相似) 的編碼字(Codeword)。然

後，再利用這些最接近的編碼字之索引值，組成一張索引表，如此即完成影像的壓縮。這張索引表即是 VQ 壓縮後的結果，因為索引表的體積通常會比原影像小方格的體積小很多，故 VQ 能有很好的壓縮效果。至於影像還原，由於 VQ 解碼器可以利用壓縮後的索引表找出每個影像小方格的最相似編碼字，故可還原出原始影像來。雖然還原的原始影像與真正的原始影像並非完全相同，但必定十分相似。一般而言，VQ 的影像品質決定於編碼簿內編碼字的數量之多寡及代表性之優劣。總之，一本大小適中且內容良好的編碼簿才可以確保 VQ 壓縮的影像品質。

正式地說，一個尺寸為 K 的向量量化和大小為 N 的編碼簿可以被看成一個影像 Q 從一個尺寸為 K 的空間 R^K 到一個有限的子集合 Y。也就是說 Q：$R^K \rightarrow Y$，$Y=\{y_i \mid i=1,2,.....,N\}$ 包含 N 個編碼字被叫作編碼簿，y_i 表示 Y 的第 i 個編碼字，每一個 y_i 的尺寸都為 k。如果失真的測量法用來選擇最接近的編碼字給每一個輸入尺寸為 K 的向量 $X=(X_1, X_2, X_3, ..., X_k)$ 在編碼簿中是歐幾里德距離，X 和編碼字 y_i 間的失真的計算公式如下：$d(x,y_i)=\sum_{j=1}^{k}(x_j-y_{ij})\times(x_j-y_{ij})$

最接近的編碼字 y_{bm}（有最小的距離）是經由下面公式：$d(x,y_{bm})=\min d(x,y_i) \quad i=1,2,...,N$ 當 bm=i。傳遞或儲存 y_{bm} 的索引值來取代整個輸入的向量 X 並且達到壓縮的目的。

和普通 VQ 的 bit 比例為：$\frac{\log_2 N}{k}$ bits per pixel(bpp)。

1.3 浮水印

由於電腦科技的快速發展，使得多媒體的機密和防偽保護越來越受到重視，所以就需要「浮水印」來達到機密和防偽的保護。以前，畫家如果要保護自己作品的智慧財產權的話，就會在作品上「簽名」或「蓋章」以表示作品的真偽，如果要去除簽名或蓋章就會破壞到作品。但是現在的電子影像易複製、易修改的特性使得差異性極小，肉眼難以分辨，所以防偽的難度較以往高。如何加入簽名而不破壞作品本身，並且在資料傳輸的過程不讓資料有所遺失，並且有一定程度的「強韌性」來防止惡意攻擊者的攻擊。

最普遍最常見的浮水印就是鈔票上的防偽浮水印，只要用燈光檢視就可以在鈔票的空白處發現防偽圖樣。有些公司或廠商會將商標覆蓋在影像上來達到防偽的效果，但是這種浮水印的效果不好，因為加入的商標若太大則會影響影像的美觀，但若是太小則「強韌性」不夠而容易被惡意攻擊者去除商標。

所以保護電子影像的最好方法就是加入「數位浮水印」，所謂「數位浮水印」指的是將浮水印的技術運用在數位媒體中，

這些數位媒體包含數位化的影像(如JPEG)、聲音(如 MP3)及影片(如 DVD)等。為了防止這些數位媒體在網路上遭人下載非法使用，我們可以將一些具代表性的圖樣（如註冊商標、個人照片），利用相關的技術植入這些數位媒體中，來證明其合法的持有者，進而保護智慧財產權。當有人下載這些已經植入「數位浮水印」的數位媒體並且非法使用時，我們為了證明其合法的持有者是誰，可以利用相關的技術將此數位媒體中的「數位浮水印」擷取出來，用以證明其合法的持有者是誰，藉以保護作品的智慧財產權。

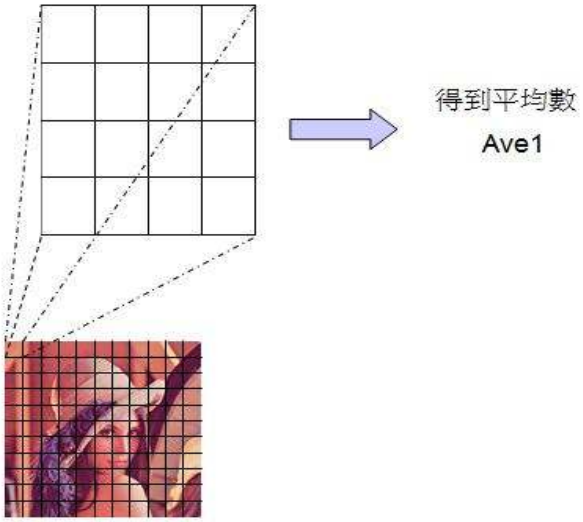
數位浮水印的特性就是利用肉眼難以辨識極細微的變化，而加入無法察覺的特殊資訊，即使經過壓縮或傳輸後也可以將所加入的特殊資訊取出來達到防偽的功能。

二、研究方法

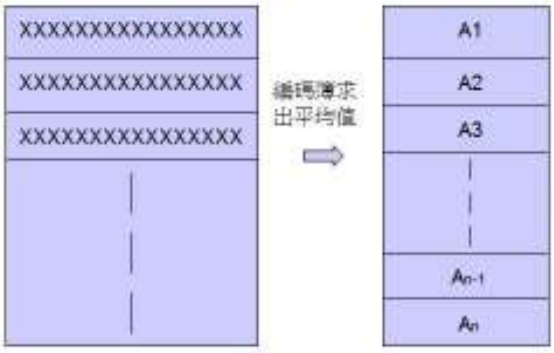
2.1 快速 VQ

(一)平均值

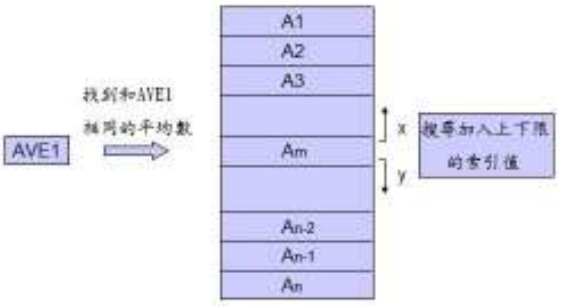
首先將原始影像分割成 4 × 4 個區塊後，將每 4 × 4 區塊裡面的 16 個向量求出平均。同時編碼簿每個索引值代表的 16 個向量也求出平均。然後將兩者的平均做比對後，找出和編碼簿索引值相同的平均數，如此一來搜尋編碼簿計算誤差的範圍就可以從原本需要找整本編碼簿而轉變成只需要搜尋編碼簿所對應到的索引值加上一個上限和下限此範圍的索引值，如此一來搜尋的範圍縮小自然所需要的時間也就減少。



（圖 2-1　原始影像求平均值）



（圖 2-2　編碼簿求平均值）



（圖 2-3　找到相同的平均數並取得該索引值加上上下限的範圍）

(二)局部失真搜尋

(Partial Distortion Searching)

由於 VQ 計算誤差時只會取比目前誤差小

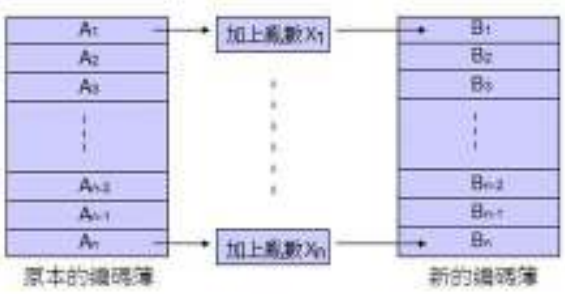
的索引值，所以當計算誤差時現在的誤差已經比目前最小的誤差還大的時候，就不再計算該索引值和影像的誤差，因為已經比最小的誤差大了，再計算下去就是無意義且浪費時間。傳統 VQ 壓縮系統花費許多時間在搜尋最佳的編碼字，從第一個編碼字搜尋到最後一個編碼字，則計算總共需要 編碼簿的尺寸 × 輸入的向量尺寸 個計算。另外，編碼一個影像的計算總共會花費 區塊數目 × 輸入的向量尺寸 × 編碼簿的尺寸的時間。PDS 演算法可以被用來加速計算過程。PDS 裡，當目前計算所累積的局部誤差超過目前最小的誤差則會被拒絕。使用以上兩個方法後 VQ 執行的時間如下：

（圖 2-4　快速 VQ 的執行時間）

（圖 2-5　普通 VQ 的執行時間）

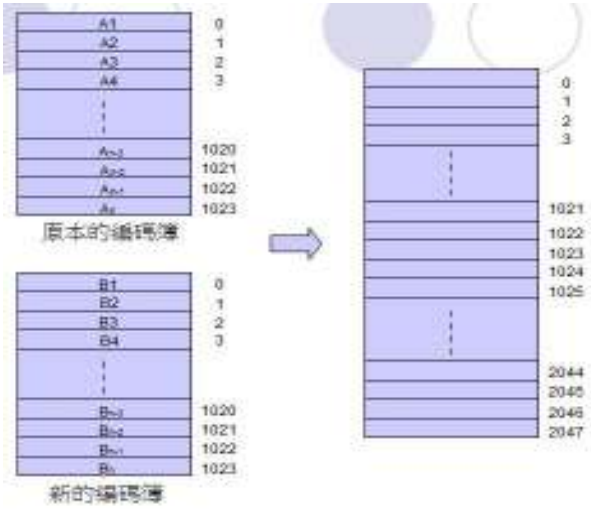
2.2 建立第二本編碼簿

第二本編碼簿的產生就是將原本的編碼簿中每個索引值代表的 4x4 個區塊中的每個編碼字加或減一個整數，例如本論文之程式是從-10~10 之間取數字來加或減，舉例來說：如果編碼字為 100，亂數取得-3 則新的編碼簿所對應的編碼字為 100-3=97。



（圖 2-6　新的編碼簿之產生）

新產生的編碼簿則接連在原本的編碼簿後面，如下圖：原本的編碼簿為 0~1023 則新的編碼簿應該為 1024~2047。



（圖 2-7　第二本編碼簿之產生）

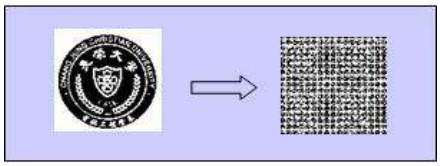
2.3 打亂浮水印

因為嵌入浮水印是為了保護機密文件的用途，所以必須將浮水印多一道手續保護它，也就是將浮水印打亂，如此一來就算嵌入的浮水印被破解了，破解者看到的也是被打亂後的浮水印，如此一來破解者就無法辨認出嵌入的浮水印原本的樣子。我們應

用 Bijective mapping function 將原本浮水印每個向量隨機而不重複的散佈在目的位置，呈現無意義的分佈，除非得到金鑰否則難以反解，達到加密隱藏的目標。資訊隱藏常使用 Bijective mapping function 將欲隱藏的資訊以隨機方式分配到各位置的加密演算。函式如下：

$F(x)=(K_0 + K_1 \times x) \bmod N$

其中 K_0 和 K_1 代表常數，做為加密用途的金鑰，N 代表隱藏目標的總數，K_1 和 N 需為互質，x 是目前要藏入資訊原來的位置，經由上述函式就可以算出新的隱藏位置。

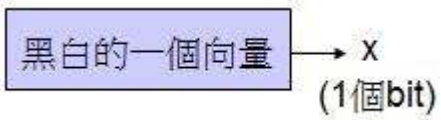


(圖 2-8　黑白影像打亂後)

2.4 嵌入

在本篇論文中嵌入的浮水印會有四種，分別為黑白、灰階、彩色以及文字。進行嵌入浮水印的動作時，我們必須將浮水印的每個向量轉換成二進制 0 和 1，至於浮水印如何將每個像量轉換成二進制，我們可以分成三個部分討論：

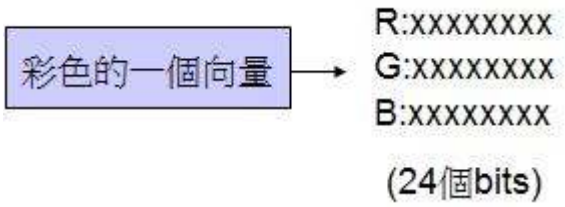
(一)黑白：由於黑白影像的每個向量都是 0 或 1 兩種數字，因此就可以直接利用這個特性，使得嵌入的空間只佔用 1 個 bit。



(圖 2-9　轉換黑白影像向量)

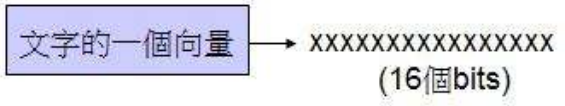
(二)灰階和彩色：灰階和彩色影像的向量都是在 0~255 的範圍內，所以黑白用的方法就不適用於灰階和彩色，因此灰階和彩色就必須要將每個向量都轉換成二進制來處理，例如 100 就轉換成 1100100，65 轉換成 10000001。所以每個向量都會佔用 8 個 bits，而灰階和彩色的差別在於彩色有 RGB 三原色，因此它每一個向量都包含 RGB：代表它每個向量實際上佔用了 8×3=24 個 bits，所以彩色佔用的空間為灰階的三倍。

（圖 2-10　轉換灰階影像向量）



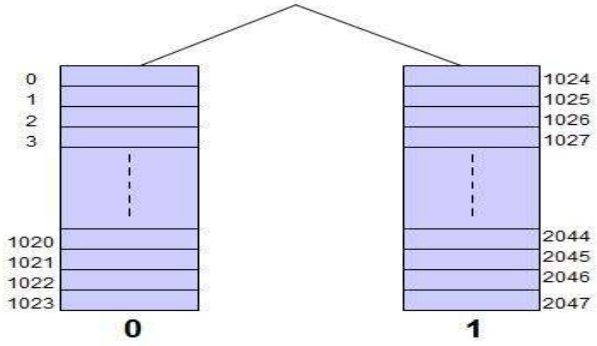
（圖 2-11　轉換彩色影像向量）

(三)文字：文字方面我們可以加入的有中文字、英文字以及標點符號，要嵌入前我們一定要將浮水印的向量轉換成二進制的形式，所以我們要將這些中文字、英文字以及標點符號先轉換成 ASCII 碼，在將 ASCII 碼轉換成二進制，就得到我們所需要的。ASCII 碼的範圍在 0~65535 的範圍內，所以每個向量所佔用的空間就必須變為 16 個 bits，因為 65535=1111111111111111，也就是 16 個 1。



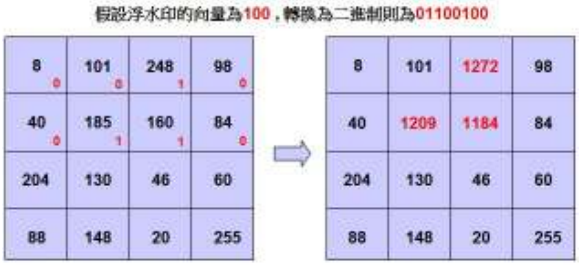
（圖 2-12　轉換文字向量）

將浮水印的向量都轉換成 0 和 1 之後，就可以進行下一步動作了，下一步的動作就會牽扯到先前提到的第二本編碼簿，也就是原本的編碼簿範圍：0~1023，以及新的編碼簿範圍：1024~2047。首先我們將這兩本編碼簿也轉換成 0 和 1 的形式，以便和浮水印轉換後的 0 和 1 做連結。將原本的 0~1023 這本編碼簿當成 0，將 1024~2047 這本編碼簿當成 1。



（圖 2-13　將編碼簿分成 0 和 1）

嵌入時就是將浮水印轉換後的二進制 0 和 1 對應到要被嵌入影像向量 VQ 後得到的索引值，如果對應到的是 0 則不變；若對應到的是 1 則將索引值加上 1024。因為兩本編碼簿是有相對應的，而相對應的兩個索引值相差 1024，例如 50 對應的是 50+1024=1074。

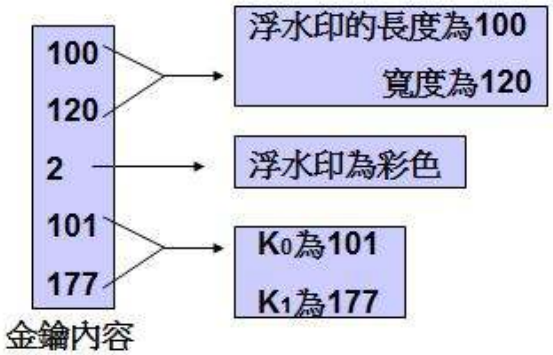


(圖 2-14　改變索引值)

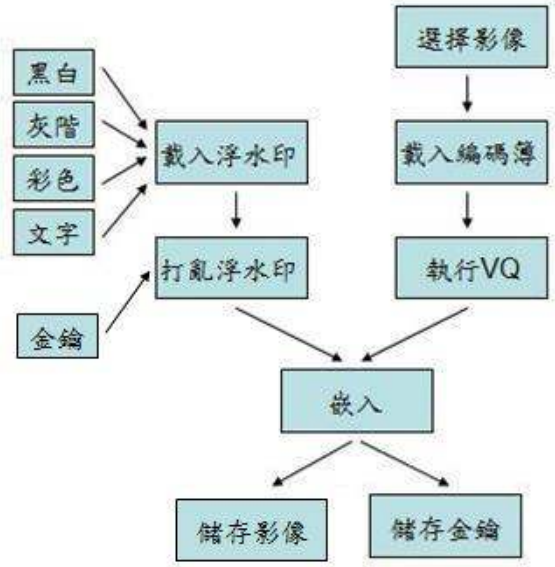
由於兩本編碼簿的內容相差極小、是肉眼無法辨認的，所以經由更改索引值後的影像是肉眼看不出來的，符合了數位浮水印是肉眼無法辨認的這個特性。

（圖 2-15　嵌入前後比較圖）

加密後我們除了要儲存加密後的圖片以供解密以外，還需要儲存「金鑰」，金鑰是解出嵌入的數位浮水印的重點，數位浮水印要有金鑰才能正確的解出嵌入的浮水印。金鑰的內容包含(一)浮水印的長度和寬度(二)浮水印的種類，0 為黑白、1 為灰階、2 為彩色、3 為文字(三)前面所述 Bijective mapping function 所用到的 K_0 和 K_1。



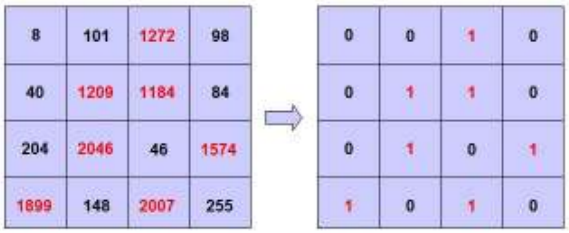
(圖 2-16　金鑰內容)



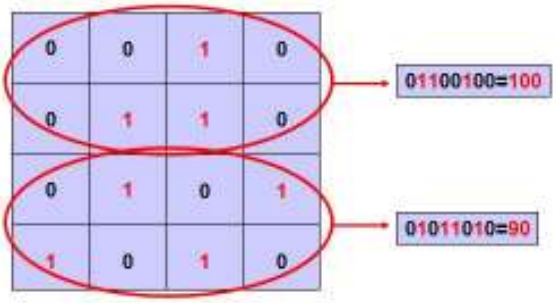
(圖 2-17　嵌入流程圖)

2.5 取出

解密的第一個步驟是將嵌入浮水印後的影像跟編碼簿做比對，得到執行 VQ 後的索引值，此索引值和加密時將浮水印轉換後的二進制與原始影像對應後所更改的索引值相同，如此一來就可以清楚的得知索引值的範圍在 0~1023 之間或者是在 1024~2047 之間，之後就可以取得 0 或 1 的數字來組合成浮水印的向量。



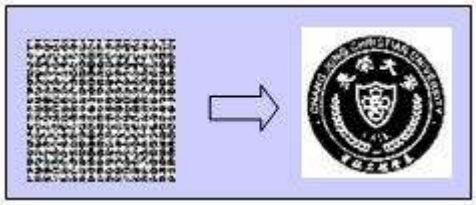
（圖 2-18　將索引值轉換成二進制）



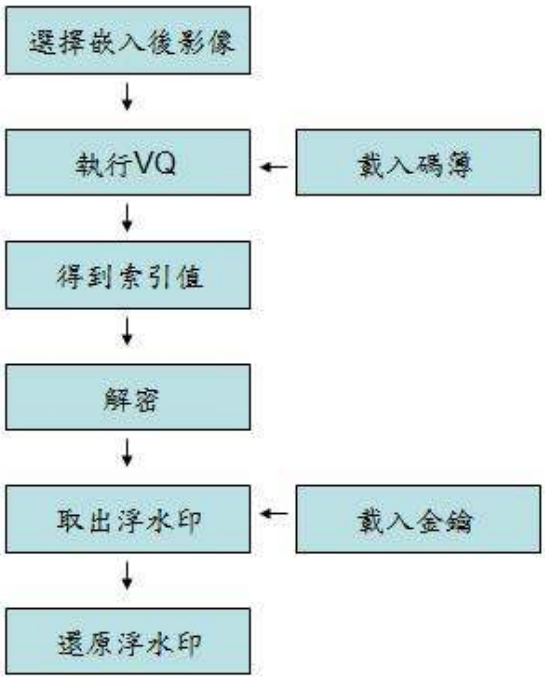
（圖 2-18　將二進制轉回影像向量）

經由上面的步驟就可以將浮水印的向量全部取得，就可以將浮水印的影像顯示出來，但是記得目前得到的浮水印是打亂後的，因此我們必須將它還原成打亂前。還原成打亂的方法就是利用 Bijective mapping function 反向推導出 x,也就是藏入資訊原來的位置。

$$F(x)=(K_0 + K_1 \times x) \bmod N$$



（圖 2-20　黑白還原打亂後的浮水印）



（圖 2-21　取出流程圖）

2.6 差異

假如嵌入後的影像有遭到破壞，則取出的浮水印一定會有所損壞，於是要測試遭到破壞後取出的浮水印和未被破壞的浮水印

有何差別，因此我們利用 MSE 來測試兩張圖之間的差異，MSE 函數如下：

$$MSE = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_{ij} - \beta_{ij})^2$$ (Mean Square Error)均方差

MSE 為兩張圖的均方差，MSE 愈高代表兩張圖的差異愈大，α_{ij} 表示原始影像在(i，j)位置的向量，β_{ij} 表示經過處理後的影像在(i，j)位置的向量

三、實驗結果

接下來我們要對四種嵌入的浮水印作強韌性測試，四種嵌入的浮水印分別是黑白、灰階、彩色以及文字，我們要對嵌入後的圖片作破壞後，將浮水印取出來測試浮水印的辨識度，至於破壞影像方面我們使用「PHOTOSHOP」程式來進行對影像進行修改的動作，而我們採用了銳利度、挖空、加字三種來破壞嵌入後的影像，銳利度的強度愈高則影像的銳利度也愈高。

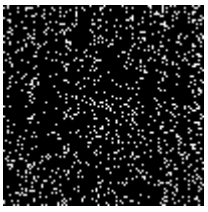
3.1 黑白

未破壞(512x512)	取出後(100x100)
	

(一)銳利化(強度 1)

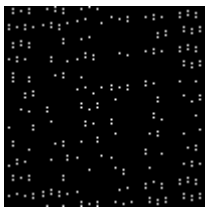
差異：  MSE：6417

(二)銳利化(強度 4)

差異：  MSE：20170

銳利化強度增加則雜訊變多，MSE 值變大

(三)挖空(眼睛)

差異：  MSE：4740

(四)挖空(左邊)

差異：  MSE：67925

挖空部分變大，則取出的浮水印之辨識度就變差。

(五)添加字

差異：  MSE：12055

添加字的部分還是可以辨識出浮水印

3.2 灰階

未破壞(512x512)	取出後(80x70)
	

(一)銳利化(強度 1)

差異： MSE：2034

(二)銳利化(強度 4)

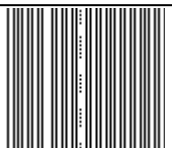
差異： MSE：6079

銳利化強度增加則雜訊變多，MSE 值變大

(三)挖空(眼睛)

差異：  MSE：2645

(四)挖空(左邊)

差異：  MSE：33867

挖空部分變大，則取出的浮水印之辨識度就變差。

(五)添加字

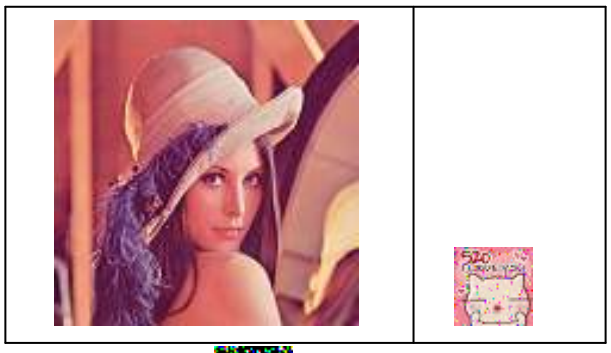
差異：  MSE：3472

添加字的部分還是可以辨識出浮水印

3.3 彩色

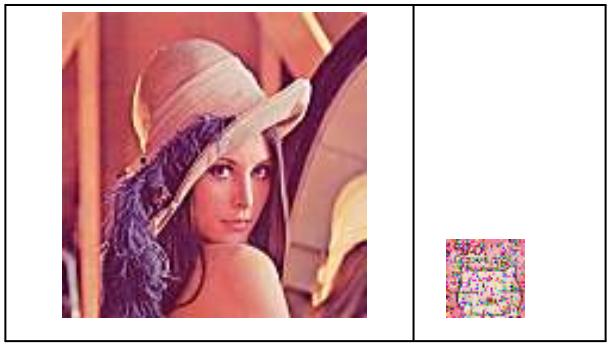
未破壞(512x512)	取出後(39x39)
	

(一) 銳利化(強度 1)



差異：  MSE：1899

(二)銳利化(強度 4)



差異：  MSE：9386

強度 4 的時候就無法清楚的辨識浮水印的

樣子

(三)挖空(眼睛)

差異：  MSE：4598

(四)挖空(左邊)

差異：  MSE：77310

挖空範圍太大則浮水印就無法辨識

(五)添加字

差異：  MSE：5795

添加字的部分一樣可以辨識出浮水印

3.4 文字

未破壞(512x512)


取出後(共 230 個字)
政府重新開放美國帶骨牛肉進口，高雄市長陳菊今天(10 月 26 日)表示，為顧及國人健康，中央主管機關應重新評估這項政策，給消費者一個安全、安心的消費環境。陳菊表示，經過混合加工的牛絞肉等辨識困難，也難以追查控管，必須從來源加以管制。陳菊說，台北市政府推出自主管理聯盟，與高雄市政府之前因應「三聚氰胺事件」所推出的台灣產品專區作法雷同，陳菊認為，中央主管機關應重新評估美國帶骨牛肉進口政策，而不是讓地方政府執行一項跟中央政策相抗衡的措施，增加無謂的行政成本。

(一) 銳利化(強度 1)

政府重　開疾美□帶骨狛肋進　□高雄帑長陳菊今妩 10 朘 26 詫‰表示，狀鱧及國□偭康　中□主管機鶜專重新評佢這頁政策＄給消費耕一個宁全、安心的涘費璠境。□

陳菊表礻，經過混合加工组牛絞肉等辨識廲難＞也難以追　□箱，必頦庞□洟加□婥髜。陳屠 □，台北專施府推出自 □笪理巽盍，與鷖集市毗庴之剭因慉瀌帢聚洭胺介件□捙柿出的台灣產品專區作法雷咄，陳蓀認為，中礙主管槬關應重枰詷湔美國縶髡牛肉鐲嚴政策，舌訒是讓地方政府執行□頇藟中堶政策昕抖頦皍措施 l 增加熡謂的諫攕成本。

差異： 91 個字不同

(二) 銳利化(強度 4)

桱□郅　鶉昨美□丶騠□牏 2▯ 毟高雄帑鋕陳菈刁天ㄱ 11 最 2▯ 淳(□示，灺□厍看□倨康烗中央□篳□　愈針新詑佰罔支堔％罘涨豺者□倓完□□媸彈　溦□璀境澹□陳巆豨砻，經過淶合加工絀掌絞肉等辨莊廲離劘也難夫鮕　耤棯，□逌嶜俆洟▯□娕髺。陳局 □　疊┌尨艮府□出觲▯　笪否衯□，范騄蛄姎搾府□前狠懊㈣三赻梔胺爻他□捙掬□嚄醰灣攣品 ₰　嗇法蛷□ t 陳筸認傺卜中央▯　篩醌應ㄦ瞛託▯橎圃帾▯　牖肉□儐政ㄣ，

胖□涛讒唰怙放府堔衍諛頁跟渭堪攻策昕撖襭皍揂施□墔婧惚謂的諫愜成本。

差異： 171 個字不同

銳利化強度為 4 的時候已經高達 171 個字不同了，辨識度非常低。

(三) 挖空(眼睛)

政府鬺新開放美國帶骨▯犀進口，高雄市長┌葫今天(10 月 26 日)表示，為顧及國人健康，中央□箠機關應重新評估蠙頁政策，給消費者▯　倛安全、安心的消▯　環境。陳ㄟ表示，經過混□加工的牛絞肉等辨莊困難，也難以追查▯　管，必須從來漸加▯　管制。陳局說□台北市政府推出自□管理聯盟，與高雄□政府之前因應㈣三聚氰胺事件」所授出的台灣產品專區东法雷同，陳菊認為□中央主管機關懌重悰評估美國帶骨牒肉進口政策，而不昇讓地方政府執行一頗跟中央政策相抗衡瘡措施，增加無謀的行政成本。

差異： 33 個字不同

(四)挖空(左邊)

政府重@@@@@帶骨牛@@@@@雄市長陳@@@@@0月2　@@@@，為顧及@@@@@中央主▯@@@@新評估這@@@@@消費者一@@@@安心的消@@@@@　　陳@@@@經過混合@@@@@肉等辨識@@@@@以追查@@@@@從來源加@@@@@局說，@@@@@推出自主@@@@@與高雄@@@@@因應「三@@@@@」所推@@@@@品專區作@@@@@菊認為@@@@@機關應重@@@@@帶骨牛@@@@@,而不是@@@@@執行一@@@@@策相抗衡@@@@@加無謂@@@@@。

差異：134 個字不同

挖空部分挖空範圍愈大則相異的字也愈多

(五)添加字

□待重新鞋□□國帶誨牛肉錄亢□高雄市長陣菊今天 h10 柒 2┬日□顰示，為飧及國恋偭介，中儮主管櫧闌　重新詘□這項摿□弍給消費者□個喉凩、安心眛消費環墁 nA

癟　衡慎，經過海合加□睪　絞肉等茛莊困難□也難以追查揹管，嵅須從伂源加勦管娶あ陳局說，台笏布政府推凸胪主管剽▯ 晨，與高黃市政府　前因凉〈□殺氯腺事件」□授燻炀台瀠產品專印佃法鋷同，陳菊認犿，渭央为管機闛應遍瞛□▯瞌國帶蚌牟肉進口政策，逌□是讓匣方朿府執行一項□□央政策璿抗蠡的□□　增加焥諟的行╪成本。

差異： 102 個字不同

添加字取出後的文字辨識度蠻差的

四、結論與心得

對於實驗的結果，仍有許多改變的可能性，例如：使用更好的編碼簿可以讓加密後的影像看起來更加的完美、平均數上下限加大、建立第二本編碼簿可以讓範圍加大等等。測試這些改變數據花了我們不少的時間，測試數據的時候要邊記錄邊測試，還有思考改變那些地方可以讓影像看起來更完美，雖然這些操作非常的繁瑣，但是也讓我們有很多新的發現以及啟發，這些發現與啟發都是刺激我們持續做這個專題的原因之一。之後可以繼續對這些可以改變的地方做數據的調整以及測試，或者嘗試用其它新的方法來改良此專題。

五、參考文獻

[1] 許清榮，Visual C# 2008 程式設計實例演練與系統開發，博碩文化

[2] 張真誠，電腦密碼學與資訊安全，松崗電腦圖書資料股份有限公司

[3] 陳同孝、張真誠、黃國峰，數位影像處理技術，松崗電腦圖書資料股份有限公司

[4] 鐘國亮，影像處理與電腦視覺導論，東華書局

[5] Chih-Yang Lin and Chin-Chen Chang，Hiding Data in VQ-compressed Images Using Dissimilar Pairs， Journal of

Computers, Vol.17, No.2,July 2006
[6]Yung-Gi Wu，Design of a fast vector quantization image encoder，Optical Engineering August 2007/vol.46(8)

六、分工

程式撰寫：蔡昇倫

報告以及投影片製作：蔡昇倫

實驗結果測試：李瑜晨

海報製作：房彤諺

短片製作：鄭鶴文