

# 無損式浮水印在影像篡改偵測與重建之應用

## Lossless Watermarking for Image Tamper Detection and Reconstruction

吳昭明

國立虎尾科技大學

航空與電子科技研究所

[cmwu@nfu.edu.tw](mailto:cmwu@nfu.edu.tw)

吳正綱

國立中興大學

電機工程系通訊所

[wuchengom@gmail.com](mailto:wuchengom@gmail.com)

陳後守

國立中興大學

電機工程系通訊所

[houshou@dragon.nchu.edu.tw](mailto:houshou@dragon.nchu.edu.tw)

### 1、摘要

本論文提出一個基於無損式浮水印的影像篡改偵測與重建技術，為了能夠偵測篡改以及重建影像，影像的重建資訊將使用區塊化差值擴張無損式浮水印來自我嵌入，影像重建資訊包含修復碼、同位檢查碼和溢位/欠位還原資訊。同位檢查碼可用來偵測影像遭受篡改的區塊，當偵測出影像區塊遭受到篡改時，可有效率的利用隱藏在其對應區塊的修復碼修復遭受竄改區塊，若隱藏在其對應區塊中的修復碼也遭受破壞，則使用影像修補技術提高重建影像的品質。此外，由於重建資訊是使用區塊化的差值擴張無損式浮水印方式嵌入，因此未遭受篡改的影像區塊可以無失真的還原原始區塊的內容。也就是說，當嵌入影像未受到任何篡改時，嵌入影像重建資訊所造成的失真將可被完全移除。實驗結果顯示，此一架構除了能夠有效率的偵測篡改區塊和重建外，重建影像的品質還能優於嵌入影像的品質。

**關鍵字：**無損式浮水印、差值擴張、篡改偵測、影像修復、影像修補。

### 1、Abstract

A new image tamper detection and

reconstruction scheme based on lossless watermarking is proposed. To achieve tamper detection and image reconstruction, the reconstruction information of an image is self-embedded by block-wise difference expansion lossless watermarking algorithm. The reconstruction information consists of recovery code, parity-check code, and reverse side information. The parity-check code is used for tamper detection. Image blocks detected as tampered can be recovered effectively from the corresponding recovery code stored in another block. If the corresponding recovery code is also tampered, image inpainting technique is applied to improve the quality of reconstructed image. In addition, since the reconstructed information is embedded by lossless watermarking, all of the non-tampered image blocks can be restored to the original . , i.e., the distortion caused by embedding of the reconstruction information can be removed completely. Experimental results demonstrate that not only the tampered image blocks can be effectively detected and reconstructed but also the quality of reconstructed image will be better than that of watermarked image.

Keywords: lossless watermark, difference expansion, tamper detection, image recovery, image inpainting

## 2、簡介

隨著多媒體以及網路的發展，數位化儼然成為不可抵擋的趨勢，底片、唱片、磁帶逐漸走入歷史，取而代之的是記憶卡、光碟、數位視訊影碟。藉由隨身存取裝置和光碟機等裝置，我們可以將照片(picture)、音訊(audio)、視訊(video)等數位內容儲存至電腦硬碟或網路硬碟，不但能隨時滿足視聽上的要求，甚至還能不受限地任意傳播、散佈、編輯、複製、篡改這些數位化內容，引發盜版問題以及版權爭議等社會問題。因此，對於數位化版權內容的保護及其完整性的認證是一件刻不容緩需要解決的問題。

Cox[8]針對此問題提出了兩個數位浮水印的應用：一、版權處理(copyright management)。二、完整性認證(integrity authentication)。版權處理包含了版權保護(copyright protection)和數位指紋(digital fingerprinting)，分別用作智慧財產權的保護以及追蹤非法的複製盜用者。完整性認證則主要用來認證數位媒體的真確性，也就是檢測端能夠判斷媒體是否遭受任何的修改。

近年來許多學者廣泛地討論篡改認證技術(tamper proofing technique)，其中包含影像篡改偵測(image tamper detection)[7-8,13,15-16,21]以及偵測篡改與修復(tamper detection and recovery)[3-4,12,18-20]技術。影像篡改偵測技術主要是將認證資料當作浮水印嵌入影像中，其所嵌入的認證資料能夠被用來檢測該影像是否遭受篡改並且定位出遭受篡改的位置。而偵測篡改與修復架構則是進一步取出影像的特徵資訊(feature information)作為浮水印並嵌入至原始影像內，當嵌入浮水印的影像遭受竄改時，能夠藉由取出浮水印去修復(recovery)受損區塊。

一般的浮水印架構在浮水印嵌入時影像會

遭受到輕微的修改，通常肉眼並不會察覺，但在某些特殊應用，如醫學、法律鑑定等，即使此種輕微的失真也不允許，因此無損式浮水印為另一個廣受探討的議題，此種架構能夠由嵌入浮水印後的影像取出浮水印並還原原始影像。Feng 等人[5]將無損式浮水印分為三類：包含基於資料壓縮之無損式浮水印[9,14-15]、基於擴張差值之無損式浮水印[1-2,10-11]以及基於直方圖平移之無損式浮水印[17]。

由於無損式浮水印具有取出浮水印並還原原始影像的特性，因此本質上即可用作篡改認證的應用。早期的無損式浮水印文獻[7-8,13]往往也提到其在篡改認證上的功用。其作法是以無損式浮水印的演算法將認證資料當作浮水印嵌入原始影像；認證時，先由嵌入浮水印影像取出認證資料，再依據無損式浮水印演算法還原原始影像並重新計算認證資料，若取出的認證資料和重新計算的認證資料一致，則可認證無損式浮水印所還原之原始影像的完整性。

此一無損式浮水印認證方式必須先作原始影像還原方可達成影像認證的目的，Celik 等人[15]提出一種新的無損式浮水印認證架構，此一架構對於認證通過的影像同樣可作無失真的還原，但不同於傳統的無損式浮水印認證方式，必須先做原始影像還原才可驗證，新架構是先通過完整性驗證後才需要還原原始影像。在認證失敗或是只需要驗證而不需要還原原始影像的情況下，此一架構可有效減少計算量。

Lin 等人[20]提出使用基於擴張差值之無損式浮水印[10]來嵌入影像的特徵資訊。其嵌入步驟依序可分成兩個階段：一、有失真嵌入同位檢查碼：首先將原影像分割成不重疊的 $2 \times 2$ 區塊並將所有像素的最低位元歸零，接著計算出二進制的區塊整數平均值，並以互斥或求得 $p_1$ ，最後將同位檢查碼 $p_1$ 和 $\bar{p}_1$ ( $\bar{p}_1$ 為的補數)嵌入在區塊的最低位元即可得嵌入影像 $I_{em}$ 。二、無損式嵌入修復資訊：將 $I_{em}$ 使用3階離散小波轉換(DWT)，取

出LL2和LL3每一個係數的前8個重要位元作為其修復資訊，接著使用 Tian[10]差值擴張的方式以水平配對嵌入後再以垂直配對嵌入剩下的修復資訊。當同位檢查碼偵測出篡改時，即可利用修復資訊修復篡改，若影像沒有遭受任何篡改時，即可還原到嵌入影像 $I_{em}$ 。

本論文以 Wu 等人[2]所提出區塊化差值擴張無損式浮水印方法為基礎並經適當修改來嵌入浮水印，浮水印的內容包含區塊修復碼 $\mathcal{R}$ 、同位檢查碼 $\mathcal{P}$ 、溢位/欠位還原資訊 $\mathcal{A}$ 。由於使用無損式浮水印方法嵌入在原始影像上，所以當嵌入浮水印的影像未受到任何篡改時，可無失真的完整還原原始的影像。

### 3、新的演算法

本論文所提出的演算法可分為三個部份：浮水印嵌入、偵測篡改區域以及影像重建。

#### 3.1、嵌入浮水印

本論文以 Wu 等人[2]所提出區塊化差值擴張無損式浮水印方法為基礎並經適當修改來嵌入浮水印，浮水印的內容包含區塊修復碼 $\mathcal{R}$ 、同位檢查碼 $\mathcal{P}$ 、溢位/欠位還原資訊 $\mathcal{A}$ 。

##### 3.1.1、區塊化差值擴張無損式浮水印

首先將原影像分割成 $3 \times 3$ 不重疊的區塊 $B$ ，每一個區塊的中間點稱為參考點 $p_r$ ，其他的像素以 $p_i, i = 1, 2, \dots, 8$ 表示。每一個區塊將有八組像素配對 $(p_i, p_r), i = 1, 2, \dots, 8$ ，區塊化差值擴張演算法將保留參考點 $p_r$ ，僅針對 $p_i$ 嵌入浮水印。區塊內每一組配對像素的差值可表示成：

$$h_i = p_i - \left\lfloor \frac{p_r}{2} \right\rfloor \times 2, \quad (1)$$

$\lfloor x \rfloor$ 表示不大於 $x$ 的最大整數。假設要嵌入的隱藏資訊 $b \in \{0, 1\}$ ，嵌入隱藏資訊後的像素配對記為

$(p'_i, p_r)$ ，則：

$$p'_i = \left\lfloor \frac{p_r}{2} \right\rfloor \times 2 + 2h_i + b \quad (2)$$

在取出藏匿於區塊的資料時，區塊內各配對像素新的差值 $h'_i$ 定義如下：

$$\begin{aligned} h'_i &= p'_i - \left\lfloor \frac{p_r}{2} \right\rfloor \times 2 \\ &= 2h_i + b \end{aligned} \quad (3)$$

則隱藏的資訊位元 $b$ 由下式決定：

$$b = \begin{cases} 1 & \text{當 } h'_i \text{ 為奇數} \\ 0 & \text{當 } h'_i \text{ 為偶數} \end{cases} \quad (4)$$

注意， $\lfloor p_r/2 \rfloor \times 2$ 是一個偶數，從公式(3)可知，若 $h'_i$ 為偶數則 $p'_i$ 是偶數；反之，若 $h'_i$ 為奇數則 $p'_i$ 是奇數，所以取出隱藏的資訊位元 $b$ 可簡單的由 $p'_i$ 取出，即：

$$b = \begin{cases} 1 & \text{當 } p'_i \text{ 為奇數} \\ 0 & \text{當 } p'_i \text{ 為偶數} \end{cases} \quad (5)$$

要還原區塊內的元素值 $p_i$ ，必須先取得區塊內各配對像素原來的差值 $h_i$ ：

$$h_i = \frac{h'_i - b}{2} \quad (6)$$

區塊內原來的像素 $p_i$ 可用下式進行無失真的還原：

$$p_i = \left\lfloor \frac{p_r}{2} \right\rfloor \times 2 + h_i \quad (7)$$

由於差值擴張後的像素可能會發生溢位(像素值>255)或者欠位(像素值<0)的情況，所以原始影像在嵌入前必須先針對溢位和欠位的像素值做適當的調整。使得每一個區塊都能夠嵌入浮水印。

### 3.1.2、產生浮水印

嵌入區塊的浮水印包含三個部分：區塊修復碼 $\mathcal{R}$ 、同位檢查碼 $\mathcal{P}$ 、溢位/欠位還原資訊 $\mathcal{A}$ 。每區塊修復碼 $\mathcal{R}$ 的長度為5個位元，首先將區塊中9個像素值取整數平均值，將此一平均值表示為二進位數 $(m_7, m_6, m_5, m_4, m_3, m_2, m_1, m_0)_2$ ，我們取其前5個重要位元作為此區塊的修復碼，即：

$$\mathcal{R} = (m_7, m_6, m_5, m_4, m_3)。 \quad (8)$$

當某一區塊遭受到篡改時，可先將此區塊的修復碼 $\mathcal{R}$ 後面串接三個0位元，而得到區塊的近似平均值 $m$ ，即

$$\begin{aligned} m &= \mathcal{R} \parallel (0,0,0) \\ &= (m_7, m_6, m_5, m_4, m_3, 0,0,0)_2 \end{aligned} \quad (9)$$

再用此一近似平均值 $m$ 替代區塊中的9個像素值，此即達到區塊修復的目的。同位檢查碼 $\mathcal{P}$ 主要用於偵測區塊是否遭受篡改，而同位檢查碼 $\mathcal{P}$ 是由兩個互補的位元 $p$ 和 $v$ 串接而成，即：

$$\mathcal{P} = (p, v) \quad (10)$$

其中

$$p = m_7 \oplus m_6 \oplus m_5 \oplus m_4 \oplus m_3。 \quad (11)$$

區塊修復碼 $\mathcal{R}$ 和同位檢查碼 $\mathcal{P}$ 合稱為區塊特徵碼 $\mathcal{F}$ 。為了讓原本會溢位或者欠位的像素也能夠還

原成原來的像素值，必須額外嵌入一些附屬資訊，此為溢位/欠位還原資訊 $\mathcal{A}$ 。

### 3.1.3、浮水印嵌入位置

圖二所示為一個影像區塊中的9個像素，區塊修復碼 $\mathcal{R}$ 和其同位檢查碼 $\mathcal{P}$ 將分別隱藏在相對應區塊的 $p_1 \sim p_5$ 以及 $p_6 \sim p_7$ 。而保留下來的 $p_8$ 則嵌入溢位/欠位還原資訊 $\mathcal{A}$ 。相對應區塊的對應方式是將影像分成四個象限，以對角方向之象限作為互相配對的基準，並經由隨機亂數攪亂其區塊的順序。

$p_1$	$p_4$	$p_6$
$p_2$	$p_7$	$p_7$
$p_3$	$p_5$	$p_8$

圖二、影像區塊像素

## 3.2、偵測篡改區塊

每一個區塊取出浮水印後，利用區塊特徵碼 $\mathcal{F}$ 來偵測區塊是否遭受篡改。偵測篡改程序將分為兩個階段進行：檢查特徵碼 $\mathcal{F}$ 和擴大篡改區域。

### 3.2.1、階段一：偵測篡改區塊

座標為 $(x, y)$ 的區塊 $B_{x,y}$ 是否有被竄改，首先由區塊的像素 $p_1 \sim p_7$ 取出隱藏於此區塊的修復碼 $\mathcal{R} = (m_7, m_6, m_5, m_4, m_3)$ 及同位檢查碼 $\mathcal{P} = (p, v)$ ，接著用 $\mathcal{R}$ 重新計算同位檢查位元 $p_e$ ，

$$p_e = m_7 \oplus m_6 \oplus m_5 \oplus m_4 \oplus m_3 \quad (12)$$

如果 $p_e = p$ ，且 $p_e \neq v$ ，則此區塊 $B_{x,y}$ 未遭受竄改；反之，則區塊 $B_{x,y}$ 有遭受竄改。我們用 $t_{x,y}$ 來表示此一階段區塊是否遭受竄改的偵測結果：

$$t_{x,y} = \begin{cases} 0 & \text{區塊 } B_{x,y} \text{ 未遭受竄改} \\ 1 & \text{區塊 } B_{x,y} \text{ 有遭受竄改} \end{cases} \quad (13)$$

### 3.2.2、階段二：擴大篡改區塊

若一影像區塊有遭受竄改，則周圍區塊亦遭受竄改的機率將大為提高。因此竄改區塊擴大的程序將可有效降低偵測失誤(missing)的機會。針對影像區塊 $B_{x,y}$ ，我們以其為中心計算周圍區塊在第一階段偵測時判定其有受竄改的區塊數 $n_{x,y}$ 表示如下：

$$n_{x,y} = \sum_{x',y' \in W_{x,y}} t(x',y') \quad (14)$$

其中 $W_{x,y}$ 是一個 $3 \times 3$ 的移動視窗且中心坐落於區塊 $B_{x,y}$ 上。若 $n_{x,y} > 0$ 則認定此區塊 $B_{x,y}$ 也遭受竄改，若 $n_{x,y} = 0$ 則判定此區塊 $B_{x,y}$ 未受竄改。經過兩階段程序後，區塊 $B_{x,y}$ 是否有受竄改將由 $T_{x,y}$ 表示如下：

$$T_{x,y} = \begin{cases} 1 & B_{x,y} \text{ 遭受篡改, 即 } n_{x,y} > 0 \\ 0 & B_{x,y} \text{ 未遭受篡改, 即 } n_{x,y} = 0 \end{cases} \quad (15)$$

## 3.3、影像重建

影像重建依序包含三個階段：1、判定未受竄改區塊作無失真還原。2、使用驗證後的修復碼作有失真的修復受篡改區塊，3、若遭受竄改區塊其相對應之修復碼亦受攻擊，則利用周圍已作無失真還原或有失真修復之區塊修補。

### 3.3.1、階段一：無失真還原區塊

由於每一個區塊均採無失真嵌入浮水印，故針對 $T_{x,y} = 0$ 的區塊可使用公式(3)、公式(6)以及公式(7)還原。

### 3.3.2、階段二：有失真修復區塊

針對 $T_{x,y} = 1$ 且修復碼 $\mathcal{R}$ 未被損壞的區塊可進行有失真的修復。首先將長度為5個位元的修復碼 $\mathcal{R}$ 串接3個0位元，再將此8個位元轉換為十進制 $\mathcal{R}_{\text{dec}}$ ，接著把有遭受竄改的區塊 $B_{x,y}$ 內所有的像素皆以 $\mathcal{R}_{\text{dec}}$ 取代，完成修復後再將 $T_{x,y}$ 設為0。

### 3.3.3、階段三：影像修補

受篡改區塊若其相對應修復碼 $\mathcal{R}$ 也被攻擊時，可利用周圍已無失真還原區塊和已完成有失真修復區塊進行影像修補。針對每一個區塊 $B_{x,y}$ 定義修補係數 $\xi_{x,y}$ 如下：

$$\xi_{x,y} = \begin{cases} 1 & \text{if } T_{x,y} = 1 \text{ 且 } (9 - \sum_{x',y' \in W_{x,y}} T_{x',y'}) > 0 \\ 0 & \text{其它} \end{cases} \quad (16)$$

修補係數 $\xi_{x,y} = 1$ 表示區塊 $B_{x,y}$ 可進行修補；修補係數 $\xi_{x,y} = 0$ 表示區塊無需或者目前無法進行修補。針對所有 $\xi_{x,y} = 1$ 的區塊可求得影像修補值 $C_{x,y}$ ：

$$C_{x,y} = \frac{\sum_{x',y' \in W_{x,y}} (1 - T_{x',y'}) \left[ \frac{1}{9} \sum_{j=1}^3 \sum_{k=1}^3 B_{x',y'}(j,k) \right]}{9 - \sum_{x',y' \in W_{x,y}} T_{x',y'}} \quad (17)$$

此修補值是由周圍影像區塊中 $T_{x,y} = 0$ 的區塊所求得的像素平均值。利用影像修補值 $C_{x,y}$ 替代要修補區塊的所有像素，並將完成修補區塊的 $T_{x,y}$ 設為0。此一修補動作必須重覆遞迴修補，直至所有 $T_{x,y}$ 的內容全為0，此時即完成所有的區塊的重建動作。

## 4、實驗結果與討論

使用四張影像大小為 $510 \times 510$ 的影像進行實驗，包括 Lena、Elaine、Truck 以及 House。首先以增加裁切(crop)影像區塊個數的方式模擬重建影像品質的變化，其模擬結果如圖三所示。由圖三可發現，重建影像的品質會出現一斷層的現象，此一現象是由於溢位/欠位還原資訊 $\mathcal{A}$ 遺失，導致大部分的溢位/欠位區塊無法重建(包括還原、修復和修補)。另外，由圖三還可發現，本論文突破傳統影像修復架構的效能限制，即使嵌入容量僅僅每像素嵌入 $8/9$ 個位元且篡改範圍為25%影像大小，但由於使用無損式浮水印架構使得未受篡改區塊得以無失真還原外，需要修補的區塊也有機會使用部分的無失真像素進行修補，所以重建影像的品質仍遠大於嵌入浮水印影像的品質。

本論文也實驗三種不同屬性的篡改行為，如圖四和圖十一是將別的影像圖樣移植加以覆蓋、圖五至圖八以及圖十三則屬於裁切影像區塊、圖十和圖十二則是複製鄰近影像方式的篡改行為。每一種篡改皆依序展示原影像、嵌入影像、篡改影像、偵測出篡改的區塊、還原未篡改區塊影像、修復篡改區塊影像、尚未修補區塊以及修補影像。從圖九的 F6 和圖十二的 I6 可發現沒有區塊需要做影像修補，這是因為相對應的修復碼是嵌入在對角象限，若被篡改區域不是同時包含對角象限時，相對應的修復碼皆可正確的取出並用來修復篡改區塊。

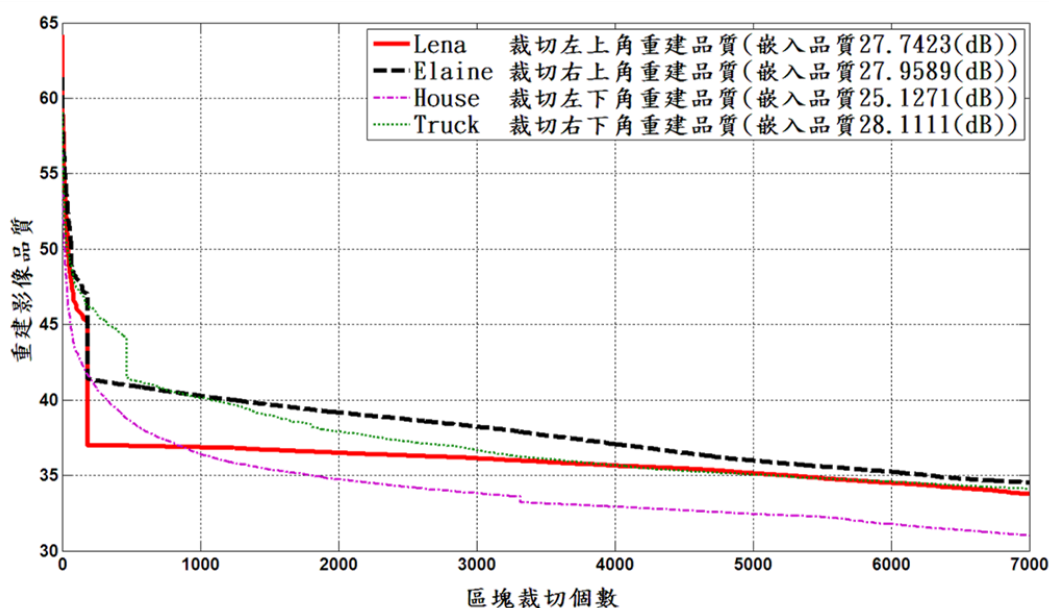
由以上的實驗結果可看出本論文即使嵌入容量小於 1，還是能夠非常有效率的定位出篡改區塊和重建篡改區塊。

## 5、參考文獻

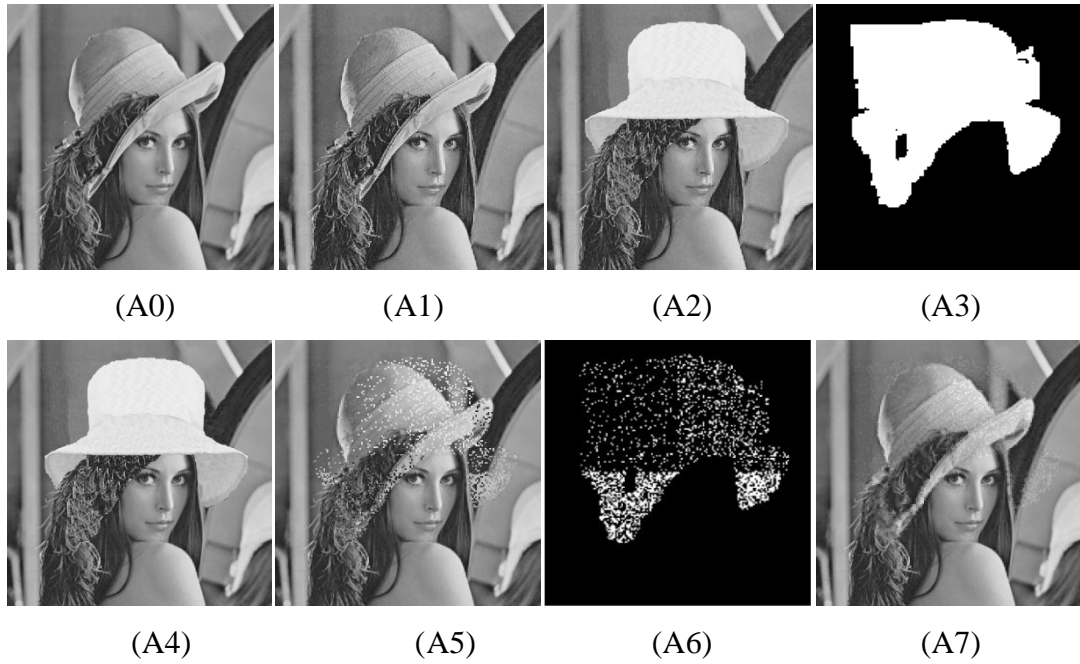
[1] A. M. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform", IEEE Transactions on Image Processing, Vol. 13, No. 8, pp.

- 1147-1156, Aug. 2004.
- [2] C. M. Wu, C. K. Wu and H. S. Chen, "A Simple Reversible Watermarking Scheme Based on Difference Expansion", Proc. ITC-CSCC, pp. 1188-1191, Jul. 2009.
- [3] C. L. Wang, R. H. Hwang, T. S. Chen and H.Y. Lee, "Detecting and Restoring System of Tampered Image Based on Discrete Wavelet Transformation and Block Truncation Coding", Proceedings of the 19th International Conference on Advanced Information Networking and Applications, Vol. 2, pp. 79-82, 2005.
- [4] C. T. Wang, T. S. Chen and S. H. He, "Detecting and Restoring the Tampered Images Based on Iteration-free Fractal Compression", Journal of Systems and Software, Vol. 67, No. 2, pp. 131-140, Aug. 2003.
- [5] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu, "Reversible Watermarking: Current Status and Key No.s", International Journal of Network Security, Vol. 2, No. 3, pp. 161-171, May 2006.
- [6] I. J. Cox, M. L. Miller and J. A. Bloom, "Digital watermarking", Morgan Kaufmann Publishers, USA, 2001.
- [7] J. Fridrich, "Image Watermarking for Tamper Detection", IEEE International Conf. on Image Processing, Vol. 2, pp. 404-408, Chicago, IL, USA, Oct. 1998.
- [8] J. Fridrich, "Methods for Tamper Detection in Digital Images", Proceedings of Multimedia and Security Workshop at ACM Multimedia, pp. 19-23, Orlando, FL, 30-31 Oct. 1999.
- [9] J. Fridrich, M. Goljan and R. Du, "Invertible Authentication," in Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents III, vol. 3971, pp. 197-208, San Jose, California, Jan. 2001.
- [10] J. Tian, "Reversible Data Embedding Using a Difference Expansion", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 8, pp. 890-896, Aug. 2003.
- [11] L. Kamstra and H. J. A. M. Heijmans, "Reversible Data Embedding Into Images

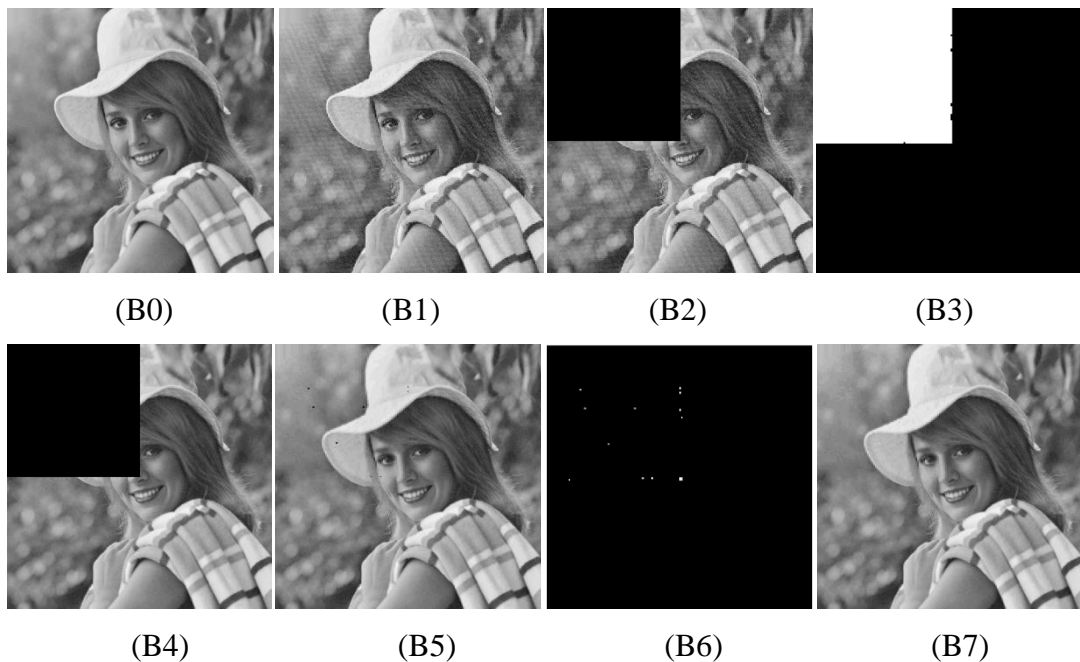
- Using Wavelet Techniques and Sorting”, IEEE Transactions on Image Processing, Vol. 14, No. 12, pp. 2082-2090, Dec. 2005
- [12] T. Y. Lee and S. D. Lin, “Dual Watermark for Image Tamper Detection and Recovery”, Pattern Recognition, Vol. 41, No. 11, pp. 3497-3506, Nov. 2008.
- [13] M. U. Celik, G. Sharma, E. Saber, and A. Murat Tekalp, “Hierarchical Watermarking for Secure Image Authentication with Localization”, IEEE Transaction on Image Processing, Vol. 11, No. 6, pp. 585-595, Jun. 2002.
- [14] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, “Lossless Generalized-LSB Data Embedding”, IEEE Transactions on Image Processing, Vol. 14, No. 2, pp. 253-266, Feb. 2005.
- [15] M. U. Celik, G.Sharma and A. M. Tekalp, “Lossless Watermarking for Image Authentication: A New Framework and An Implementation”, IEEE Transactions on Image Processing, Vol. 15, No. 4, pp. 1042-1049, Apr. 2006.
- [16] R. Ni, Q. Ruan and Y. Zhao, “Pinpoint Authentication Watermarking Based on a Chaotic System”, Forensic Science International, Vol. 179, No. 1, pp. 54-62, Jul. 2008.
- [17] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, “Reversible Data Hiding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 3, pp. 354-362, Mar. 2006.
- [18] P. L. Lin, P. W. Huang, A. W. Peng, “A Fragile Watermarking Scheme for Image Authentication with Localization and Recovery”, IEEE Sixth International Symposium on Multimedia Software Engineering, pp. 146-153, 13-15 Dec. 2004.
- [19] P. L. Lin, C. K. Hsieh, P. W. Huang, “A hierarchical Digital Watermarking Method for Image Tamper Detection and Recovery”, Pattern Recognition, Vol. 38, No. 12, pp. 2519-2529, Dec. 2005.
- [20] S. D. Lin and A. L. Chiang, “Tamper Detection and Recovery Using a Reversible Watermarking Technique”, Journal of Computers, Vol. 20, No. 1, pp. 2-10, Apr. 2009.
- [21] S. Walton, “Image Authentication for a Slippery New Age”, Dr. Dobb's Journal, Vol. 20, pp. 18-26, 1995.



圖三、增加影像區塊裁切個數的方式模擬重建影像品質的變化

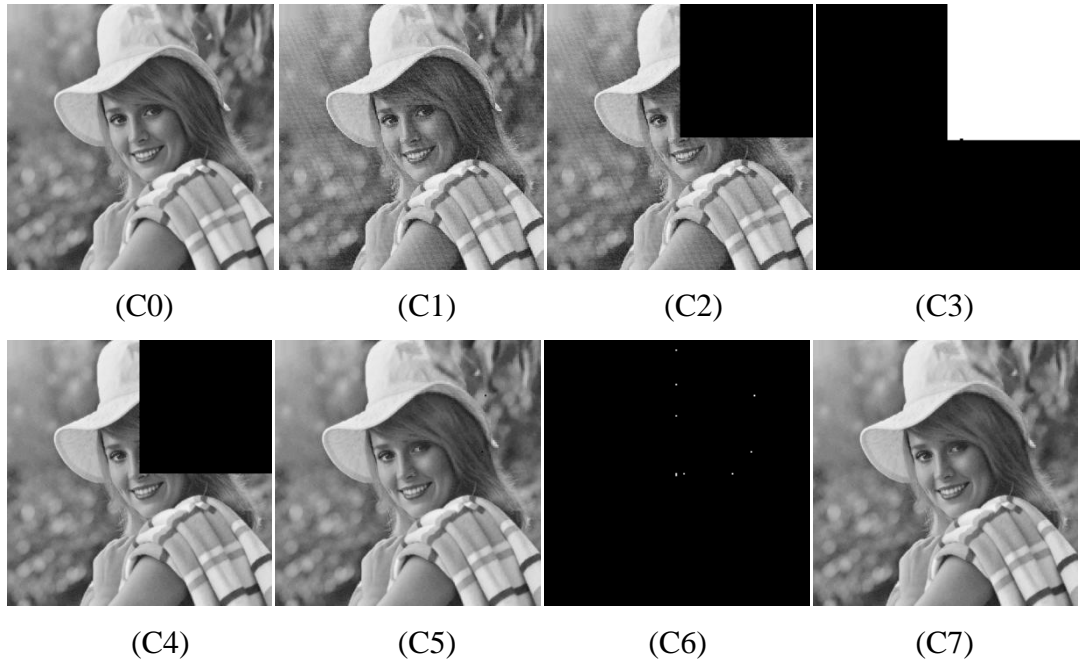


圖四：(A0)原影像 (A1)嵌入影像 PSNR=27.7423 (A2)篡改影像 PSNR=14.4917 (A3)偵測篡改區塊 (A4)無失真還原未篡改區塊 PSNR=14.5891 (A5)修復篡改區塊 PSNR=21.0934 (A6)尚未修補區塊(A7)修補遺失回復資訊之篡改區塊 PSNR=28.5816。

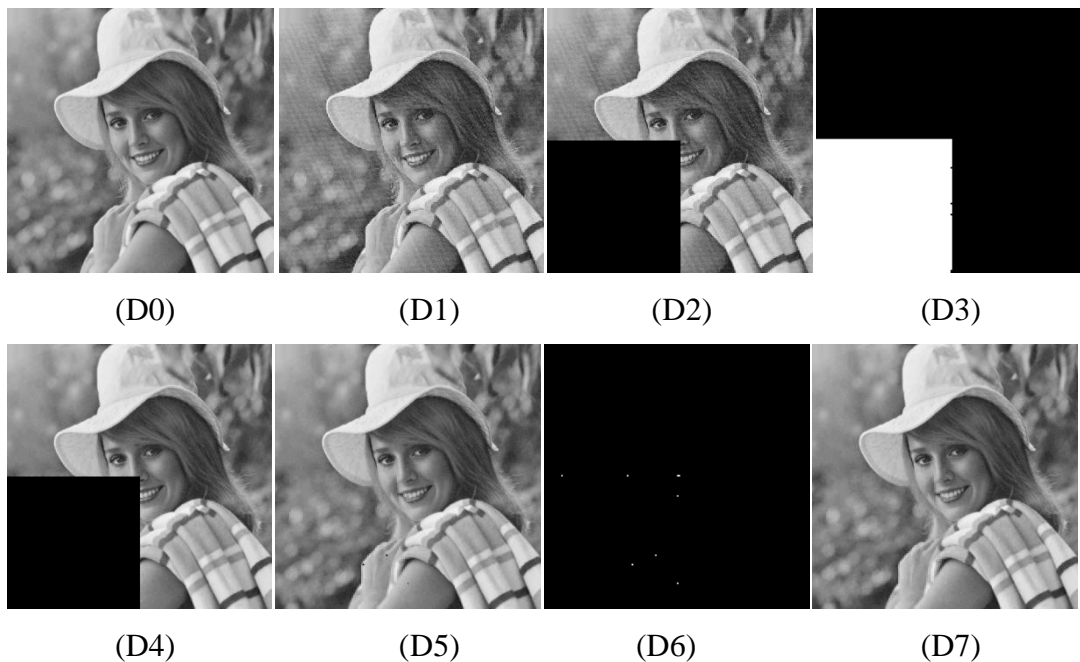


圖五：(B0)原影像 (B1)嵌入影像 PSNR=27.9589 (B2)篡改影像 PSNR=9.7546 (B3)偵測篡改區塊 (B4)無失真還原未篡改區塊 PSNR=9.8041 (B5)修復篡改區塊 PSNR=33.5691 (B6)尚未修補區塊(B7)修補遺失回復資訊之篡改區塊 PSNR=34.3175。

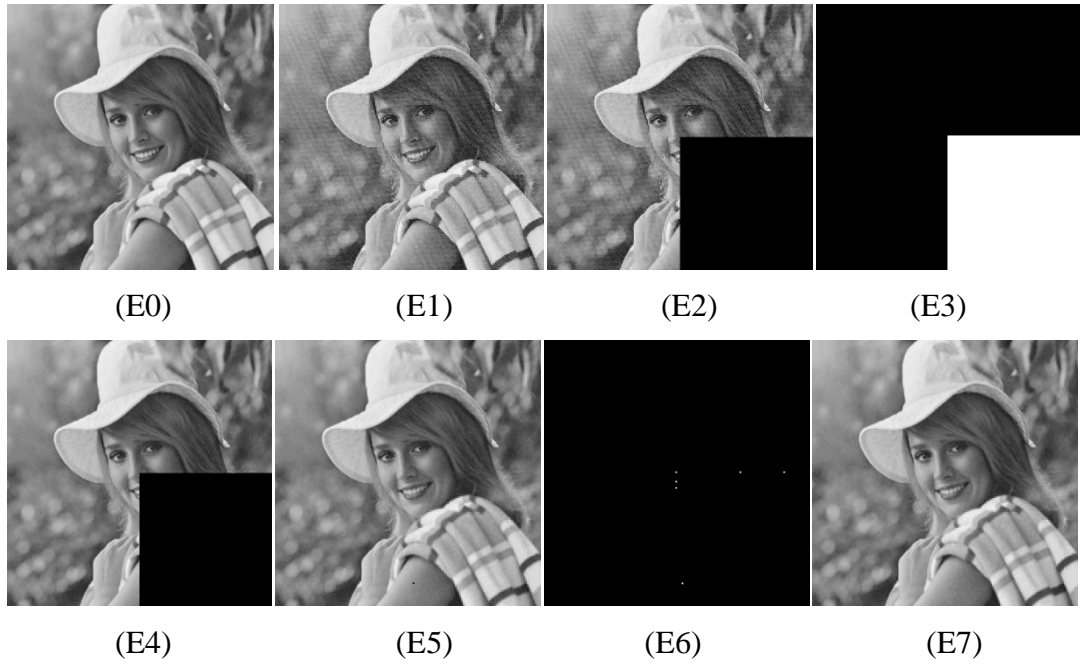




圖六：(C0)原影像 (C1)嵌入影像 PSNR=27.9589 (C2)篡改影像 PSNR=12.7887 (C3)偵測篡改區塊 (C4)無失真還原未篡改區塊 PSNR=12.8797 (C5)修復篡改區塊 PSNR=33.5942(C6)尚未修補區塊(C7)修補遺失回復資訊之篡改區塊 PSNR=33.7699。



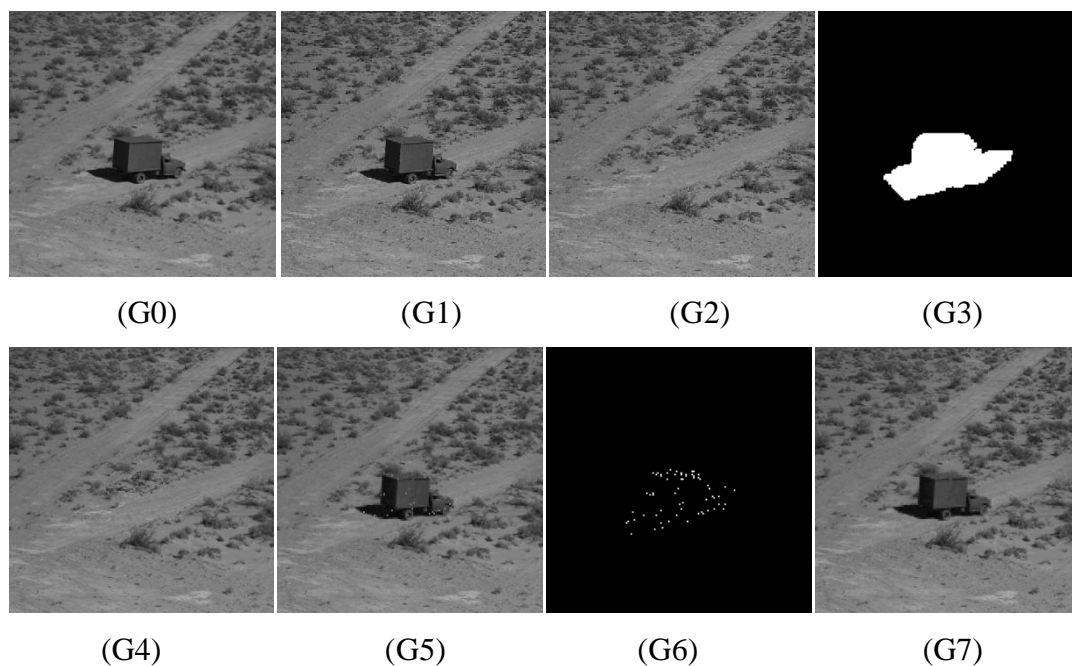
圖七：(D0)原影像 (D1)嵌入影像 PSNR=27.9589 (D2)篡改影像 PSNR=11.2744 (D3)偵測篡改區塊 (D4)無失真還原未篡改區塊 PSNR=11.3436 (D5)修復篡改區塊 PSNR=33.6952 (D6)尚未修補區塊(D7)修補遺失回復資訊之篡改區塊 PSNR=34.2416。



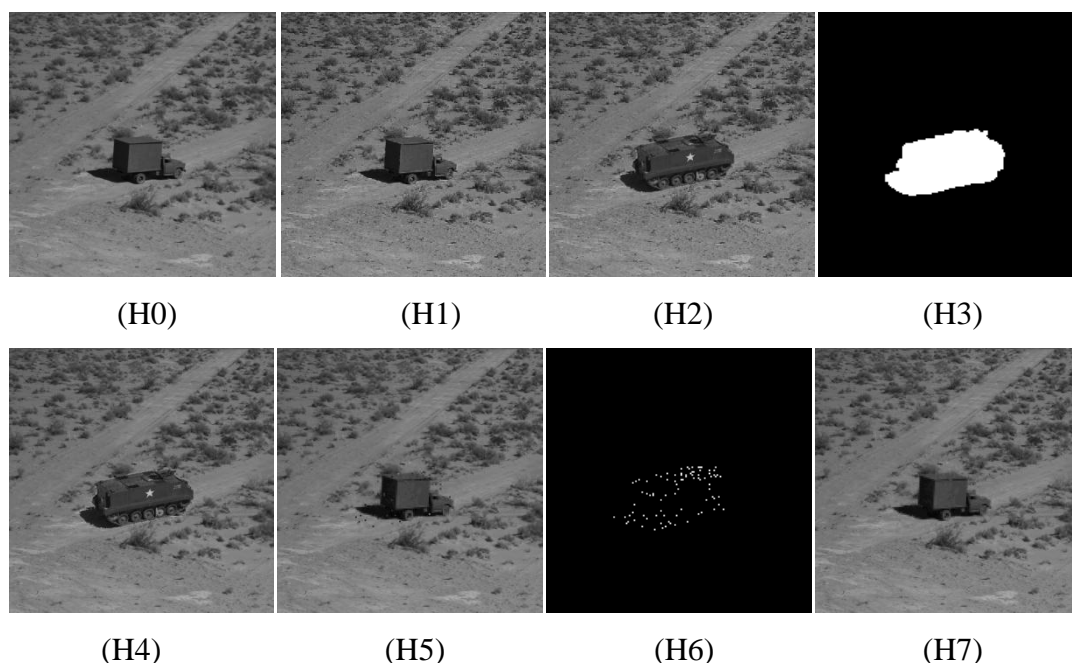
圖八：(E0)原影像 (E1)嵌入影像 PSNR=27.9589 (E2)篡改影像 PSNR=10.3719 (E3)偵測篡改區塊 (E4)無失真還原未篡改區塊 PSNR=10.4232 (E5)修復篡改區塊 PSNR=33.0479 (E6)尚未修補區塊 (E7)修補遺失回復資訊之篡改區塊 PSNR=33.1399。



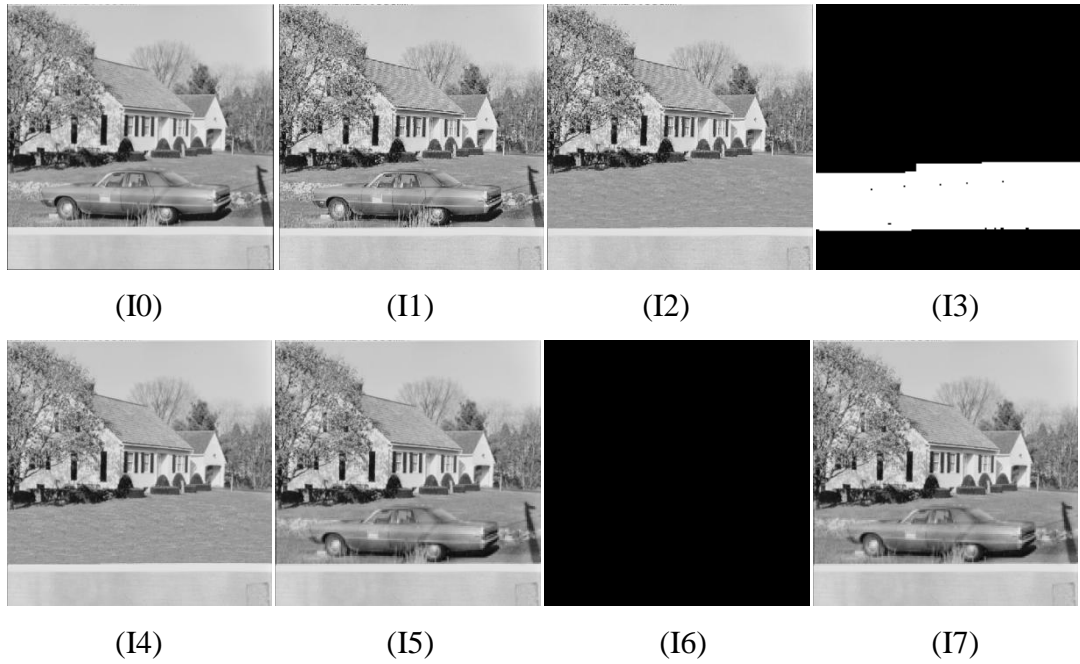
圖九：(F0)原影像 (F1)嵌入影像 PSNR=27.9589 (F2)篡改影像 PSNR=10.3466 (F3)偵測篡改區塊 (F4)無失真還原未篡改區塊 PSNR=10.3991 (F5)修復篡改區塊 PSNR=33.5652 (F6)尚未修補區塊 (F7)修補遺失回復資訊之篡改區塊 PSNR=33.5652。



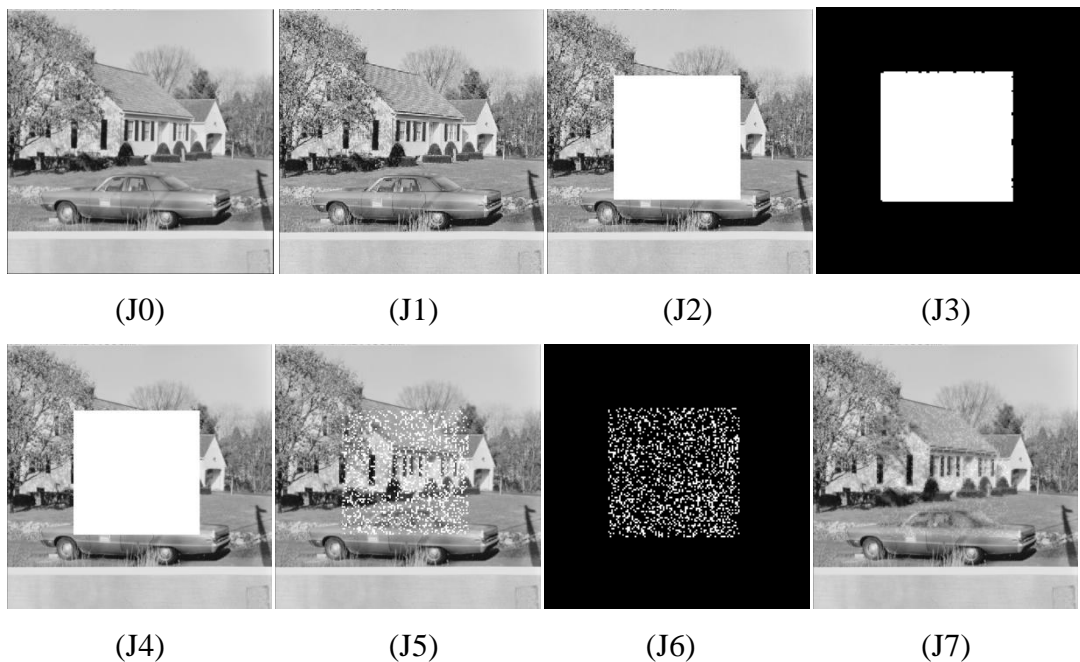
圖十：(G0)原影像 (G1)嵌入影像 PSNR=28.1117 (G2)篡改影像 PSNR=22.5972 (G3)偵測篡改區塊 (G4)無失真還原未篡改區塊 PSNR=23.8630 (G5)修復篡改區塊 PSNR=35.7472 (G6)尚未修補區塊(G7)修補遺失回復資訊之篡改區塊 PSNR=37.8637。



圖十一：(H0)原影像 (H1)嵌入影像 PSNR=28.1117 (H2)篡改影像 PSNR=23.5570 (H3)偵測篡改區塊 (H4)無失真還原未篡改區塊 PSNR=25.1835 (H5)修復篡改區塊 PSNR=35.8093(H6)尚未修補區塊(H7)修補遺失回復資訊之篡改區塊 PSNR=37.5164。



圖十二：(I0)原影像 (I1)嵌入影像 PSNR=25.1271 (I2)篡改影像 PSNR=20.1541 (I3)偵測篡改區塊 (I4)無失真還原未篡改區塊 PSNR=21.2649 (I5)修復篡改區塊 PSNR=30.3989 (I6)尚未修補區塊(I7)修補遺失回復資訊之篡改區塊 PSNR=30.3989。



圖十三：(J0)原影像 (J1)嵌入影像 PSNR=25.1271 (J2)篡改影像 PSNR=12.7238 (J3)偵測篡改區塊 (J4)無失真還原未篡改區塊 PSNR=12.8650 (J5)修復篡改區塊 PSNR=18.4700 (J6)尚未修補區塊 (J7)修補遺失回復資訊之篡改區塊 PSNR=27.3674。