基於區塊關聯的影像驗證與復原方法

邱金鴻　　韓文舜　　許慶昇
銘傳大學資訊管理學系　銘傳大學資訊管理學系　銘傳大學資訊管理學系
Mamacox94@hotmail.com　Hannibal0416@homail.com　cshsu@mail.mcu.edu.tw

摘　要

近年來，數位浮水印被廣泛地使用在數位影像的著作權保護與完整性驗證上，而完整性驗證大多使用易碎型浮水印技術來達成。易碎型浮水印的做法，是將影像特徵值轉換成浮水印並藏入影像本身；當要進行影像驗證時，便取出預藏在影像中的浮水印來與影像特徵值比對，以判定影像是否遭受竄改，甚至進一步將被竄改的區域復原。許多影像驗證方法都以區塊為基礎，並將區塊視為獨立的個體來進行浮水印的產生與嵌藏，而忽略了區塊間的關聯性與周圍影像資訊之重要性。此外，許多方法將竄改偵測的驗證訊息與影像復原訊息視為不同的浮水印，而沒有考慮到浮水印可同時具有上述兩種功能的可能，所以無法達到交叉驗證與互為備援的效果。因此,本研究將嘗試使用區塊間的關聯性，來發展出一套影像驗證與復原方法，使浮水印能同時具備影像竄改偵測與復原的功能，以提昇影像驗證的準確性與復原的品質。實驗結果顯示，我們的方法在竄改偵測的錯誤率及影像復原品質上皆有不錯的表現。

關鍵字：易碎型浮水印、影像驗證、影像竄改偵測。

1. 前言

近年來，由於電腦的普及與網路的發達，使得數位資訊與數位作品(如：文字、影像、聲音、影片)的傳送、流通與交易變得更加地便利與頻繁。然而，經過數位化之後的訊息或作品卻也存在著容易遭受變造與塗改的問題，如果訊息接收者無法有效辨別訊息的真偽，將可能導致可怕的危害與糾紛。例如：醫學影像若被有心人士竄改,可能造成保險理賠糾紛或醫師誤診；新聞圖片如遭竄改或破壞，可能會扭曲新聞真相、違反新聞道德或甚至非法侵害他人權益；刑事案件佐證影像如遭竄改或破壞，可能會導致法官的誤判，進而嚴重侵害兩位當事人的合法權益；個人私密影像如遭竄改或破壞，可能會導致個人名譽受損。因此，如何有效驗證數位影像的完整性，已經成為一個熱門且重要的研究課題。

近年來，數位浮水印被廣泛地使用在數位影像的著作權保護與完整性驗證上。數位浮水印依抵抗攻擊的強度可分為強韌型浮水印、易碎型浮水印與半易碎型浮水印三種。強韌型浮水印具有不易被破壞的特性，因此它經常被應用於影像的版權保護上(Agreste & Andaloro, 2008; Kamel & Albluwi, 2009; Tsolis, Nikolopoulos, Drossos, Sioutas, & Papatheodorou, 2009; Wu, Ma, Dong, & Reveret, 2008)。易碎型浮水印對於攻擊的抵抗力較弱，容易因為影像遭到破壞而無法存活，因此它經常被應用在影像的竄改偵測上(Wang & Tsai, 2008)。半易碎浮水印的強韌度則介於強韌型與易碎型兩者之間，它對於非惡意的攻擊(例如 JPEG 失真壓縮)具有較高的抗攻擊

能力，然而對於惡意的竄改攻擊則較為敏感(Chamlawi, Khan, & Usman, 2010)。目前，大部份的影像竄改偵測方法，都是透過易碎型或半易碎型浮水印技術來達成。

易碎型浮水印的做法，是將影像特徵值轉換成浮水印並藏入影像本身；當要進行影像驗證時，便取出預藏在影像中的浮水印來與影像特徵值比對，以判定影像是否遭受竄改，甚至進一步將被竄改的區域復原。關於浮水印的隱藏與偵測，在技術上可分為頻率域與空間域兩類。頻率域的技術使用一些轉換函數(例如：快速傅立葉轉換、離散餘弦轉換、離散小波轉換等)，將空間域的像素值轉換成頻率域的係數，然後再將浮水印隱藏於這些頻率域的係數當中(Aslantas, Ozer, & Ozturk, 2009; Lin & Lin, 2009; Yen & Tsai, 2008)。而空間域的技術則是直接修改像素值，來達成浮水印的嵌藏與偵測。許多空間域的易碎型浮水印方法，都是以區塊為單位來進行驗證訊息的產生與隱藏，以偵測區塊是否遭到竄改。此種方法能藏入的資訊量取決於區塊的大小；區塊愈大，則能隱藏的訊息量就愈多，所以竄改偵測的錯誤率就愈低，但竄改定位精準度就會愈差，也就是只能精準到區塊的大小(Zhang, Wang, Qian, & Feng, 2010)。有些方法則是直接以像素為單位來做竄改偵測，雖然這類方法的訊息隱藏空間較少，但定位精準度卻可以精準定位到單一像素(Zhang & Wang, 2007)。此外，有些方法則是結合區塊驗證與像素驗證的技巧，來達到降低驗證錯誤率與提昇定位精準度的目標(Zhang & Wang, 2009)。空間域的易碎型浮水印技術，通常是將浮水印藏入影像的最低位元平面(Least Significant Bits, LSBs)中，以避免對影像造成太多的破壞而影響了影像品質。因為影像中能被用來隱藏浮水印的空間是有限的，所以為了要提高竄改偵測的正確性，就要將資訊隱藏的空間大量使用在儲存驗證訊息，因而犧牲了影像復原的品質；反之，要提高影像復原的品質，就要將資訊隱藏空間大量使用在儲存復原訊息上，而降低竄改偵測的正確性。許多影像驗證方法都以區塊為基礎，並將區塊視為獨立的個體來進行浮水印的產生與嵌藏，而忽略了區塊間的關聯性與周圍影像資訊之重要性。此外，許多方法將竄改偵測的驗證訊息與影像復原訊息視為不同的浮水印，而沒有考慮到浮水印可同時具有上述兩種功能的可能，所以無法達到交叉驗證與互為備援的效果。基於影像周圍的資訊相近的特性，本研究將嘗試使用區塊間的關聯性，來發展出一套影像驗證與復原方法，使浮水印能同時具備影像竄改偵測與復原的功能，以提昇影像驗證的準確性與復原的品質。

2. 文獻探討

近年來，有許多影像竄改偵測與影像復原的方法相繼被提出。Zhang and Wang (2009)提出了一個結合區塊驗證與像素驗證的易碎型浮水印方法，他們的方法不但能判斷一個 8×8 區塊是否遭受竄改，還能精準定位被竄改的像素。在驗證時，他們的方法可以將遭受竄改的區塊正確地標示出來，然後再根據未遭破壞的驗證資訊判斷哪些像素遭受竄改。這個方法在竄改比例不高的時候表現得還不錯，然而當竄改比例提高時，這個方法不只會產生很多誤判，也會無法精準定位到被竄改的像素。特別是，當每個 8×8 區塊內都只有一個像素被竄改時，也就是竄改比例為 1/64 時，因為無法取得任何可用的驗證訊息，所以會將區塊內的其他 63 像素全部認定為遭受竄改，而導致很高的偽陽性錯誤率。

Lin, Hsieh, and Huang (2005)提出了一

個階層式的影像竄改偵測與復原方法，他們將原始圖形切割成不重疊的4×4區塊，每個 4 × 4 區塊再細分成四個不重疊的 2 × 2 子區塊。接著，利用小區塊與大區塊之像素平均值的關係來產生一個位元的認證碼，再利用子區塊前六個 MSBs 來產生一個同位元檢查碼，最後將子區塊的平均值、認證碼與同位元檢查碼藏入區塊中，完成浮水印嵌入。其竄改偵測共分為四個階段：第一階段檢測 2 × 2 子區塊的認證碼與同位元檢查碼，以判斷子區塊是否遭竄改；在第二階段中，如果 4×4 區塊中有一個子區塊被判定為錯誤，則整個區塊判定為錯誤；第三個階段檢測以 4×4 區塊為中心，若其周圍的八個大區塊中，有五個或五個以上的區塊被標示為錯誤，則判定中心區塊為錯誤；第四階段則是檢測有無向量量化攻擊。在復原階段，如果浮水印藏入的區塊被標示為正確則直接提取浮水印資訊來做復原；反之，如果被標示為錯誤，則使用周圍影像資訊的平均值來做復原。他們的方法在第一階段檢測中只使用兩個位元來做竄改偵測，因此會有很高的誤判機率。而第二階段雖然能降低第一階段的偽陰性錯誤，但只要有一個被判定錯誤就會導致其他三個也被判定錯誤，會發生原本是正確的區塊卻被判定為錯誤的機率，導致偽陽性錯誤的發生。第三階段也是只要周圍超過五個被判定為錯誤，就有可能導致中心區塊原本是正確的卻被誤判為錯誤的偽陽性錯誤。他們的方法雖然有考慮到使用周圍影像的資訊來做竄改偵測，但沒有使用在影像復原上。而在影像復原階段，只要浮水印嵌入的區塊被判定為錯誤，就只能使用周圍影像的資訊來做復原，因而降低了影像復原的品質。

Lee and Lin (2008)提出雙重浮水印的影像偵測及復原的方法，他們將原始圖形切割成許多不重疊的 2×2 區塊，將影像水平切割把上部份的區塊與下部份區塊分為兩組對應的區塊稱為夥伴區塊，分別算出夥伴區塊的像素平均值，結合夥伴區塊的像素平均值，做互斥或閘運算及同位元檢查編碼產生兩個位元的驗證資訊，最後產生出十二個位元的浮水印資訊，將這十二個位元的浮水印分別藏入對應的另一組夥伴區塊的三個 LSBs 中，就完成了浮水印藏入。在竄改偵測中分三個階段，第一個階段比對藏入的認證碼與算出的認證碼是否相同，第二階段則是以區塊為中心將其周圍 3×3 的八個區塊分成四組，只要其中有一組的區塊全部被標示為錯誤，則中心的區塊也一起標示為錯誤，在第三階段也是以區塊為中心其周圍 3 × 3 的八個區塊中有五個或五個以上的區塊被標示為錯誤，則判定中心區塊為錯誤，在復原階段，被判定為錯誤的區塊有兩次的機會可以做復原，如對應的區塊被標示為正確則直接提取浮水印資訊來做復原，如對應的區塊被標示為錯誤還可從對應區塊的另一個夥伴區塊取出浮水印來做復原，如果對應區塊的夥伴區塊也被標示為錯誤則使用周圍影像資訊的平均值來做復原。他們的方法因為藏入了兩份復原的資訊，一個復原資訊被破壞掉還有另一個夥伴區塊的機會做復原，但也因為這樣壓縮到能做竄改偵測碼的空間，在第一階段只使用兩個位元來做竄改偵測有相當的機率會誤判，而在第二、三階段也會因為周遭區塊的因素導致中心區塊發生偽陽性的錯誤，雖然有兩次復原的機會，當兩個浮水印都被破壞時也只能使用周圍影像的平均值來做復原，復原的效果也不佳。

3. 本文提出的方法

假設原始影像為 $m \times n$ 的非失真灰階

影像，其中 m 代表影像的長度(列數)，而 n 則代表影像的寬度(欄數)。令 Q 代表大於零的整數。圖 1 為本研究的浮水印嵌藏規則，其中 P_i 代表一個小區塊中的第 i 個像素，而 B_j 則代表一個像素值中的第 j 個位元。以下為浮水印的嵌藏、影像驗證與影像復原方法的詳細步驟。

	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
P_0						a_1	a_5	c_1
P_1						a_2	b_1	c_2
P_2						a_3	b_2	H
P_3						a_4	b_3	H

圖 1：浮水印嵌藏規則

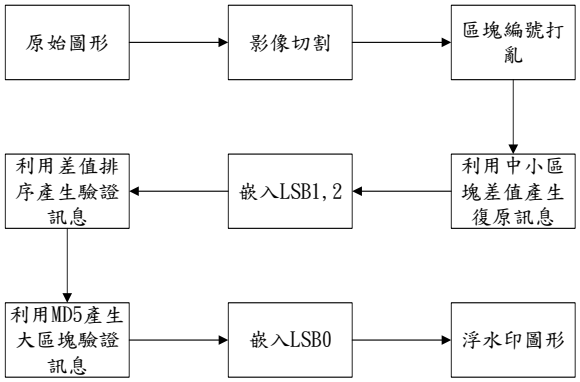


圖 2：浮水印嵌入流程圖

3.1 浮水印的產生與嵌藏

我們將利用周圍影像的資訊與 MD5 函數，產生出驗證訊息與復原訊息，將其藏入原始影像中，圖 2 為本研究的浮水印嵌入流程圖。

步驟一：首先將圖形切割成不重疊的 4×8 大區塊；接著，再將每個大區塊切割成兩個不重疊的 4×4 中區塊；最後，再將每個中區塊切割成四個不重疊的 2 × 2 小區塊。

步驟二：我們先將小區塊給予編號，從 0 開始由左至右由上至下編號；接著，我們將原始影像切割成四大區域，並對每個區域內部的小區塊編號使用偽隨機序列來做打亂。然後我們對四大區域的編號做置換，以中區塊為一個單位，每個區域相同位置的中區塊為一組，將其內部的 4 個小區塊編號置換到其他 3 大區域，相同號碼的對換，置換方式如圖 3 所示。舉例來說，左上區域內部小區塊的浮水印，將會分散的嵌藏到其他三個區域，當左上區域的影像被破壞時，可由其他三個區域提取浮水印來做影像復原。

步驟三：計算每一個 2 × 2 小區塊的像素值平均數 $\overline{g}_s$ 與其相對應的 4 × 4 中區塊的像素值平均數 $\overline{g}_m$。令 $g^{\langle k\rangle}$ 代表將一非負整數 g 的前 k 個 LSBs 設定為零的結果，其公式如下：

$$g^{\langle k\rangle} = 2^k \cdot \left\lfloor \frac{g}{2^k} \right\rfloor 。 \quad (1)$$

我們使用下列公式產生驗證資訊 b_1、b_2 與 b_3：

$$b_1 = \begin{cases} 0 & \text{if } \overline{g}_s^{\langle 3\rangle} \le \overline{g}_m^{\langle 3\rangle}, \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

$$b_2 = \left\lfloor \frac{y}{2} \right\rfloor \bmod 2 \ , \quad (3)$$

$$b_3 = y \bmod 2 \ , \quad (4)$$

其中

$$y = \arg\min_{x\in\{0,1,2,3\}} \left\| \overline{g}_s^{\langle 3\rangle} - \overline{g}_m^{\langle 3\rangle} \right| - x\cdot Q \Big| 。 \quad (5)$$

驗證資訊 b_1 用於區別中、小區塊平均值的大小，而 b_2 與 b_3 則指出將小區塊平均值加或減幾倍的 Q 會最接近中區塊平均值。

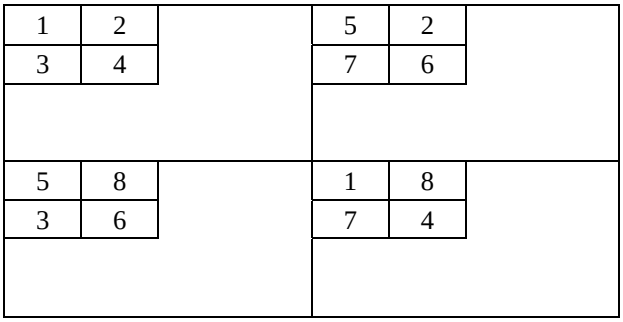


圖 3：打亂對照圖

步驟四：根據圖 1 的浮水印嵌藏規則，將小區塊像素平均值前五個 MSBs (a_1, a_2, a_3, a_4, a_5)與(b_1, b_2, b_3)，嵌藏於打亂後的對應小區塊內。

步驟五：利用中區塊平均值 $\bar{g}_m$ 與其內部每個小區塊平均值 $\bar{g}_s$ 的差值 d 產生小區塊驗證碼 c_1、c_2，並根據圖 1 的浮水印嵌藏規則，將之隱藏於小區塊本身之中。小區塊驗證碼產生方式如下：

$$c_1 = \left(\left\lfloor \frac{r(d)}{2} \right\rfloor \bmod 2 \right) \otimes z_1 \ , \qquad (6)$$

$$c_2 = \left(r(d) \bmod 2 \right) \otimes z_2 \ , \qquad (7)$$

其中

$$d = \bar{g}_m^{\langle 1 \rangle} - \bar{g}_s^{\langle 1 \rangle} \ , \qquad (8)$$

z_1 與 z_2 代表小區塊座標壓縮成的兩個位元，⊗代表 bit-wise XOR，且 $r(d)$代表中區塊內的四個差值經由小至大排序後，排列在 d 之前的差值個數。

步驟六：利用 MD5 函數與 XOR 折疊法，將大區塊內部的 240 個位元(圖 1 中的 H 位元除外)、區塊座標與密鑰，轉換成 16 個位元的大區塊驗證資訊 H；接著，再根據圖 1 的浮水印嵌藏規則，將這 16 個位元嵌藏於大區塊本身中。

3.2 影像竄改偵測

在這節我們介紹竄改偵測的技術，拿到一張嵌入浮水印的圖形，可以利用下面步驟來檢驗影像是否有遭受到竄改，當檢測到影像遭受的竄改時，再進一步利用嵌入的浮水印做影像復原，圖 4 為本研究的竄改偵測與復原流程圖。

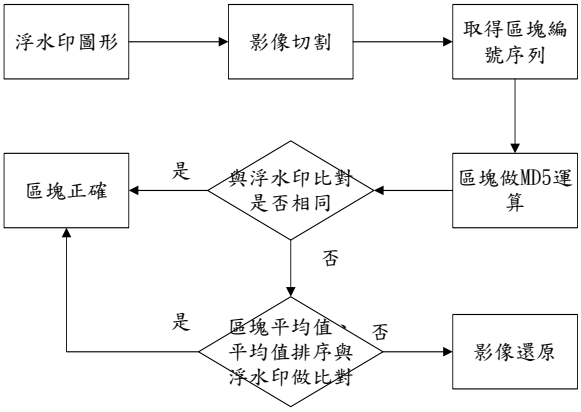


圖 4：竄改偵測與復原流程圖

步驟一：根據大區塊驗證資訊判斷大區塊是否遭受竄改。若判斷遭到竄改，則標示大區塊內部所有小區塊為錯誤；反之，則標示大區塊內部所有小區塊為正確。

步驟二：對於每個被標示為錯誤的小區塊，若其相對應的小區塊被標示為正確，則找出預藏的浮水印(a_1, a_2, a_3, a_4, a_5)，來跟小區塊的像素平均值做比對。若比對結果一致，則進一步根據(c_1, c_2)驗證碼來判斷小區塊是否正確，若是，則將此一小區塊標示為正確。

3.3 影像復原

在這節將介紹被竄改的圖形如何來做影像復原，要是沒有辦法直接提取到小區塊平均值的浮水印來做復原，還可以利用中區塊內部其他小區塊的浮水印資訊來做復原，最後則是使用周圍 3×3 影像平均值來做復原，以下是影像復原的四個步驟。

步驟一：對於每一個被標示為錯誤的小區塊，若其對應小區塊被標示為正確，則使用對應小區塊中的(a_1, a_2, a_3, a_4, a_5)做復原。

步驟二：對於每一個尚未被復原的錯誤小區塊，若其中區塊內部的另外三個小區塊的對應小區塊其中有一個被標示為正確，則取出其(a_1, a_2, a_3, a_4, a_5)與(b_1, b_2, b_3)的復

原資訊，並利用下列中區塊平均值估計式來復原小區塊：

$$R=\left(\sum_{i=1}^{5}a_i\cdot 2^{8-i}\right)+(1-2b_1)\cdot(2b_2+b_3)\cdot Q \quad (9)$$

步驟三：對於每一個尚未被復原的錯誤小區塊，如果其所屬中區塊內的另外三個小區塊其中有一者被標示為正確且

$$(c_1, c_2)\otimes(z_1, z_2)=(0, 0)，$$

表示此區塊的像素平均值最接近中區塊的像素平均值，則我們使用這個區塊的像素平均值來復原其中區塊內部的錯誤小區塊。

步驟四：對於每一個尚未被復原的錯誤小區塊，利用周圍八個區塊中正確區塊的像素平均值來做復原。

4. 實驗結果

為了驗證本方法的效能，我們首先定義下列偵測錯誤率評估指標：

$$FNR = FN/(FN + TP)，\quad (10)$$

$$FPR = FP/(FP + TN)，\quad (11)$$

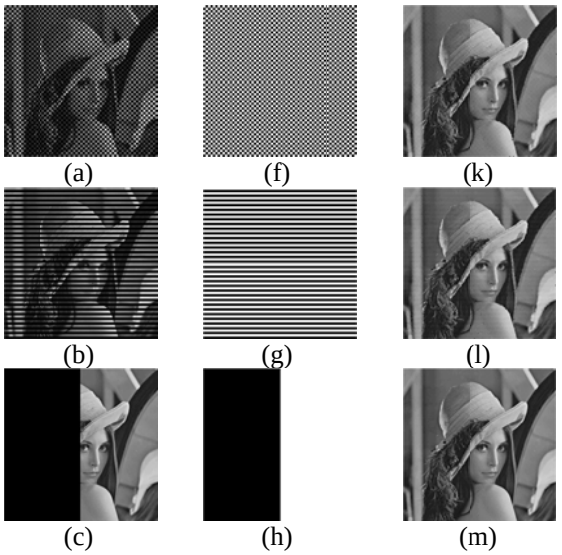
其中*FN*代表False Negative的像素個數，*FP*代表False Positive的像素個數，*TN*代表True Negative的像素個數，而*TP*則代表True Positive的像素個數。此外，我們也使用下列指標來評估影像復原的品質：

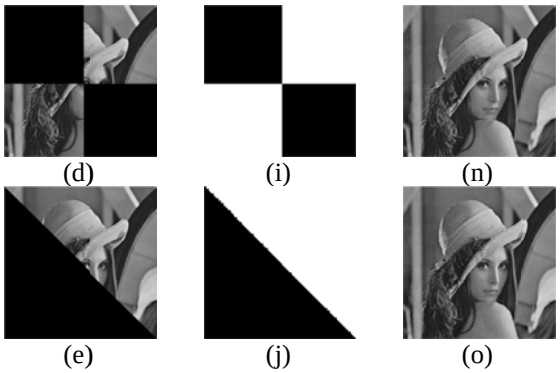
$$PSNR=10\times\log\frac{255^2}{MSE}\ \text{(dB)}，\quad (12)$$

其中

$$MSE=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{i,j}-p'_{i,j})^2\ 。\quad (13)$$

在以下的實驗中，我們設定參數 $Q = 2$。為了驗證本研究之方法的效能，我們進行了下列多項攻擊實驗，並與 Lee & Lin (2008)的方法做比較。圖 5 為 50%的剪裁攻擊實驗，其中圖 5 (a) ~ 圖 5 (e)為五種不同的 50%剪裁攻擊；圖 5 (f) ~圖 5 (j)為偵測結果，黑點表示偵測為遭受竄改的區域，白點表示偵測為未遭受竄改的區域；圖 5 (k) ~ 圖 5 (o)為復原後的結果。表 1 為 50%剪裁攻擊的實驗數據，其中 *PSNR* 代表影像復原品質。圖 6 為 75%的剪裁攻擊實驗，其中圖 6 (a) ~ 圖 6 (g)為七種不同的 75%剪裁攻擊，圖 6 (h) ~ 圖 6 (n)為偵測結果，而圖 6 (o) ~ 圖 6 (u)則為復原後的結果。圖 6 的實驗數據如表 2 所示。接著，我們進行圖 7 的拼貼攻擊實驗，其中我們將影像的某些重要區域複製並拼貼於他處，圖 7 (a) ~ 圖 7 (c)為各種拼貼攻擊的實驗，圖 7 (d) ~ 圖 7 (f)為 Lee & Lin (2008)的偵測結果，其中黑點表示偵測為遭受竄改的區域，白點表示偵測為未遭受竄改的區域，而圖 7 (g) ~ 圖 7 (i)則為 Lee & Lin (2008)的復原結果。圖 7 (j) ~ 圖 7 (l) 為本研究方法的偵測結果，圖 7 (m) ~ 圖 7 (o)為本研究方法復原後的結果。圖 7 的相關實驗數據如表 3 所示。最後，我們進行圖 8 的合成攻擊實驗，其實驗數據如表 4 所示。以上實驗結果顯示，在拼貼與合成攻擊時，我們的方法的影像復原品質皆優於 Lee & Lin (2008)的方法，且在遭受到拼貼攻擊時我們更能有效的偵測出錯誤。





(d) (i) (n)

(e) (j) (o)

圖 5：50%剪裁攻擊

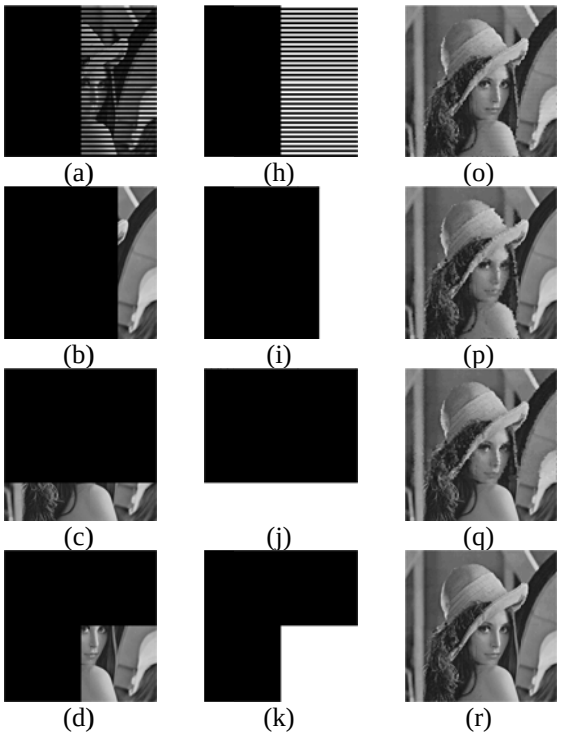
表 1：50%剪裁攻擊實驗數據

Lee & Lin (2008)的方法

竄改方式	*FNR*	*FPR*	*PSNR*
圖 5 (a)	0.0000	0.0000	27.75
圖 5 (b)	0.0000	0.0000	30.89
圖 5 (c)	0.0000	0.0000	31.75
圖 5 (d)	0.0000	0.0000	32.40
圖 5 (e)	0.0000	0.0097	30.49
平均值	0.0000	0.0019	30.66

本研究的方法

竄改方式	*FNR*	*FPR*	*PSNR*
圖 5 (a)	0.0000	0.0000	29.35
圖 5 (b)	0.0000	0.0000	29.09
圖 5 (c)	0.0000	0.0000	30.04
圖 5 (d)	0.0000	0.0000	30.76
圖 5 (e)	0.0000	0.0079	28.82
平均值	0.0000	0.0016	29.61

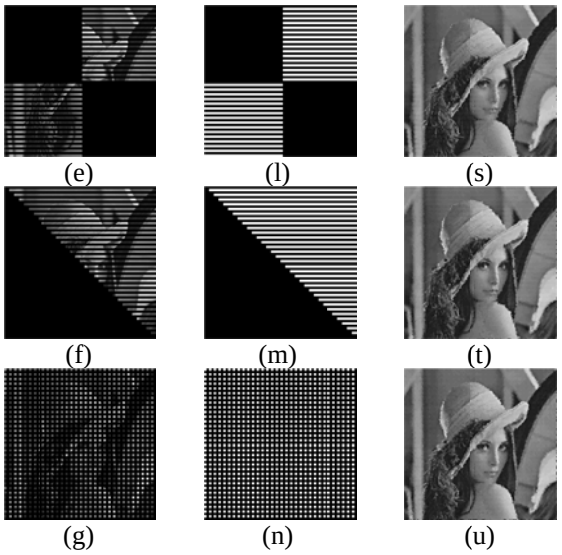


(a) (h) (o)

(b) (i) (p)

(c) (j) (q)

(d) (k) (r)



(e) (l) (s)

(f) (m) (t)

(g) (n) (u)

圖 6：75%剪裁攻擊

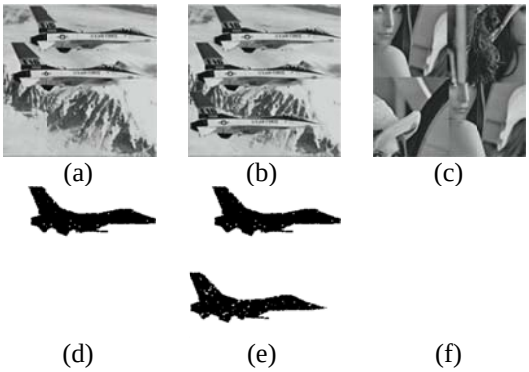
表 2：75%剪裁攻擊實驗數據

Lee & Lin (2008)的方法

竄改方式	*FNR*	*FPR*	*PSNR*
圖 6 (a)	0.0000	0.0038	26.20
圖 6 (b)	0.0000	0.0000	26.30
圖 6 (c)	0.0000	0.0000	18.18
圖 6 (d)	0.0000	0.0001	20.25
圖 6 (e)	0.0000	0.0038	24.56
圖 6 (f)	0.0000	0.0039	23.71
圖 6 (g)	0.0000	0.2422	23.01
平均值	0.0000	0.0362	23.17

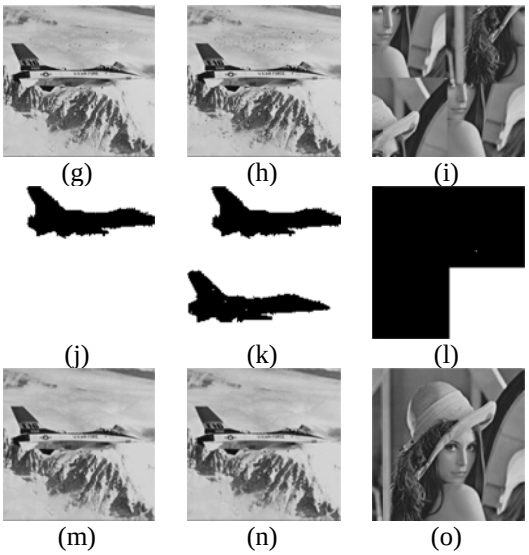
本研究的方法

竄改方式	*FNR*	*FPR*	*PSNR*
圖 6 (a)	0.0000	0.0000	25.86
圖 6 (b)	0.0000	0.0000	25.26
圖 6 (c)	0.0000	0.0000	26.00
圖 6 (d)	0.0000	0.0000	25.90
圖 6 (e)	0.0000	0.0000	26.30
圖 6 (f)	0.0000	0.0000	25.49
圖 6 (g)	0.0000	0.0000	25.98
平均值	0.0000	0.0000	25.83



(a) (b) (c)

(d) (e) (f)



(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

圖 7：拼貼攻擊

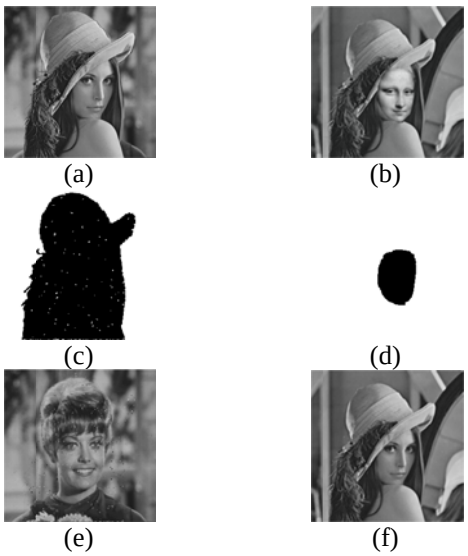
表 3：拼貼攻擊實驗數據

Lee & Lin (2008)的方法

竄改方式	竄改比例	*FNR*	*FPR*	*PSNR*
圖 7 (a)	11%	0.0200	0.0033	36.16
圖 7 (b)	21%	0.0295	0.0085	29.30
圖 7 (c)	75%	1.0000	0.0000	12.70
平均值	36%	0.3498	0.0039	26.05

本研究的方法

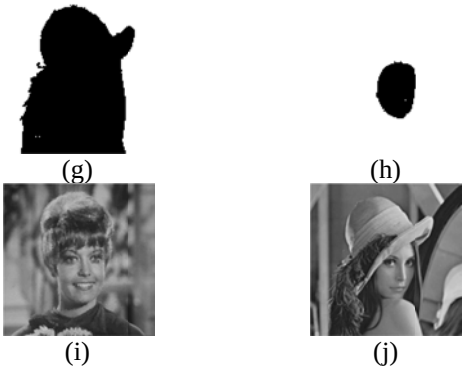
竄改方式	竄改比例	*FNR*	*FPR*	*PSNR*
圖 7 (a)	12%	0.0019	0.0115	43.47
圖 7 (b)	24%	0.0025	0.0280	32.41
圖 7 (c)	75%	0.0001	0.0621	25.39
平均值	37%	0.0015	0.0339	33.76



(a) (b)

(c) (d)

(e) (f)



(g) (h)

(i) (j)

圖 8：合成攻擊

表 4：合成攻擊實驗數據

Lee & Lin (2008)的方法

竄改方式	竄改比例	*FNR*	*FPR*	*PSNR*
圖 8 (a)	54%	0.0092	0.0156	28.12
圖 8 (b)	7%	0.0008	0.0036	40.58
平均值	31%	0.0050	0.0096	34.35

本研究的方法

竄改方式	竄改比例	*FNR*	*FPR*	*PSNR*
圖 8 (a)	54%	0.0005	0.0305	29.57
圖 8 (b)	7%	0.0012	0.0044	41.12
平均值	31%	0.0009	0.0175	35.35

5. 結論與未來工作

本研究利用區塊間的關聯性，使浮水印可同時具備竄改驗證與影像復原兩種功能，以達到交叉驗證與互為備援的效果。實驗結果顯示，相較於 Lee & Lin (2008)的方法，我們的方法不管是在偽陰性錯誤率、偽陽性錯誤率以及影像復原品質上皆有較佳的表現。未來我們將進行更多的理論分析，以強化結論的說服力。此外，我們也將在不同的參數值下，進行更完整的實驗，以釐清不同參數值的可能影響。

參考文獻

[1] Agreste, S. & Andaloro, G. (2008). A new approach to pre-processing digital image for wavelet-based watermark. Journal of Computational and Applied Mathematics, 221(2), 274-283.

[2] Aslantas, V., Ozer, S., & Ozturk, S. (2009). Improving the performance of DCT-based fragile watermarking using intelligent optimization algorithms. Optics Communications, 282(14), 2806-2817.

[3] Chamlawi, R., Khan, A., & Usman, I. (2010). Authentication and recovery of images using multiple watermarks. Computers & Electrical Engineering, 36(3), 578-584.

[4] Kamel, I. & Albluwi, Q. (2009). A robust software watermarking for copyright protection. Computers & Security, 28(6), 395-409.

[5] Lee, T. & Lin, S. (2008). Dual watermark for image tamper detection and recovery. Pattern Recognition, 41(11), 3497-3506.

[6] Lin, P., Hsieh, C., & Huang, P. (2005). A hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recognition, 38(12), 2519-2529.

[7] Lin, T. & Lin, C. (2009). Wavelet-based copyright-protection scheme for digital images based on local features. Information Sciences, 179(19), 3349-3358.

[8] Tsolis, D., Nikolopoulos, S., Drossos, L., Sioutas, S., & Papatheodorou, T. (2009). Applying robust multibit watermarks to digital images. Journal of Computational and Applied Mathematics, 227(1), 213-220.

[9] Wang, S. & Tsai, S. (2008). Automatic image authentication and recovery using fractal code embedding and image inpainting. Pattern Recognition, 41(2), 701-712.

[10] Wu, X., Ma, L., Dong, Z., & Reveret, L. (2008). Robust watermarking motion data with DL-STDM. Computers & Graphics, 32(3), 320-329.

[11] Yen, E. & Tsai, K. (2008). HDWT-based grayscale watermark for copyright protection. Expert Systems with Applications, 35(1-2), 301-306.

[12] Zhang, X. & Wang, S. (2007). Statistical fragile watermarking capable of locating individual tampered pixels. IEEE Signal Processing Letters, 14(10), 727-730.

[13] Zhang, X. & Wang, S. (2009). Fragile watermarking scheme using a hierarchical mechanism. Signal Processing, 89(4), 675-679.

[14] Zhang, X., Wang, S., Qian, Z., & Feng, G. (2010). Reversible fragile watermarking for locating tampered blocks in JPEG images. Signal Processing, 90(12), 3026-3036.