利用影像內插與失真評估之資料隱藏演算法

林宗瀚[1,*] 王宗銘[2]

摘 要

本篇論文針對灰階影像提出了一個具有高藏量、高僞裝影像品質，以及可回復之資料隱藏技術。我們的演算法是基於影像內插並使用混合式OPAP搭配LSB-matching所做的資訊隱藏技術。本研究利用平均每個像素的期望變動量分析，可以明確的預測在僞裝影像品質固定之下其所產生之最高藏量，以及在嵌入量固定之下其所產生之最高僞裝影像品質。我們發現Jung等人之演算法在某些情況下無法將秘密訊息完全正確取出，我們修正其演算法使秘密訊息能夠正確無誤取出。我們再另外提出一個資料隱藏方法與Jung等人之方法比較。於本論文中，我們不僅改正了Jung等人演算法之錯誤。我們另外提出的方法具有高藏量、高僞裝影像品質以及可回復之特性，僅需要極少的額外資訊來取出秘密訊息。此外，本方法也可抵擋RS偵測攻擊。

關鍵詞：資料隱藏、影像內插、失真評估、僞裝學、僞裝分析。

DATA HIDING USING IMAGE INTERPOLATION AND DISTORTION ESTIMATION SCHEMES

Zong-Han Lin[1,*] Chung-Ming Wang[2]

ABSTRACT

This paper proposes a data hiding algorithm that provides high capacity, high visual quality, and reversibility for gray level image. Inspired from the image interpolation proposed by Jung et al., we use hybrid OPAP and LSB-matching to achieve goal of the data hiding. When the visual quality of the stego image is fixed, our method can predict the maximal capacity through an analysis of the Expected Variation per Pixel (EVP) technique. On the other hand, the EVP technique can predict an optimal visual quality of the stego image when we fix the capacity of the stego image. We discover that Jung *et al*.'s method is incapable of correctly extracting secret data in some conditions. We correct their algorithm so that it can extract message without encountering any errors. Experimental results verify that our method provides high capacity and high visual quality. In addition, our scheme supports the feature of reversibility, requiring very little extra information to extract the secret message. Furthermore, the proposed method can withstand the RS steganalysis attack.

Key Words: data hiding, image interpolation, distortion estimation, steganography, steganalysis.

1 國立中興大學資訊科學與工程學系碩士班研究生；*Institute of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan, R.O.C.*
2 國立中興大學資訊科學與工程學系教授
* Corresponding Author, E-mail: s9756044@csmail.nchu.edu.tw

一、前　言

資料隱藏是將秘密訊息藏入數位化的多媒體中的一門技術 [1, 2]。數位化的多媒體指的是影像、聲音、視訊等這些媒介。以對影像做資料隱藏來說，在嵌入秘密訊息之前的影像稱爲掩護影像（Cover Image），而嵌入之後的影像稱爲僞裝影像（Stego Image）。一個基本的資料隱藏流程就是送方將秘密訊息嵌入掩護影像中產生僞裝影像，然後透過網際網路將僞裝影像傳送給收方，收方接收到之後再將僞裝影像中的秘密訊息取出。然而，一個資料隱藏技術的優劣可由以下四個指標來評斷：1. 藏量（Capacity）：一張影像可以嵌入的秘密訊息的位元個數；2. 品質（Quality）：嵌入資料後的僞裝影像與未嵌入資料的掩護影像之間很難用人眼直接辨識出來，在資料隱藏技術中通常是以訊號對雜訊比（Peak Signal to Noise Ratio, PSNR）值來做評估，僞裝影像品質至少要高於30 dB以上才能夠免於被人類視覺系統所察覺；3. 不需要掩護影像（Not Need Cover Image）：從僞裝影像要取出秘密訊息時，不需透過掩護影像就能將訊息取出；4. 可回復（Reversibility）：在不需要任何其他資訊的情況下，可從僞裝影像回復到掩護影像，這樣才能夠驗證訊息的完整性。

在資料隱藏研究的領域雖已有許多專家學者對此提出了各式各樣的見解 [3, 4, 5, 6, 7]，但是在文獻上很少有使用影像內插爲基礎來做的資料隱藏技術 [8]。在Jung [8]等人的方法中，先將解析度爲512 × 512之影像縮小爲解析度256 × 256之影像，再透過他們提出的影像內插方法將之內插放大爲512 × 512之影像，最後再將秘密訊息嵌入內插放大後的影像中。其演算法的細節我們將在下個章節說明。他們的實驗結果分成兩個部份來呈現：第一個部份爲影像內插的品質；第二個部份爲藏量與僞裝影像品質。我們發現Jung等人的方法產生的影像內插品質不夠好，平均每張影像內插品質只有24.39 dB，藏量與僞裝影像品質皆有可改進之處，藏量平均每張影像嵌入251,028個位元，大約只有0.96 bpp（bits per pixel），僞裝影像品質平均每張影像約爲39.67 dB。此外，我們也發現Jung等人的方法無法完全正確地取出秘密訊息，我們修正他們的方法使其可完全正確將秘密訊息取出。在下一個章節我們會做明確的說明。

在本篇論文中，我們針對了以上所提及的缺失來做改進，就影像內插方面來說：我們實驗用的影像一共有六張，但Jung等人所使用的影像僅有四張。爲了達到比較之公允性我們實驗數據僅比較Lena、Baboon、Jet、Peppers等影像。以此四張影像而言使用我們提出的影像內插方法（EDP + MED）所產生的效果較佳，所產生出來的影像內插品質平均每一張影像約可達29.25 dB，較Jung等人的方法平均高出了4.86 dB。整體而言影像內插品質提升了2.81 dB-6.3 dB。就藏量及僞裝影像品質而言：我們做了平均每個像素的期望變動量分析，可以明確的預測出在嵌入多少個位元之下其所產生之僞裝影像品質，以及在多少僞裝影像品質之下其所產生之藏量。在僞裝影像品質與Jung等人方法的數據相近之下我們方法得到的藏量較他們多4.86%-118.10%；在嵌入量與Jung等人方法的數據相近之下我們方法得到的僞裝影像品質較他們多1.39-13.76 dB。

本論文其他章節的架構如下：第二節說明相關的研究，主要是回顧 [8]以及說明他們方法的缺失並將其改正；第三節詳細敘述我們提出的演算法；第四節展示並且說明我們的實驗結果；第五節展示資訊僞裝偵測結果；最後，第六節總結本文並且進一步地提出未來的研究方向。

二、相關文獻探討

2.1 Jung等人影像內插與資訊隱藏演算法之缺失與改進

本節中，我們回顧本方法欲比較的資料隱藏技術 [8]；第一節我們回顧Jung等人所提的方法；第二節我們敘述Jung等人所提方法之缺失。具體而言，我們將以二個例子說明該法在所提之例證下，無法還原原始掩護影像，導致無法正確取出秘密訊息。

2.1.1 Jung等人所提的影像內插方法

其演算法流程如下所述：首先將解析度爲512 × 512的輸入影像縮小爲解析度256 × 256的原始影像，再使用影像內插將解析度爲256×256的原始影像放大爲解析度512×512的掩護影像，然後將秘密資訊嵌入掩護影像中。取出秘密訊息時不需要掩護影像就能將秘密資訊取出以及還原到解析度爲256 × 256的原始影像，但是經過我們實作後的結果證實，Jung他們的方法在取出秘密訊息時有瑕疵，我們將在之後舉實例說明之。此外，我們將針

對此缺失說明其改進方式。其演算法主要分成兩個部分：第一個部分爲影像內插；第二個部分爲資料隱藏。

1. 影像內插

Jung他們所提出的影像內插法名爲Neighbor Mean Interpolation（NMI），是使用鄰近像素的平均值來取得內插像素值。假設解析度爲256 × 256的原始影像像素爲$p(x_i, y_i)$，$i = 0,\cdots,255$放大之後解析度爲512 × 512的掩護影像像素爲$p'(x_j, y_j)$，$j = 0,\cdots,511$，放大方式是將原始影像的像素值直接複製到像素座標爲原始影像像素座標二倍的掩護影像中，例如原始影像像素座標爲$p(0, 0)$就直接將該像素值複製到$p'(0, 0)$中，同樣地，當原始影像像素座標爲$p(0, 1)$就直接將該像素值複製到$p'(0, 2)$中，以此類推直到將原始影像像素值全部複製到放大的影像中爲止。接著將相鄰的兩個像素取平均值並放入之間的像素中，例如$p'(0, 1) = (p'(0, 0) + p'(0, 2)) / 2$。最後剩餘的像素值可由$(p'(x_{j-1}, y_{j-1}) + p'(x_{j-1}, y_j) + p'(x_j, y_{j-1})) / 3$得知，例如$p'(1, 1) = (p'(0, 0) + p'(0, 1) + p'(1, 0)) / 3$，直到將所有的像素做完爲止，就可以得到一張解析度爲512 × 512的掩護影像。

2. 資料隱藏演算法之錯誤與缺失

首先將經過NMI內插放大得到的掩護影像以不重疊的方式將每四個像素當做一個區塊如圖9所示。其中p_0位置的像素都是固定不動的，以便於之後還原原始影像之用，其他p_1、p_2、p_3可用來嵌入秘密訊息。先計算出p_0與p_1、p_2、p_3各別的差值d_i，可個別得到嵌入的位元個數$n_i = \lfloor \log_2 |d_i| \rfloor$，接著依照嵌入的位元個數取出秘密訊息，並轉換成整數b_i，i=1…3，最後將p_1、p_2、p_3分別加上各別產生出的整數值b_i就可得到嵌入秘密訊息的像素值如圖1所示。直到將所有的區塊做完就能得到一張解析度爲512 × 512的僞裝影像。在取出秘密訊息的過程中不需要掩護影像，因爲掩護影像可由僞裝影像取得。假設僞裝影像像素爲$p''(x_j, y_j)$，$j = 0,\cdots,511$，掩護影像像素$p'(0,1)$可由$\lfloor (p''(0,0) + p''(0,2)) / 2 \rfloor$得知，而嵌入$p'(0,1)$的秘密訊息就會等於$p''(0,1) - p'(0,1)$。同樣地$p'(1,0)$可由$\lfloor (p''(0,0) + p''(2,0)) / 2 \rfloor$得知，而嵌入$p'(1, 0)$的秘密訊息就會等於$p''(1, 0) - p'(1, 0)$，最後$p'(1, 1)$可由$\lfloor (2 \times p''(0,0) + p''(0,2) / 2 + p''(2,0) / 2) / 3 \rfloor$得知，而嵌入$p'(1, 1)$的秘密訊息就會等於$p''(1, 1) - p'(1, 1)$如圖2所示。直到將每個區塊做完爲止。

Jung等人方法之訊息取出技術如圖2所示。其數學表示方式如式(1)所示，$p''(i, j)$爲僞裝影像之像素值，$x, y = 0, 1, \ldots 127$且$K = 2$。然而，第四項數學表示方式有錯誤，導致兩個缺失：（一）無法還原掩護影像；（二）無法正確取出秘密訊息。我們以下述之範例說明。

$$b = \begin{cases} p''(i,j) - \lfloor (p''(i,j) + p''(i,j))/K \rfloor & if\ i = K \cdot x,\ j = K \cdot y, \\ p''(i,j) - \lfloor (p''(i,j-1) + p''(i,j+1))/K \rfloor & if\ i = K \cdot x,\ j = K \cdot y + 1 \\ p''(i,j) - \lfloor (p''(i-1,j) + p''(i+1,j))/K \rfloor & if\ i = K \cdot x + 1,\ j = K \cdot y \\ p''(i,j) - \lfloor (K \cdot p''(i-1,j-1) + p''(i-1,j+1)/K + p''(i+1,j-1)/K)/(K+1) \rfloor & otherwise \end{cases} \quad (1)$$

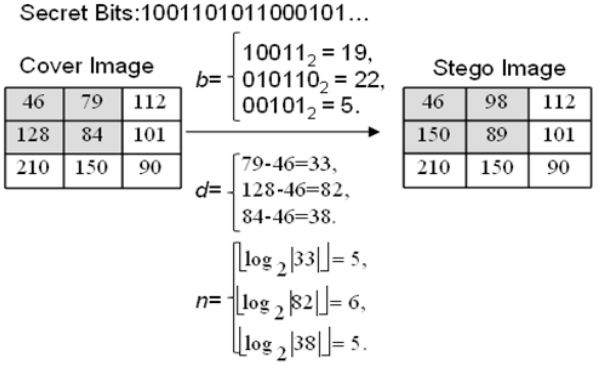


圖1　嵌入秘密訊息的過程

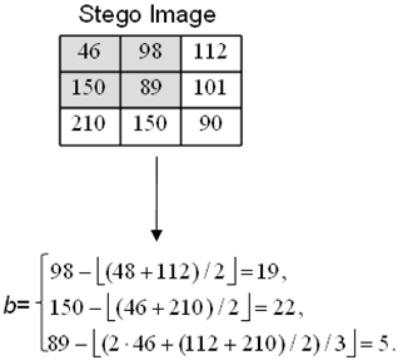


圖2　取出秘密訊息的過程

經過我們實作證實在某些情況下式(1)中的第四項數學式子將無法還原掩護影像導致無法將秘密訊息正確取出。以下我們舉一個實際的例子說明。假設掩護影像像素值$p'(0,0)$、$p'(0,1)$、$p'(1,0)$、$p'(1,1)$分別爲61、68、62、63，僞裝影像像素值$p''(0,0)$、$p''(0,2)$、$p''(2,0)$分別爲61、76、64。以下執行透過僞裝影像還原掩護影像的動作：

$p'(0,0) = 61$，$p'(0,1) = [(61 + 76) / 2] = 68$，
$p'(1,0) = [(61 + 64) / 2] = 62$，
$p'(1,1) = [(2 \times 61 + 76 / 2 + 64 / 2) / 3] = 64$

不等於原本掩護影像位於$p'(1,1)$的像素值63，由此可知無法由僞裝影像完全正確的得到掩護影像，然而取出秘密訊息也須仰賴求得的掩護影像，因此也無法正確無誤地將秘密訊息取出，如圖3所示。

另外一個實例。假設掩護影像像素值$p'(0,0)$、$p'(0,1)$、$p'(1,0)$、$p'(1,1)$分別爲57、79、67、67，僞裝影像像素值$p''(0,0)$、$p''(0,2)$、$p''(2,0)$分別爲57、102、78。以下執行透過僞裝影像還原掩護影像的動作：

$p'(0,0) = 57$，
$p'(0,1) = \lfloor (57+102)/2 \rfloor = 79$，
$p'(1,0) = \lfloor (57+78)/2 \rfloor = 67$，
$p'(1,1) = \lfloor (2\times 57+102/2+78/2)/3 \rfloor = 68$

不等於原本掩護影像位於$p'(1,1)$的像素值67，由此可知無法由僞裝影像完全正確的得到掩護影像，然而取出秘密訊息也須仰賴求得的掩護影像，因此也無法正確無誤地將秘密訊息取出，如圖4所示。

以上兩個實例可明顯看出位於$p''(1,1)$像素之秘密訊息皆無法正確取出，原因是 Jung等人的方法計算掩護影像方式爲

$$p'(1,1)=\lfloor (2\times p''(0,0)+p''(0,2)/2+p''(2,0)/2)/3 \rfloor$$

或是

$$p'(1,1)=\lfloor (2\times p''(0,0)+(p''(0,2)+p''(2,0)\ /2)/3 \rfloor$$

因此訊息是否能被正確的取出取決於$p''(0,0)$是奇數或是偶數。由範例可知當$p''(0,0)$爲奇數的時候就無法將秘密訊息完全正確取出。

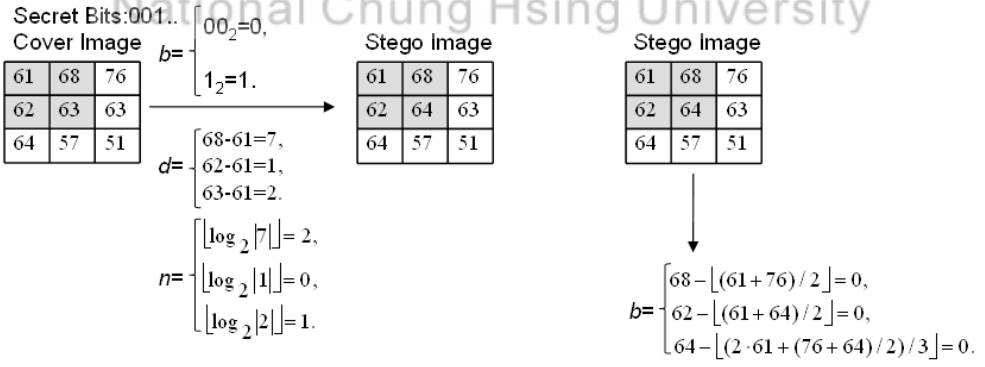


圖3 無法完全正確地取出秘密訊息的實例（一）[8]

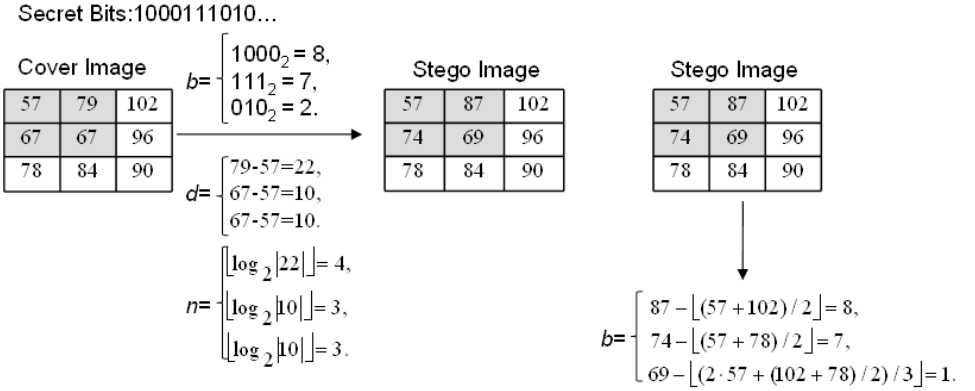


圖4 無法完全正確地取出秘密訊息的實例（二）[8]

2.1.2　修正後之資料隱藏演算法

我們改正Jung等人方法之錯誤數學式。修正式(1)中的第四項數學式子，正確之數學表示方式如式(2)所示。

針對$p'(1, 1)$像素而言，執行影像內插時產生的掩護影像像素

$p'(1, 1) = (p'(0, 0) + p'(0, 1)+ p'(1, 0)) / 3$，

其中

$p'(0, 1) = (p'(0, 0) + p'(0, 2)) / 2$，
$p'(1, 0) = (p'(0, 0) + p'(2, 0)) / 2$。

因此嵌入秘密訊息後還原位於掩護影像$p'(1, 1)$像素之像素值其式子應改為：

$$p'(1,1)=\left\lfloor (p''(0,0)+(p''(0,0)+p''(0,2))/2+(p''(0,0)+p''(2,0))/2)/3 \right\rfloor$$

就能完全正確將秘密訊息取出。

依照2.1節之例證，我們詳加說明改正後之數學式可完全解決上述無法還原掩護影像、與無法正確取出秘密訊息之缺失：以實例（一）而言從偽裝影像還原位於掩護影像$p'(1, 1)$像素之像素值為$p'(1,1)$=[(61 + (61 + 76) / 2 + (61 + 64) / 2) / 3] = 63等於原本掩護影像位於$p'(1,1)$的像素值63，所以秘密訊息可由偽裝影像值減去掩護影像值求得63-63=0，如圖5所示。以實例（二）而言從偽裝影像還原位於掩護影像$p'(1, 1)$像素之像素值為

$$p'(1,1)=\left\lfloor (57+(57+102)/2+(57+78)/2)/3 \right\rfloor=67$$

等於原本掩護影像位於$p'(1,1)$的像素值67，所以秘密訊息等於69 - 67 = 2，如圖6所示。故使用我們改正後的式子可以完全正確將秘密訊息取出。

2.2　資訊偽裝偵測（Steganalysis）

資訊偽裝偵測的目的在於找出可能有藏匿秘密訊息的數位媒體(例如：影像)。此技術又可分成兩種：（一）目標偵測（Targeted Steganalysis）[5]：針對特定的資料隱藏方法對偽裝媒體所造成的特徵進行分析；（二）盲目偵測（Blind Steganalysis）[9]：對任一種資料隱藏方法而言，只要收集適量的掩護媒體與偽裝媒體並進行有效之訓練。其所得出之特徵資料庫，即可用來偵測該種藏密法。雖然盲目偵測支援多種藏密技術偵測的能力，但是是其所需的影像資料庫需達到一定規模才能有預期的效果。且在資料隱藏技術不停演進與發展的現況下，要涵蓋空間域與時間域的盲目偵測機制之研發與維護，仍有其相當程程之難程。因此以下本文僅針對目標偵測進行說明。

$$b=p''(i,j)-\left\lfloor \begin{matrix}(p''(i-1,j-1)+(p''(i-1,j-1)+p''(i-1,j+1))/K+\\(p''(i-1,j-1)+p''(i+1,j-1)/K)/(K+1)\end{matrix} \right\rfloor \quad otherwis \qquad (2)$$

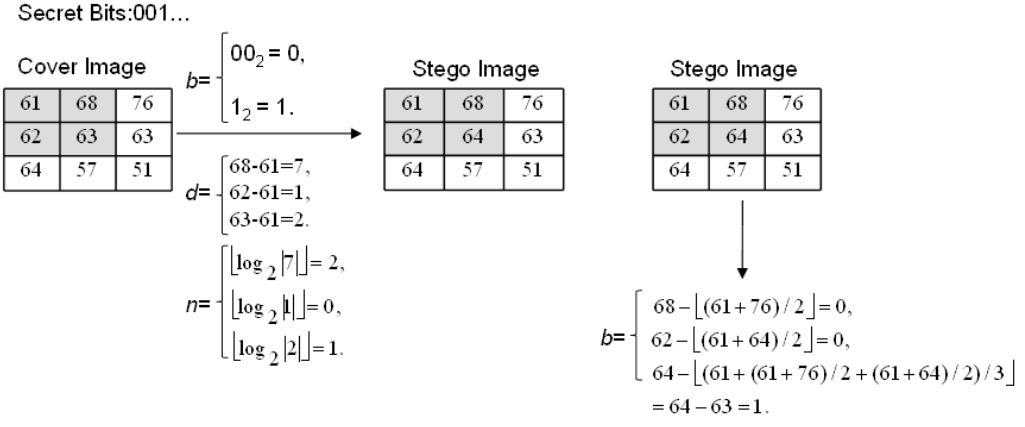


圖5　修正後可完全正確取出秘密訊息的實例（一）

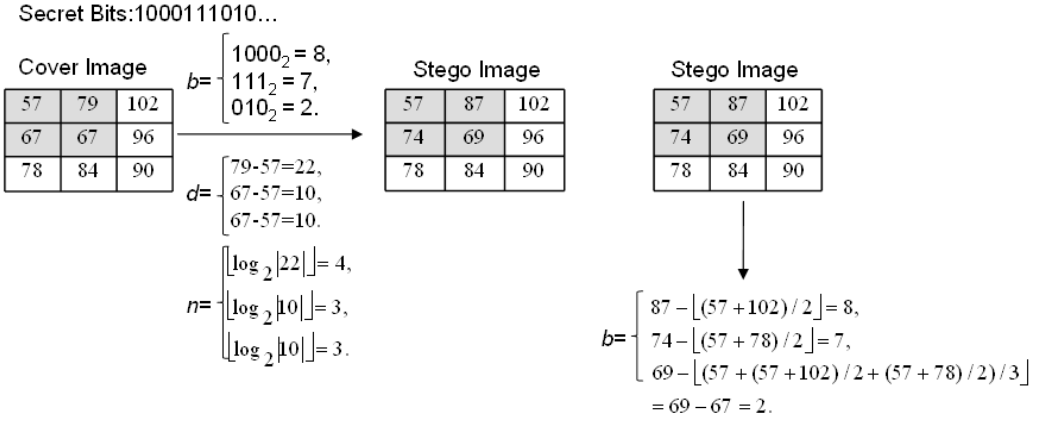


圖6　修正後可完全正確取出秘密訊息的實例（二）

其中較著名的資訊偽裝偵測技術—雙重統計分析法（Dual Statistics Method），又稱爲RS偵測[10]，是一個針對LSB藏密法具有相當威力的藏密分析法。在一般的情形下，即使僅代換掉每個像素的最後一個位元，此檢測法亦能有效偵測出來。其偵測的靈敏度可達每個像素嵌有0.005位元之秘密訊息量的偽裝媒體。此方法是將相鄰像素間區分多個群組G。先運用鑑別函數（Discrimination Function）來取得圖形的平順性（Smoothness），即像素群組的一般性（Regularity），透過遮罩（Mask）（$M \in \{-1,0,1\}$）及翻轉函數（Flipping Function）運算來將圖形區分爲Regular（R）、Singular（S）及Unusable（U）等三個群組；不同之遮罩及翻轉比率對此三個群組將造成不同影響，因此可藉此得出其藏密的機率。RS偵測技術對不同比例的翻轉偵測結果如圖7所示。

其中p爲藏匿秘密訊息之像素百分比。由於資料嵌入對圖形產生改變的機率只有1/2，所以當翻轉比率達到50%時，對圖形所造成之變化則將接近於全圖藏密的結果。透過F翻轉與F_{-1}翻轉可成功地偵測出圖形藏密的可能性。RS偵測法之運作方式如下：

(1) 先將欲偵測之偽裝影像根據遮罩大小分成n個相鄰像素組合成之群組$G=(x_1,x_2,\cdots,x_n)$。

(2) 使用鑑別函數f找出平順（Smooth）與雜亂（Noise）的像素群組，如式(3)所示：

$$f(x_1,x_2,...,x_n)=\sum_{i=1}^{n-1}\left|x_{i+1}-x_i\right| \qquad (3)$$

(3) 翻轉函數F與F_{-1}分別如式(4)、(5)所示：

$$F(x)=\begin{cases}(x-1), x \bmod 2=1\\(x+1), x \bmod 2=0\end{cases} \qquad (4)$$

$$F_{-1}(x)=F_1(x+1)-1 \qquad (5)$$

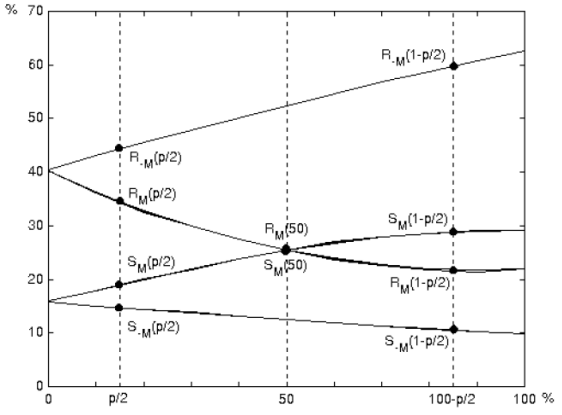


圖7　RS計算不同比率翻轉量之統計圖

(4) 利用翻轉函數遮罩M將圖形區分爲Regular（R）、Singular（S）及Unusable（U）等三個群組。如式(6)所示：

$$\begin{aligned} &R: \mathrm{f(F(G))>f(G)};\\ &S: \mathrm{f(F(G))>f(G)};\\ &U: \mathrm{f(F(G))>f(G)}; \end{aligned} \tag{6}$$

(5) 當影像未嵌入資料時，Fridrich已證明式（7）會成立，若不成立則影像可能有藏匿機密資訊：

$$R_m \cong R_{-m} \ and \ S_m \cong S_{-m} \tag{7}$$

三、我們提出的方法

我們提出的方法其演算法流程如圖8所示。主要分成三個階段：第一個階段爲產生掩護影像；第二個階段爲嵌入秘密訊息；第三個階段爲取出秘密訊息，以及還原原始影像。

演算法步驟如下：

Step 1：將輸入影像每四個像素爲一組並只選取左上角像素構成原始影像。

Step 2：將原始影像之每個像素座標乘2並複製到欲放大的影像中。

Step 3：像素與像素之間的空像素值取鄰近像素平均值處理。

Step 4：剩餘像素用我們提出的內插法處理，詳細細節請參考3.1節。

Step 5：以3×3之遮罩以不重疊的方式只對中間像素作高斯模糊處理，就能產生掩護影像。

Step 6：將掩護影像每四個像素爲一組，左上角像素值固定不動，右上角像素使用秘鑰打亂嵌入順序並使用混合式OPAP嵌入演算法，其餘兩個像素使用LSB-matching嵌入秘密訊息。

Step 7：整張掩護影像皆嵌入完畢，就能產生一張僞裝影像。

Step 8：將僞裝影像每四個像素爲一組，取出左上角像素就能還原原始影像。

Step 9：將僞裝影像每四個像素爲一組，使用秘鑰依序取出右上角像素之秘密訊息。其餘兩個像素也一並取出各別的LSB就能完全取出秘密訊息。

3.1 產生掩護影像

我們產生掩護影像的三個步驟分別爲縮小、內插、高斯模糊，每個步驟的細節如下所述。

Step 1：縮小

我們首先將大小爲512 × 512解析度之輸入影像縮小爲僅具有256 × 256解析度之原始影像，以方便能和 [8]比較。假設輸入影像的像素爲$p(x_i, y_i)$，$i = 0,\cdots,511$，縮小之後得到的原始影像的像素爲$p'(x_j,$

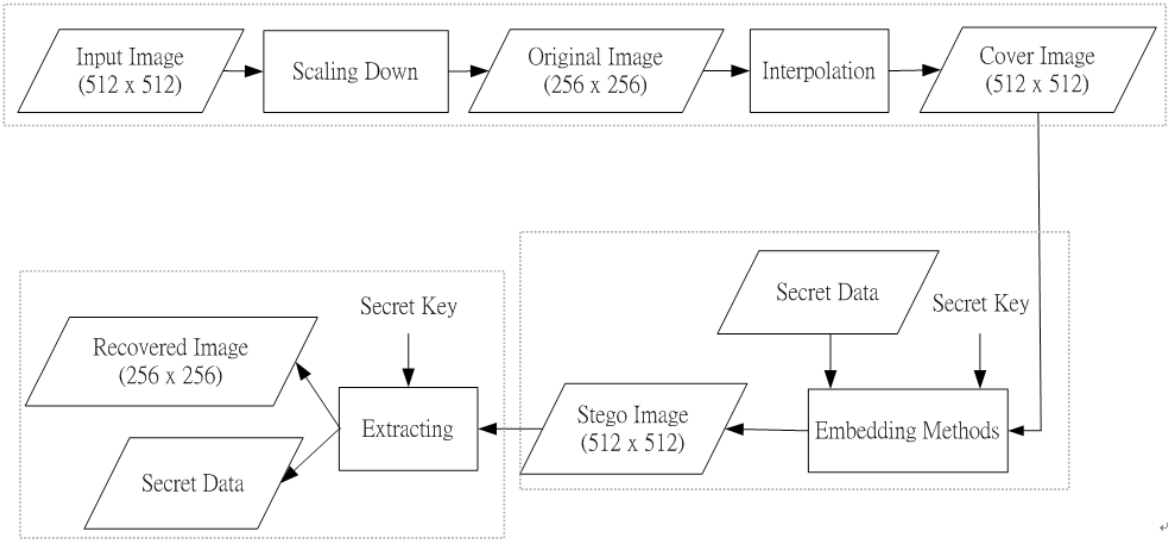


圖8　演算法流程圖

$y_j)$，$j = 0,\cdots,255$ ，其縮小的方式如式（8）：

$$p'(x_j, y_j) = p(x_i, y_i), \quad \text{if } x_i \ mod\ 2 = 0 \text{ and } y_i \ mod\ 2 = 0 \tag{8}$$

由式(8)可知，只要輸入影像的像素座標能夠同時被2整除，就將輸入影像的該像素值直接複製到原始影像的像素中，直到將所有原始影像的像素填滿為止，我們舉一個例子實際說明，如圖 9 所示。

Step 2：內插

我們在此步驟將具有256 × 256解析度之原始影像內插成為具有512 × 512的放大影像。為了能無失真的還原原始影像，我們使用三個處理方式來達成內插之目的，分別為像素複製、鄰近像素平均處理、內插處理。

1. 像素複製

透過式（9）將原始影像的像素值全部複製到放大之後的影像中，假設原始影像的像素為$p'(x_j, y_j)$，$j = 0,\cdots, 255$，放大之後影像的像素為$p''(x_i, y_i)$，$i = 0,\cdots, 511$。

$$p''(x_i, y_i) = p'(x_j, y_j) \quad x_i = 2 * x_j \ 、\ y_i = 2 * y_j \tag{9}$$

由上面的式子可知，就是將原始影像的像素值直接複製到像素座標為原始影像像素座標二倍的放大影像中，例如原始影像像素座標為$p'(0, 0)$就直接將該像素值複製到$p''(0, 0)$中，同樣地，當原始影像像素座標為$p'(0, 1)$就直接將該像素值複製到$p''(0, 2)$中，以此類推直到將原始影像像素值全部複製到放大的影像中為止。

2. 鄰近像素平均處理

將相鄰的兩個像素取平均值並放入之間的像素中。

3. 內插處理

最後剩餘的像素我們以其他的方式來處理，我們實作的內插方式有Nonlinear Interpolation [11]、Bilinear Interpolation [13]、Neighbor Mean Interpolation(NMI) [8]、簡單版GAP [12] (SGAP, Simple Gradient-Adjusted Predictor)、Reversible Data-Embedding [14]。以下我們對這些內插方法做個簡單的說明。Nonlinear Interpolation是透過3×3像素遮罩以重疊的方式處理，每處理一次就會內插出四個像素，而我們只挑選其中一個像素來填補剩餘的像素；Bilinear Interpolation是指經放大後的像素顏色為原影像中與其最臨近的四個像素（2×2），依其距離施以不同比重運算而得的結果；NMI之前已提過是以像素的平均值來做影像內插的動作；SGAP是一種利用預估像素周圍的四個像素產生預估值的低複雜度預估器、Reversible Data-Embedding此方法包含了Edge Directed Prediction (EDP) [15]、Median Edge Detector (MED) [16]這兩種技術。另外我們將前述的方法互相混合像是：EDP搭配Nonlinear、EDP搭配Bilinear、EDP搭配NMI、EDP搭配MED、EDP搭配SGAP，以上搭配方式都是以EDP優先，如果預測結果溢位或是無法預測時，則採用後者。

前述的內插方法中以EDP搭配SGAP所產生出來的影像品質最好，我們就針對這個方法作詳細的說明。在做EDP的時候必須先取得欲預測像素之鄰近四個像素值，並使用式（10）儲存鄰近四個像素

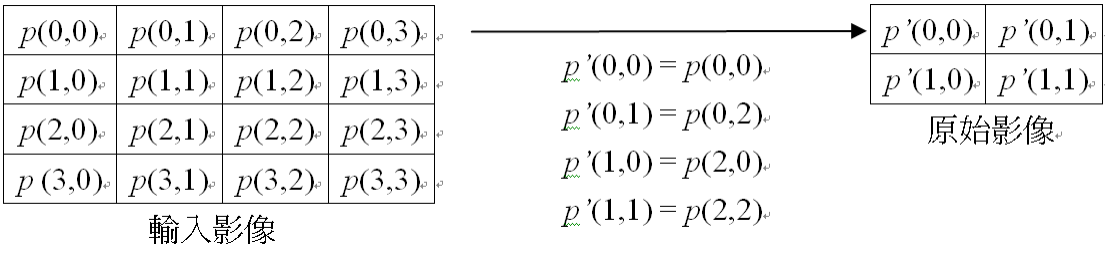


圖9 輸入影像Scaling down的例子

值，接著再分別找出此四個像素之鄰近二個像素的像素值，一共有八個像素，並用式（11）儲存這八個像素值，接著由式（12）求得預測係數 $\vec{a}$ ，最後由式（13）求得預測值 $\hat{Y}(n)$ ，舉個例子如圖10所示。

$$\vec{Y}_{4x1} = [Y(1)\,Y(2)\,Y(3)\,Y(4)]^T \tag{10}$$

$$C_{4x2} = \begin{bmatrix} Y(1)_{left} & Y(1)_{upper} \\ Y(2)_{left} & Y(2)_{upper} \\ Y(3)_{left} & Y(3)_{upper} \\ Y(4)_{left} & Y(4)_{upper} \end{bmatrix} \tag{11}$$

$$\vec{a} = \left(C^T C\right)^{-1}\left(C^T \vec{Y}\right) = \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} \tag{12}$$

$$\hat{Y}(n) = \left\lceil a(1)\,Y(1) + a(2)\,Y(2) \right\rceil \tag{13}$$

在做SGAP的時候必須先取得欲預測像素之鄰近四個像素，我們分別以符號 a、b、c、d 來表示，接著在透過式（14）來求得預測值 $\hat{P}$，如圖11所示。舉個例子假設 a、b、c、d 的像素值分別爲50、51、51、52，透過式（14）可得知預測像素值 $\hat{P} = \frac{51+52}{2} + \frac{51-50}{4} = 51$。

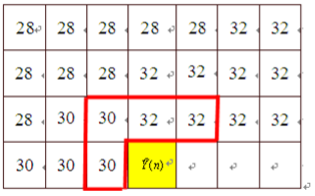
$$\hat{p} = \frac{b+d}{2} + \frac{c-a}{4} \tag{14}$$

Step 3：高斯模糊

爲了使我們預估出來的像素能夠與其鄰近之像素更相近以便取得較佳影像品質，我們針對預估的像素作高斯模糊，使其與鄰近像素更均勻化。我們以一個大小爲3 × 3的遮罩，作爲高斯模糊的遮罩如圖12所示，針對位於遮罩中心點之像素並以不重疊的方式在放大的影像上處理。將各像素值與遮罩權重的乘積做累加再除以權重總數就可以得到新的像素值，將得到的新像素值直接取代位於遮罩中心點下的像素，做完之後就可以得到掩護影像了。

3.2　嵌入秘密訊息

首先我們將掩護影像分成每四個像素爲一組如圖13所示。其中 p_0 位置的像素都是固定不動的，以便於之後還原原始影像之用，其他像素 p_1、p_2、p_3 可用來嵌入秘密訊息。爲了方便與 [8]比較，像素 p_1 我們使用目前最好的嵌入技術OPAP [3]並將其混合來嵌入秘密訊息；爲了降低嵌入秘密訊息之後僞裝影像失真度，我



$$\vec{Y}_{4x1} = [Y(1)\,Y(2)\,Y(3)\,Y(4)]^T = [30\;32\;30\;32]^T$$

$$C_{4x2} = \begin{bmatrix} Y(1)_{left} & Y(1)_{upper} \\ Y(2)_{left} & Y(2)_{upper} \\ Y(3)_{left} & Y(3)_{upper} \\ Y(4)_{left} & Y(4)_{upper} \end{bmatrix} = \begin{bmatrix} 30 & 30 \\ 30 & 32 \\ 30 & 28 \\ 32 & 32 \end{bmatrix}$$

$$C_{2\times4}^T C_{4x2} = \begin{bmatrix} 3724 & 3724 \\ 3724 & 3732 \end{bmatrix}_{2\times2}$$

$$C_{2\times4}^T \vec{Y}_{4\times1} = \begin{bmatrix} 3784 \\ 3788 \end{bmatrix}_{2\times1}$$

$$\vec{a} = \left(C^T C\right)^{-1}\left(C^T \vec{Y}\right) = \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 0.5161 \\ 0.5000 \end{bmatrix}$$

$$\hat{Y}(n) = \left\lceil a(1)\,Y(1) + a(2)\,Y(2) \right\rceil = \left\lceil 0.5161 * 30 + 0.5 * 32 \right\rceil = 32$$

圖10　EDP的例子

a	b	c
d	$\hat{P}$	

圖11　欲預測像素之鄰近四個像素

1	2	1
2	4	2
1	2	1

圖12　高斯模糊的遮罩

p_0	p_1
p_2	p_3

圖13　將掩護影像分成每四個像素爲一組

們將像素p_2、p_3以LSB-matchin [17]的方式嵌入秘密訊息。OPAP就是將LSB處理後的像素值$\pm 2^k$並選擇與原本像素值相差最小的，其中k指的是每個像素嵌入的位元個數。在我們的方法中我們使用混合式OPAP來嵌入秘密訊息，將1-OPAP~5-OPAP彼此相互混合。以2-OPAP爲例，假設秘密訊息爲(00) 2，p_1像素值爲35，先以LSB嵌入方式可產生像素值32，再經過OPAP處理之後可以產生像素值36。LSB-matching的概念是兩兩像素爲一組，每組最多只需更動一個位元就能嵌入兩個位元。舉例來說，假設秘密訊息爲(00)2，p_2、p_3像素值分別爲35(00100011) 2、34(00100010) 2，經過LSB-matching處理後p_2、p_3分別爲36(00100100) 2、34(00100010) 2。

我們針對每個2×2大小的區塊做了每個像素平均期望變動量的預估。以便於驗證使用混合式OPAP搭配LSB-matching嵌入秘密訊息後所得到之僞裝影像品質。首先我們先針對OPAP與LSB-matching做平均每個像素期望均方誤差(EMSE, Expected Mean Square Error)分析。OPAP方面，由於k-OPAP的變動範圍爲$-2^{k-1},..., 2^{k-1}-1$，因此平均每個像素期望均方誤差我們以*OPAP_EMSE*表示如式(15)所示。LSB-matching方面，由於LSB-matching的概念是每兩個像素爲一組，每組最多只需更動一個位元就能嵌入兩個位元。其中，更動一個像素的一個位元可藏入2個位元的機率是0.75，不用更動任何像素可藏入2個位元的機率是0.25，因此兩個像素p_2、p_3若使使用LSB-matching嵌入2個位元，兩像素之期望均方誤差共爲0.75，我們以*LM_EMSE*表示。最後，以一個區塊爲單位算出每個像素平均期望變動量並代入訊號對雜訊比的公式裡就能得到預測的PSNR值。

$$OPAP\_EMSE = ((-2^{k-1})^2 + (-2^{k-2})^2 + ... + (2^{k-1}-1)^2)/2^k \quad (15)$$

以像素p_1嵌入2-OPAP，像素p_2、p_3使用LSB-matching嵌入爲例，由於2-OPAP的變動的範圍是從-2, -1, 0, +1，因此平均每個像素期望均方誤差爲$((-2)^2+(-1)^2+0+1^2)/2^2=1.5$，像素$p_2$、$p_3$的期望均方誤差共爲0.75，$p_0$像素沒有變動，我們可算出一個區塊平均每個像素的變動量爲(1.5 + 0.75) / 4 = 0.5625，再代入訊號對雜訊比的公式中就可得到預測的PSNR值

$$PSNR=10\log_{10}\left(\frac{255^2}{0.5625}\right)\cong 50.63 \text{ dB}$$

我們將1-OPAP～5-OPAP的平均每個像素期望均方誤差*OPAP_EMSE*數據列在表1中。另外，我們將每個區塊的嵌入量還有一個區塊平均每個像素的變動量以及預測的PSNR值數據列在表2中。在表格中最上面一列的區塊嵌入量指的是p_0位置的像素固定不動之外，其他像素p_1、p_2、p_3嵌入量的總合，我們每個區塊的嵌入量爲3-7個位元。透過期望變動量的預測我們可以明確的掌握住嵌入秘密訊息時所需採用的OPAP混合方式以及嵌入秘密訊息之後所得到的僞裝影像品質的範圍。混合式OPAP搭配LSB-matching之混合方式與詳細實驗數據我們呈現在實驗結果的章節中。

表1 OPAP平均每個像素期望均方誤差的數據

嵌入方式	*OPAP_EMSE*
1-OPAP	0.5
2-OPAP	1.5
3-OPAP	5.5
4-OPAP	21.5
5-OPAP	85.5

表2 期望變動量與PSNR值的預估

每個區塊的嵌入量（位元）	3	4	5	6	7
p_0 | p_1(OPAP) p_2、p_3(LSB-matching) 每個像素嵌入量（位元）	0 | 1 2	0 | 2 2	0 | 3 2	0 | 4 2	0 | 5 2
每個像素期望均方誤差	0 | 0.5 0.75	0 | 1.5 0.75	0 | 5.5 0.75	0 | 21.5 0.75	0 | 85.5 0.75
平均每個像素的期望變動量	0.3125	0.5625	1.5625	5.5625	21.5625
預估的PSNR值（dB）	53.18	50.63	46.19	40.68	34.79

我們的方法所嵌入的秘密訊息皆是以亂數產生的，爲了保護秘密訊息不易被取得，我們將Secret key當成是亂數產生器（random number generator）的種子（seed），產生一連串的隨機亂數，再配合嵌入演算法來嵌入資訊以增加安全性。

3.3　取出秘密訊息，以及還原原始影像

取出秘密訊息時，使用相同的Secret Key當嵌入是用OPAP時就直接取出像素值的LSB即可，如果嵌入是用LSB-matching時取出方式如下，假設僞裝影像的像素分別爲x_i，x_{i+1}，嵌入的秘密訊息爲$(m_1, m_2)_2$，取出時$m_1 = x_i \bmod 2$、$m_2 = (\lfloor x_i / 2 \rfloor + x_{i+1}) \bmod 2$，例如假設僞裝影像的像素分別爲36、34，嵌入的秘密訊息爲(00)2，令m_1、m_2爲取出秘密訊息的變數，取出時$m_1 = 36 \bmod 2 = 0$，$m_2 = (\lfloor 36/2 \rfloor + 34) \bmod 2 = 0$，還原原始影像時，只要僞裝影像的像素座標能同時被2整除並取出其像素值就可還原原始影像，也就是還原到縮小之後的那張圖。有別於 [8]，我們的方法不需要還原掩護影像就能完全將秘密訊息取出，但我們的方法也可還原掩護影像，只要將還原的原始影像再做影像內插即可。前面已經提過Jung他們的方法在取出秘密訊息的時候有瑕疵無法100%還原掩護影像，也連帶影響取出秘密訊息的正確性。

總而言之，我們的演算法再取出秘密訊息時不須透過掩護影像，因此在取出秘密訊息時完全不會發生錯誤。

四、實驗結果

本研究我們是使用Visual C++ 6.0版本的程式語言來實作提出的方法，實作環境爲Intel (R) Pentium Processor 1.6 GHz，1GB記憶體，測試影像來源爲University of Southern California (USC) Signal & Image Processing Institute Image Database，此實驗用的六張測試影像都是大小512 × 512的灰階影像，用來當作輸入影像，如圖 14 所示。測試結果分成三個部分：(1) 影像內插結果及執行時間、(2) 我們的方法與Jung等人的方法之比較。我們分別在4.1、4.2節呈現這些實驗結果。

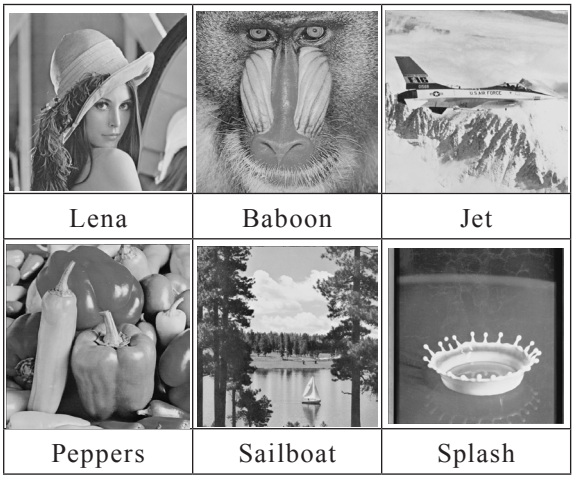


圖14　輸入影像（512×512）

4.1　影像內插結果及執行時間比較

影像作內插放大之後的品質是看Input image和Cover image之間的PSNR值，將我們的方法（M_1 -M_8）與Jung和Yoo提出的方法以及其他傳統式的方法例如：最鄰近點內插（Nearest Neighbor Interpolation, NNI）、雙線性內插（Bilinear Interpolation, BI）比較。如表3所示。由於Jung等人所提出的方法並無實作Sailboat以及Splash影像，因此爲了達到比較之公平性我們以Lena、Baboon、Jet、Peppers四張影像來說明。以我們的方法M_2爲例其所產生出最低內插影像品質與最高內插影像品質分別爲Baboon影像與Lena影像分別具有22.90 dB、32.97 dB。平均每張影像內插品質約爲29.25 dB。然而，Jung等人之影像內插方法所產生出最低內插影像品質與最高內插影像品質分別爲Baboon影像與Lena影像分別具有20.09 dB、26.82 dB。平均每張影像內插品質約爲24.39 dB。NNI所產生出最低內插影像品質與最高內插影像品質分別爲Baboon影像與Lena影像分別具有16.91 dB、22.56 dB。平均每張影像內插品質約爲20.68 dB。BI所產生出最低內插影像品質與最高內插影像品質分別爲Baboon影像與Lena影像分別具有19.59 dB、26.24 dB。平均每張影像內插品質約爲23.89 dB。實驗結果顯示：我們的方法影像內插後的品質平均比Jung等人他們的方法高出4.86 dB；也分別比NNI、BI高出8.39 dB、5.36 dB。

此外原本Jung等人所提出來的結果 [8]並沒有實作Sailboat、Splash這兩張圖，在這邊我們有實作他們提出的演算法。對於Splash這張圖而言，Jung和Yoo提出的方法只能產生32.18 dB的僞裝影像品

表3 影像內插方法PSNR值的比較（單位：dB）

方法＼影像	Lena	Baboon	Jet	Peppers	Sailboat	Splash
M_1（使用EDP + SGAP內插）	32.87	22.89	30.64	30.28	28.51	33.90
M_2（使用EDP + MED內插）	32.97	22.90	30.96	30.16	28.42	33.64
M_3（使用Nonlinear內插）	31.48	22.87	29.86	29.74	27.79	33.03
M_4（使用Bilinear內插）	32.85	22.91	30.61	30.30	28.55	33.71
M_5（使用NMI內插）	33.02	22.90	30.66	30.26	28.44	33.74
M_6（使用EDP + Nonlinear內插）	32.28	22.86	30.31	30.06	28.25	33.42
M_7（使用EDP + Bilinear內插）	32.83	22.89	30.63	30.31	28.56	33.72
M_8（使用EDP + NMI內插）	32.99	22.88	30.74	30.23	28.48	33.76
Nearest Neighbor Interpolation（NNI）	22.56	16.91	21.03	22.22	24.60	29.10
Bilinear Interpolation（BI）	26.24	19.59	24.24	25.47	25.71	32.58
Jung and Yoo的方法	26.82	20.09	24.66	25.99	27.43	32.18

質，而傳統式的內插方法Bilinear(BI)卻能產生出32.58 dB之僞裝影像品質，較Jung等人方法高出0.4 dB。然而，我們的方法（M_1-M_8）所產生出的僞裝影像品質皆優於Jung等人方法以及傳統式內插方法。其中以Lena、Jet、Peppers影像改進最多，其餘三張影像Baboon、Sailboat、Splash改進後差距不大的原因在於影像本身之特徵不同所致。因此執行影像內插時原影像之特徵也會影響到內插之後的影像品質。

最後我們提出的方法所產生的掩護影像如圖15所示。

我們將使用這些掩護影像來嵌入秘密訊息。

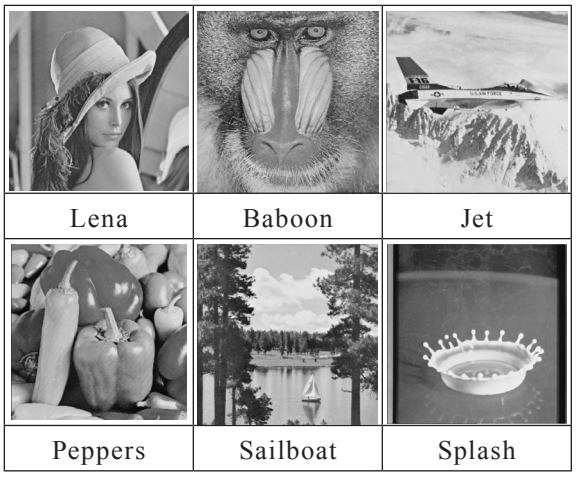


圖15 掩護影像 （512 × 512）

理論上我們提出的影像內插方法與Jung等人內插方法複雜度皆爲$O(n^2)$。實際上我們也列表比較提出之影像內插方法M_1與實作Jung等人方法的實際執行時間，如表4所示。結果顯示：每張影像內插實際執行時間與Jung等人之差距皆小於0.15秒。執行時間會較 [Jung 2009] 略久的原因是因爲本內插法需要計算逆矩陣以及執行高斯模糊等運算。雖然如此之間的差距仍微乎其微。

4.2 我們與Jung等人的方法之比較

我們將與Jung等人所提之方法分成兩方面做比較：一、嵌入量與Jung等人的方法相近時比較其僞裝影像品質；二、僞裝影像品質與Jung等人的方法相近時比較其嵌入量。

我們嵌入秘密資訊時是以一個區塊爲單位，每個區塊大小爲2 × 2如圖12所示。其中位於p_0的像素我們不去更動它，其他的像素p_1、p_2、p_3用來嵌入秘密訊息。我們將p_2、p_3的像素使用LSB-matching的方法嵌入秘密資訊，這樣每個區塊可以嵌入2個

表4 影像內插實際執行時間比較

影像	EDP + SGAP	[Jung 2009]
Lena	0.109秒	0.031秒
Baboon	0.094秒	0.015秒
Jet	0.156秒	0.016秒
Peppers	0.109秒	0.016秒

位元。在表4與表5中我們以LM Capacity表示LSB-matching之藏量。當p_2、p_3嵌完之後，我們在整張掩護影像中，對部分p_1像素以OPAP技巧，嵌入不等量之秘密訊息，我們稱之OPAP混合技巧。我們作法是針對p_1像素做期望變動量的預估，然後使用預估出來的結果算出OPAP混合比例。我們得以與Jung等人的方法所產生之偽裝影像比較其影像品質以及嵌入量。我們將分別在4.2.1以及4.2.2小節中詳細說明。

4.2.1　偽裝影像品質相近時之嵌入量比較

我們將實驗分成二個程序：一、預估程序；二、實作程序。以驗證我們所使用的預估方法之正確性以及精準度。

一、預估程序：首先，我們將我們擬欲產生之偽裝影像品質調整成略高於Jung等人的方法影像品質。在此情況下，我們希望利用分析的方式，能證明在具有相似的偽裝影像品質下，使用我們的方法，得到高於Jung等人方法的藏量。爲了方便比較，我們將偽裝影像品質設定爲5%略高於Jung等人之影像品質，以求比較之公允性。首先，我們以M_{pre}代表Jung等人方法單一像素之均方誤差。以Lena爲例證，Jung等人的方法之PSNR爲41.46 dB，我們加上0.05%之比例後，其PSNR變爲41.48，此也代表Jung等人的方法之單一像素均方誤差（M_{pre}）爲4.6239。據此，我們可以預估使用OPAP混合技巧之均方誤差如式（16）所示。

$$M_{OPAP} = (M_{pre} \times 4) - M_{LM} \qquad (16)$$

其中，M_{LM}代表使用LSB-matching之期望變動量。由於LSB-matching是每兩個像素爲一組，更動其中一個像素的一個位元可藏入2個位元的機率是0.75，不用更動任何像素可藏入2個位元的機率是0.25，因此兩個像素p_2、p_3若使使用LSB-matching嵌入2個位元，兩像素之期望變動量共爲0.75，因此，M_{LM} = 0.75。因此，我們可以根據式（16）算出M_{OPAP}=(4.6239×4) - 0.75 ≅ 17.75。

由於以OPAP技巧，嵌入3個位元之期望變動量爲5.5；若嵌入4個位元之期望變動量爲21.5。由於預估之期望變動量爲17.75，因此，使用OPAP混合技巧必須搭配3個位元，亦即3-OPAP與4位元之OPAP嵌入方式，亦即4-OPAP。接著，我們說明如何預估OPAP混合的比例。

當M_{OPAP}預估是落在3-OPAP與4-OPAP的期望

表5　在影像品質相近時本方法之預估程序與實作程序之比較結果

影像	欲達到之影像品質PSNR（dB）	p_1混合式OPAP比例預估（%）		實作產生影像品質PSNR（dB）	實作與預估之PSNR誤差	p_1 OPAP Capacity（位元）	p_2、p_3 LM Capacity（位元）	總嵌入量（位元）
Lena	41.48	3-OP 23.47	4-OP 76.53	41.4805	0.0005	246,763	131,072	377,835
Baboon	35.48	4-OP 19.64	5-OP 80.36	35.4753	0.0047	314,809	131,072	445,881
Jet	39.83	4-OP 92.50	5-OP 7.50	39.8289	0.0011	267,059	131,072	398,131
Peppers	41.95	3-OP 35.33	4-OP 64.67	41.9498	0.0002	238,990	131,072	370,062
Sailboat	37.73	4-OP 66.20	5-OP 33.80	37.7288	0.0012	284,295	131,072	415,367
Splash	42.47	3-OP 47.04	4-OP 52.96	42.4711	0.0011	231,316	131,072	362,388

變動量範圍內，以Lena影像爲例，預估OPAP混合比例的方程式可寫成M_{OPAP}= X(5.5) + (1-X)(21.5)，其中X表示3-OPAP在像素p_1中比例爲X，4-OPAP在像素p_1中比例爲(1-X)。根據上述，我們可以算出X=0.2347，所以3-OPAP約佔23.47%，4-OPAP約佔76.53%。顯然的，使用不同的掩護影像，所需搭配之OPAP混合技巧與所預估之比例也會相異。

二、實作程序：我們依照預估所產生之OPAP混合比例，依照混合比例並以一個順序密鑰決定掩護影像內p_1像素嵌入秘密訊息之順序。當掩護影像p_1像素皆按照嵌入順序完成單一像素嵌入程序後即完成此小結所提出之嵌入演算法。爲了達到公允性，我們將每張影像各執行演算法1000次，並取其平均值作爲比較值。

表5顯示我們的OPAP混合技巧以及預估之比例與實作之比例之間的比較。在表格中我們以符號OP表示OPAP。例如3-OPAP，就表示成3-OP。以Sailboat影像爲例，Jung等人的方法之PSNR爲37.71 dB，我們加上0.05%之比例後，其PSNR變爲37.73，此也代表Jung等人的方法之單一像素均方誤差（M_{pre}）爲10.9697。據此，透過式（9）預估使用OPAP混合技巧之均方誤差。由於以OPAP技巧，嵌入4個位元之期望變動量爲21.5；若嵌入5個位元之期望變動量爲85.5。當預估之期望變動量爲M_{OPAP}=(10.9697×4)-0.75⌋43.13，因此，使用4-OPAP搭配5-OPAP之混合技巧嵌入秘密訊息。預估OPAP混合比例的方程式可寫成M_{OPAP}=X(21.5) + (1-X)(85.5)，假設X是4-OPAP混合的比例，則(1-X)就是5-OPAP混合的比例，兩者混合比例的方程式可寫成43.13 = X(21.5) + (1-X)(85.5)，X ≅ 0.6620，所以4-OPAP約佔66.20%，5-OPAP約佔33.80%。因此總共產生的嵌入量爲131072 + 65536 × (4 × 0.662 + 5 × 0.338) ≅ 415,367個位元。實作時，我們依照預估出來的比例實際去嵌入秘密訊息。爲了驗證我們預估演算法之正確性及PSNR精準性，我們每張影像各執行1000次取其平均值。接著將平均所產生的PSNR值與預估時所能達到的PSNR值比對。實作結果顯示以Sailboat影像而言4-OPAP約佔66.20%，5-OPAP約佔33.80%，再分別嵌入4-OPAP及5-OPAP之秘密訊息。因此使用LSB-matching嵌入之秘密訊息個數（LM）再加上混合式OPAP嵌入之秘密訊息個數一共可以產生131072 + 65536 × (4 × 0.662 + 5 × 0.338) ≅ 415,367個位元。產生之僞裝影像品質平均爲37.7288 dB。預估結果與實作結果僞裝影像品質PSNR誤差僅有0.0012 dB。

整體實驗結果顯示：僞裝影像品質誤差最小的是Peppers影像誤差約爲0.0002 dB；誤差最大的是Baboon影像誤差約爲0.0047 dB。平均每張影像之僞裝影像品質誤差約爲0.0015 dB。據此，預估結果與實作結果極爲相似，表示我們所預估的比例以及使用的混合技巧相當的精準。

表6顯示本方法實作的結果與Jung等人所提方法之比較。由於原始Jung等人文獻沒有Sailboat與Splash影像之實驗數據，因此我們以

表6　在影像品質相近時本方法與Jung and Yoo的方法其嵌入量的比較結果

影像	我們的方法-實作程序		Jung and Yoo的方法		PSNR多出0.05%之下嵌入量提昇率（%）
	Capacity（位元）	PSNR（dB）	Capacity（位元）	PSNR（dB）	
Lena	377,835	41.4805	200,868	41.46	88.10
Baboon	445,881	35.4753	425,199	35.46	4.86
Jet	398,131	39.8289	182,546	39.81	118.10
Peppers	370,062	41.9498	195,500	41.93	89.30
Sailboat	415,367	37.7288	277,995	37.71	49.42
Splash	362,388	42.4711	132,843	42.45	172.80

Lena、Baboon、Jet、Peppers影像來比較。其中Baboon影像與Jet影像對Jung等人的方法產生的偽裝影像的嵌入量分別具有最小與最大的差別。就Baboon而言，使用我們的方法可以嵌入445,881個位元，具有偽裝影像品質35.4753 dB。然而，若使用Jung等人的方法，僅能嵌入425,199個位元，其掩護影像品質已經降低至35.46 dB。我們的方法相較於Jung之方法，可多嵌入約略4.86%之秘密訊息，得到相近之偽裝影像品質。就Jet而言，使用我們的方法可以嵌入398,131個位元，具有偽裝影像品質39.8289 dB。然而，若使用Jung等人的方法，僅能嵌入182,546個位元，其掩護影像品質已經降低至39.81 dB。我們的方法相較於Jung之方法，可多嵌入約略118.10%之秘密訊息，得到相近之偽裝影像品質。整體之實驗結果顯示：在偽裝影像品質相近之下我們方法得到的藏量比Jung他們多4.86%-118.10%。

以下是偽裝影像品質相近時，我們方法所產生的偽裝影像，如圖16所示。

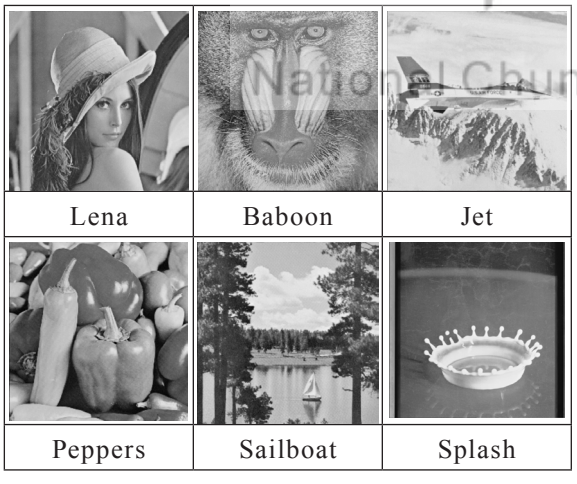


圖16　偽裝影像品質相近時，我們方法所產生的偽裝影像

4.2.2　嵌入量相近時之偽裝影像品質比較

我們將實驗分成二個程序：（一）預估程序；（二）實作程序。以驗證我們所使用的預估方法之正確性以及精準度。

（一）預估程序：首先，我們將我們擬欲產生之秘密訊息嵌入量調整成略高於Jung等人的方法所產生的秘密訊息嵌入量。在此情況下，我們希望利用分析的方式，能證明在具有相似的嵌入量下，使用我們的方法，得到高於Jung等人方法的偽裝影像品質。為了達到比較之公允性，我們將秘密訊息嵌入量設定為0.05%略高於Jung等人之秘密訊息嵌入量。我們以C_{pre}代表其方法之嵌入量。以Lena為例證，Jung等人的方法之嵌入量為200,868個位元，我們加上0.05%之比例後，其嵌入量為200,968個位元，此也代表Jung等人的方法之嵌入量（C_{pre}）為200,968個位元。據此，我們可以預估使用OPAP混合技巧之像素p_1單位像素嵌入量F如式（17）所示。

$$F = \left(C_{pre} - 131072\right)/65536 \qquad (17)$$

其中，F代表單位像素嵌入量，其值範圍為$0<F\le 5$，用以決定OPAP之混合方式。接續上例，F=(200968－131072)/65536=1.0665。因此，使用OPAP混合技巧必須搭配1個位元，亦即1-OPAP與2個位元之OPAP嵌入方式，亦即2-OPAP。接著，我們說明如何預估OPAP混合的比例。

當$1 \le F \le 2$，就以1-OPAP與2-OPAP混合技巧嵌入秘密訊息，則以Lena影像為例，預估OPAP混合比例的聯立方程式可寫成$1 \times K_1 + 2 \times K_2 = F$；$K_1 + K_2 = 1$。其中$K_1$表示1-OPAP在像素$p_1$中的比例，$K_2$表示2-OPAP在像素$p_1$中的比例。根據上述，我們可以分別解出$K_1 = -F + 2$，$K_2 = F - 1$。因此1-OPAP所佔的比例為93.35%，2-OPAP所佔的比例為6.65%。最後，我們計算單一像素之均方誤差（Mean Square Error，MSE）並換算成PSNR。

$$MSE = (0.75 + (0.5 \times 0.9335 + 1.5 \times 0.0665))/4 \cong 0.329$$

其中0.5為1-OPAP之期望變動量，1.5為2-OPAP之期望變動量，期望$PSNR=10\times\log_{10}\frac{255^2}{0.329} \cong 52.96$ dB。顯然的，使用不同的掩護影像，所需搭配之OPAP混合技巧與所預估之比例也會相異。

（二）實作程序：我們依照預估所產生之OPAP混合比例，依照混合比例並以一個順序密鑰決定掩護影像內p_1像素嵌入秘密訊息之順序。當掩護影像p_1像素皆按照嵌入順序完成單一像素嵌入程序後即完成此小結所提出之嵌入演算法。為了達到公允性，我們將每張影像各執行演算法1000次，並取其平均值作為比較值。

表7顯示我們的OPAP混合技巧以及預估之比例與實作之比例之間的比較。在表格中我們以符號OP表示OPAP。例如4-OPAP，就表示成4-OP。

表7 在嵌入量相近時本方法之預估程序與實作程序之比較結果

影像	欲達到之嵌入量C_{pre}（位元）	p_1混合式OPAP比例預估（%）		預估產生影像品質PSNR（dB）	實作產生影像品質PSNR（dB）	實作與預估之PSNR誤差（dB）
Lena	200,968	1-OP 93.35	2-OP 6.65	52.96	52.9571	0.0029
Baboon	425,412	4-OP 50.87	5-OP 49.13	36.85	36.8499	0.0001
Jet	182,637	1-OP 78.68		53.57	53.5696	0.0004
Peppers	195,598	1-OP 98.46		53.21	53.2086	0.0014
Sailboat	278,134	2-OP 75.60	3-OP 24.40	49.06	49.0643	0.0043
Splash	132,909	1-OP 2.80		55.32	55.3206	0.0006

以Sailboat影像爲例，Jung等人的方法之嵌入量爲277,995個位元，我們加上0.05%之比例後，其嵌入量變爲278,134個位元，此也代表Jung等人的方法之嵌入量（C_{pre}）爲278,134個位元。據此，我們可以預估使用OPAP混合技巧之單位像素嵌入量F。F=(278134−131072)/65536≅2.24。因此，使用OPAP混合技巧必須搭配2個位元，亦即2-OPAP與3個位元之OPAP嵌入方式，亦即3-OPAP。接著，我們說明如何預估OPAP混合的比例。

當$2 \le F \le 3$，就以2-OPAP與3-OPAP混合技巧嵌入秘密訊息，則以Sailboat影像爲例，預估OPAP混合比例的聯立方程式可寫成$2 \times K_2 + 3 \times K_3 = F; K_2 + K_3 = 1$。其中$K_2$表示2-OPAP在像素$p_1$中的比例，$K_3$表示3-OPAP在像素$p_1$中的比例。根據上述，我們可以分別解出$K_2=-F+3$，$K_3=F-2$。因此2-OPAP所佔的比例爲75.60%，3-OPAP所佔的比例爲24.40%。最後，我們計算單一像素之均方誤差並換算成PSNR。MSE = (0.75 + (1.5 × 0.756 + 5.5 × 0.244)) / 4≅0.8065，其中1.5爲2-OPAP之期望變動量，5.5爲3-OPAP之期望變動量，期望$PSNR=10\times\log_{10}\frac{255^2}{0.8065}$ ≅ 49.06 dB。實作時，我們依照預估出來的比例實際去嵌入秘密訊息。爲了驗證我們預估演算法之正確性及PSNR精準性，我們每張影像各執行1000次取其平均值。接著將平均所產生的PSNR值與預估時所產生的PSNR值比對。實作結果顯示以Sailboat影像而言2-OPAP約佔75.60%，3-OPAP約佔24.40%，再分別嵌入2-OPAP及3-OPAP之秘密訊息。因此使用LSB-matching嵌入之秘密訊息個數（LM）再加上混合式OPAP嵌入之秘密訊息個數一共可以產生131072 + 65336 × (2 × 0.756 + 3 × 0.244) ≅ 278,134個位元。產生之僞裝影像品質爲49.0643 dB。預估結果與實作結果之僞裝影像品質誤差僅有0.0043 dB。

以Peppers影像爲例，Jung等人的方法之嵌入量爲195,500個位元，我們加上0.05%之比例後，其嵌入量變爲195,598個位元，此也代表Jung等人的方法之嵌入量（C_{pre}）爲195,598個位元。據此，我們可以預估使用OPAP混合技巧之單位像素嵌入量F。F = (195598−131072)/65536≅0.9846。因此，將像素p_1中98.46%使用1-OPAP就能達到我們所要的嵌入量。最後，我們計算單一像素之均方誤差並換算成PSNR。MSE=(0.75 + (0.5 × 0.9846)) / 4 ≅ 0.3106，其中0.5爲1-OPAP之期望變動量，期望$PSNR=10\times\log_{10}\frac{255^2}{0.3106}$ ≅ 53.21 dB。實作時，我們依照預估出來的比例實際去嵌入秘密訊息。爲了驗證我們預估演算法之正確性及PSNR精準性，我們每張影像各執行1000次取其平均值。接著將平均所產生的PSNR值與預估時所產生的PSNR值比對。實作結

果顯示以Peppers影像而言1-OPAP約佔98.46%，再嵌入1-OPAP之秘密訊息。因此使用LSB-matching嵌入之秘密訊息個數(LM)再加上嵌入1-OPAP之秘密訊息個數一共可以產生131072 + 65336 × (1 × 0.9846)≅195,598個位元。產生之偽裝影像品質爲53.2086dB。預估結果與實作結果之偽裝影像品質誤差僅有0.0014 dB。

整體實驗結果顯示：偽裝影像品質誤差最小的是Baboon影像誤差約爲0.0001 dB；誤差最大的是Sailboat影像誤差約爲0.0043 dB。平均每張影像之偽裝影像品質誤差約爲0.0016 dB。據此，預估結果與實作結果極爲相似，表示我們所預估的比例以及使用的混合技巧相當的精準。

表8顯示，Baboon影像與Jet影像對Jung等人的方法產生的偽裝影像品質分別具有最小與最大的差別。由於原始Jung等人文獻沒有Sailboat與Splash影像之實驗數據，因此我們以Lena、Baboon、Jet、Peppers影像來比較。就Baboon而言，使用我們的方法可以嵌入425,412個位元，具有偽裝影像品質36.8499 dB。然而，若使用Jung等人的方法，僅能嵌入425,199個位元，其掩護影像品質已經降低至35.46 dB。我們的方法相較於Jung之方法，偽裝影像品質提昇1.3899 dB，並擁有相近之嵌入量。就Jet而言，使用我們的方法可以嵌入182,637個位元，具有偽裝影像品質53.5696 dB。然而，若使用Jung等人的方法，僅能嵌入182,546個位元，其偽裝影像品質已經降低至39.81 dB。我們的方法相較於Jung之方法，偽裝影像品質可提昇大約13.7596 dB，並擁有相近之嵌入量。整體之實驗結果顯示：在嵌入量相近之下我們方法得到的偽裝影像品質比Jung他們多1.39-13.76 dB。嵌入量相近時，我們方法所產生的偽裝影像，如圖17所示。

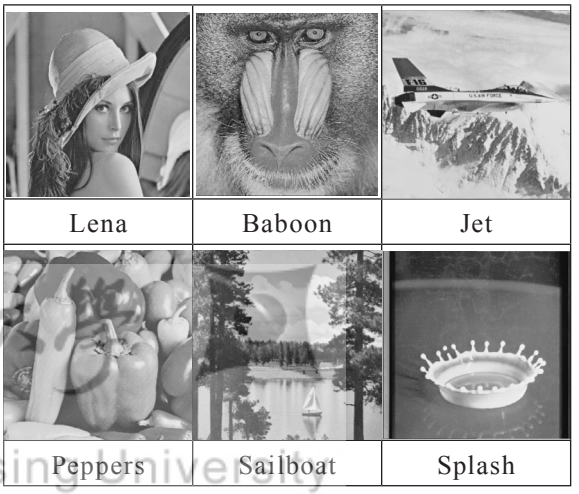


圖17　嵌入量相近時，我們方法所產生的偽裝影像

表8　在嵌入量相近時本方法與Jung and Yoo的方法其偽裝影像品質的比較結果

影像	我們的方法-實作程序		Jung and Yoo的方法		藏量多出0.05%之下PSNR提昇量 (dB)
	Capacity (位元)	PSNR (dB)	Capacity (位元)	PSNR (dB)	
Lena	200,968	52.9571	200,868	41.46	11.4971
Baboon	425,412	36.8499	425,199	35.46	1.3899
Jet	182,637	53.5696	182,546	39.81	13.7596
Peppers	195,598	53.2086	195,500	41.93	11.2786
Sailboat	278,134	49.0643	277,995	37.71	11.3543
Splash	132,909	55.3206	132,843	42.45	12.8706

表9　本論文所使用之輸入影像其RS偵測結果

輸入影像	R_m	R_{-m}	S_m	S_{-m}	R_{ratio}	S_{ratio}
Lena	35.21%	34.85%	22.45%	22.78%	1.02%	1.48%
Baboon	34.95%	35.61%	30.69%	30.00%	1.89%	2.25%
Jet	40.51%	41.63%	20.32%	19.87%	2.76%	2.19%
Peppers	34.35%	34.58%	25.21%	24.97%	0.68%	0.94%
Sailboat	35.79%	35.65%	26.89%	27.05%	0.39%	0.59%
Splash	40.31%	39.95%	20.37%	20.24%	0.87%	0.64%

表10 內插後產生的掩護影像其RS偵測結果

掩護影像	R_m	R_{-m}	S_m	S_{-m}	R_{ratio}	S_{ratio}
Lena	37.13%	36.87%	13.73%	13.89%	0.68%	1.23%
Baboon	32.84%	33.34%	24.28%	23.95%	1.52%	1.39%
Jet	45.54%	45.95%	12.58%	12.63%	0.90%	0.44%
Peppers	37.21%	37.15%	13.29%	13.61%	0.15%	2.36%
Sailboat	36.33%	36.19%	16.99%	17.07%	0.39%	0.49%
Splash	45.24%	45.58%	11.95%	11.80%	0.75%	1.25%

表11 偽裝影像品質相近時本方法所產生的Stego Image之RS偵測結果

偽裝影像	p_1混合式 OPAP比例（%）		R_m	R_{-m}	S_m	S_{-m}	R_{ratio}	S_{ratio}
Lena	3-OP 23.47	4-OP 76.53	34.39%	34.25%	19.48%	20.19%	0.42%	3.67%
Baboon	4-OP 19.64	5-OP 80.36	32.59%	34.85%	28.50%	26.48%	6.93%	7.08%
Jet	4-OP 92.50	5-OP 7.50	35.70%	38.57%	21.81%	20.50%	8.03%	6.01%
Peppers	3-OP 35.33	4-OP 64.67	34.98%	34.48%	18.89%	20.43%	1.40%	8.16%
Sailboat	4-OP 66.20	5-OP 33.80	33.75%	34.74%	22.78%	22.43%	2.93%	1.54%
Splash	3-OP 47.04	4-OP 52.96	37.27%	40.46%	20.81%	18.79%	8.57%	9.71%

表12 嵌入量相近時本方法所產生的Stego Image之RS偵測結果

偽裝影像	p_1混合式 OPAP比例（%）		R_m	R_{-m}	S_m	S_{-m}	R_{ratio}	S_{ratio}
Lena	1-OP 93.35	2-OP 6.65	32.71%	38.49%	16.47%	13.64%	17.69%	17.18%
Baboon	4-OP 50.87	5-OP 49.13	33.62%	34.02%	27.18%	27.06%	1.18%	0.44%
Jet	1-OP 78.68		38.72%	47.00%	16.60%	13.05%	21.37%	21.37%
Peppers	1-OP 98.46		33.02%	38.97%	15.94%	13.43%	18.02%	15.76%
Sailboat	2-OP 75.60	3-OP 24.40	34.56%	34.50%	19.46%	19.76%	0.17%	1.55%
Splash	1-OP 2.80		41.38%	45.74%	14.81%	12.43%	10.53%	16.05%

五、資訊偽裝偵測——RS偵測結果

由前述可知我們的方法能夠嵌入更多的秘密訊息以及擁有更高的偽裝影像品質。然而一個好的資訊隱藏技術除了需要滿足高嵌入量以及高偽裝影像品質之外，更需達到能防偽裝偵測的能力 [18]。有鑑於此，本文使用RS偵測對所有使用的輸入影像以及本方法所產生的偽裝影像進行分析，並分別將所有輸入影像、內插後產生的掩護影像以及偽裝影像之RS偵測結果分別列在表9、表10、表11、表12。其中偽裝影像偵測方面可分成兩個部份：（一）當偽裝影像品質相近時本方法所產生的Stego Image之RS偵測結果；（二）當嵌入量相近時本方法所產生的Stego Image之RS偵測結果。根據式（7）我們在表格中列出R_m、R_{-m}、S_m、S_{-m}以及$|R_m-R_{-m}|/R_m$、$|S_m-S_{-m}|/S_m$等數據。其中$|R_m-R_{-m}|/R_m$與$|S_m-S_{-m}|/S_m$分別爲R_m、R_{-m}以及S_m、S_{-m}之相異程度，在此我們稱爲RS相異程度，我們分別以R_{ratio}、S_{ratio}表示。RS相異程度越高表示偽裝影像裡的秘密訊息越容易被偵測出來。

偵測結果顯示：在完全沒嵌入秘密訊息的六張輸入影像以及內插放大後產生的掩護影像的RS相異程度之範圍爲0.15%-2.76%，表示RS偵測仍會造成些許誤判的情形。在偽裝影像偵測方面當偽裝影像品質相近時本方法所產生的Stego Image其RS相異程度之範圍爲0.42%-9.71%，整體而言RS相異程度都是偏低；當嵌入量相近時本方法所產生的Stego Image其RS相異程度之範圍爲0.17%-21.37%，很明顯的只要有使用到OPAP-1所產的偽裝影像例如：Lena、Jet、Peppers、Splash其RS相異程度就會增加。如果要降低被偵測率，我們可以使用2-OPAP犧牲些許偽裝影像品質達到防偽裝偵測的目的，實驗數據如表13所示，整體而言偽裝影像品質下降0.04dB-0.78dB。Lena爲52.25dB、Jet爲52.88 dB、Peppers爲52.43 dB，Splash爲55.28 dB。另外RS相異程度下降的比率範圍爲9.26%-94.38%。以Jet偽裝影像的RS相異程度下降最多其R_{ratio}、S_{ratio}分別爲3.66%、1.20%；相異程度下降幅度較少的影像爲Splash，其R_{ratio}、S_{ratio}分別爲8.79%、14.56%。我們實驗證實使用2-OPAP確實能有效避免RS偵測攻擊。

六、結論與未來工作

我們不僅改正了Jung等人演算法之錯誤，我們更提出一個全新的方法。本篇論文針對灰階影像提出了一個具有高藏量、高影像品質，以及可回復之資料隱藏技術。我們提出的方法是基於影像內插並使用混合式OPAP搭配LSB-matching所做的資訊隱藏技術。首先將輸入影像縮小，再以我們提出的影像內插方法將其放大，然後執行高斯模糊即可得到一張掩護影像。我們的方法得到的影像內插品質比Jung他們的方法平均最多大約高4.86 dB。

我們使用掩護影像來嵌入秘密訊息；首先將掩護影像以2 × 2爲一個區塊做分割，其中我們不去更動它，p_1使用混合式OPAP來嵌入秘密訊息，p_2、p_3使用LSB-matching來嵌入秘密訊息，在此我們做了平均每個像素的期望變動量分析，可以明確的預測在多少偽裝影像品質之下其所產生之藏量，以及出在嵌入多少個位元之下其所產生之偽裝影像品質。實驗結果顯示：掩護影像嵌入秘密訊息之後，在偽裝影像品質與Jung他們相近之下我們方法得到的藏量較Jung他們多4.86%-118.10%；在嵌入量

表13　改善嵌入量相近時本方法所產生的Stego Image之RS偵測結果

偽裝影像	p_1混合式OPAP比例（%）	PSNR（dB）	R_m	R_{-m}	S_m	S_{-m}	R_{ratio}	S_{ratio}
Lena	2-OP 53.32	52.25	35.20%	36.02%	15.66%	15.21%	2.33%	2.91%
Jet	2-OP 39.34	52.88	41.51%	43.03%	15.34%	15.52%	3.66%	1.20%
Peppers	2-OP 49.23	52.43	35.56%	36.59%	15.38%	14.96%	2.88%	2.74%
Splash	2-OP 1.40	55.28	41.85%	45.53%	14.90%	12.73%	8.79%	14.56%

與Jung他們相近之下我們方法得到的偽裝影像品質較Jung他們多1.39-13.76 dB。因此，我們提出的方法具有高藏量、高影像品質以及僅需極少數額外資訊達到可回復之特性。

我們使用著名的RS偵測技術分析我們所使用以及產生的影像，包含：輸入影像、內插放大後的掩護影像、偽裝影像。其中偽裝影像偵測方面可分成兩個部份：（一）當偽裝影像品質相近時本方法所產生的Stego Image之RS偵測結果；（二）當嵌入量相近時本方法所產生的Stego Image之RS偵測結果。實驗結果顯示：我們的資料隱藏方法所產生的偽裝影像被RS偵測技術所察覺的機率都是偏低，表示我們的方法具有防偽裝偵測的能力。

雖然我們演算法所得到的影像內插的品質皆高於 [8]，但是我們認爲尚有改進的空間。例如：使用預測效果更爲精準之像素預估器。此外偽裝影像之藏量與品質皆有再改進的空間。例如：使用具有更高藏量與偽裝影像品質之資料隱藏演算法。

參考文獻

1. Petitcolas, F.A.P., Anderson, R.J. and Kuhn, M.G., "Information hiding -- A survey," *Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1062-1078, (1999).
2. Provos, N. and Honeyman, P., "Hide and Seek: an Introduction to Steganography," *IEEE Security & Privacy*, Vol. 1, No. 3, pp. 32-44, (2003).
3. Chan, C.K. and Cheng, L.M., "Hiding Data in Images by Simple LSB Substitution," *Pattern Recognition*, Vol. 37, pp. 469-474, (2004).
4. Chang, C.C. and Tseng, H.W., "A Steganographic Method for Digital Images Using Side Match," *Pattern Recognition Letters*, No. 25, pp. 1431-1437, (2004).
5. Cox, I., Miller, M., Bloom, J., Fridrich, J. and Kalker, T., *Digital Watermarking and Steganography*, Morgan Kaufmann, 2nd ed., (2007).
6. Johnson, N.F. and Jajodia, S., "Steganography: Seeing the Unseen," *IEEE Transactions on Computers*, Vol. 31, pp. 26-34, (1998).
7. Lin, I.C., Lin, Y.B. and Wang, C.M., "Hiding Data in Spatial Domain Images with Distortion Tolerance," *Computer Standards & Interfaces*, Vol. 31, No. 2, pp. 458-464, (2009).
8. Jung, K.H. and Yoo, K.Y., "Data Hiding Method Using Image Interpolation," *Computer Standards & Interfaces*, Vol. 31, No. 2, pp. 465-470, (2009).
9. Farid, H., "Detecting Hidden Messages Using Higher-Order Statistical Models," *International Conference on Image Processing*, pp. 905-908, (2002).
10. Fridrich, J., Goljan, M. and Du, R., "Reliable detection of LSB steganography in grayscale and color images," *Proceedings of ACM, Special Session on Multimedia Security and Watermarking*, pp. 27-30, (2001).
11. Carrato, S. and Tenze, L., "A High Quality 2 Image Interpolator," *IEEE Signal Processing Letters*, Vol. 7, No. 6, pp. 132-134, (2000).
12. Gonzalez, R.C. and Woods, R.E., *Digital Image Processing*, Addison Wesley, 2nd ed., (2002).
13. Wu, X. and Memon, N., "Context-Based, Adaptive, Lossless Image Coding," *IEEE Transactions on Communications*, Vol. 45, No. 4, pp. 437-444, (1997).
14. Chang, C.C., Lin, C.C. and Chen, Y.H., "Reversible Data-Embedding Scheme Using Differences between Original and Predicted Pixel Values," *IET Information Security*, Vol. 2, No. 2, pp. 35-46, (2008).
15. Li, X. and Orchard, M.T., "Edge-Directed Prediction for Lossless Compression of Natural Images," *IEEE Transactions on Image Processing*, Vol. 10, No. 6, pp. 813-817, (2001).
16. Martucci, S.A., "Reversible compression of HDTV images using Median Adaptive Prediction and Arithmetic Coding," *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 1310-1313, (1990).
17. Mielikainen, J., "LSB Matching Revisited," *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 285-287, (2006).
18. Lou, D.C., Wu, N.I., Wang, C.M., Lin, Z.H. and Tsai, C.S. (in press), "A Novel Adaptive Steganography Based on Local Complexity and Human Vision Sensitivity," *The Journal of Systems and Software*, (2010).

Manuscript Received：*Dec. 30, 2009*
Revision Received：*Feb. 25, 2010*
and Accepted：*Feb. 27, 2010*