植基於像素排序法之可回復資訊隱藏技術

劉興漢　楊顓豪*　林永祥

國防大學管理學院資訊管理學系

摘　要

本研究是基於 Weng 等學者於 2017 年提出的最佳化（Optimal）基於像素之像素排序法（Pixel-based Pixel Value Ordering, PPVO）及 Wu 等學者於 2020 年提出的改良式方法（Improved PPVO），來建構出一套可回復式資訊隱藏方法，在提取秘密訊息的同時，可以還原出原始影像。本方法可在人眼難以察覺之細微變化下，於灰階影像中藏入秘密訊息，並在低藏密量的前提下，有效提昇藏密品質，使之更為隱密。本方法將載體影像以棋盤式切割成兩部分，再各自獨立進行 PPVO 藏密，並依照 PPVO 編碼像素的區塊複雜度（光滑程度），來適應性地調整參照區塊大小。於測試驗證上，以 BOSSBase 自然影像資料庫中之載體影像為例，在各種藏密量的設定下，所獲得之影像品質普遍優於 Weng 等學者提出的最佳化 PPVO 方法和 Wu 等學者提出的改良式 PPVO 方法。本方法並可抵抗正規/特異（Regular-Singular, RS）分析和像素差直方圖分析，並在廣泛性測試有良好的表現。

關鍵詞：資訊隱藏，可回復資訊隱藏，適應性像素修改，PPVO

A Reversible Image Steganographic Scheme Based on Pixel Value Ordering

Hsing-Han Liu, Chuan-Hao Yang*, and Yung-Hsiang Lin

Department of Information Management, Management College, National Defense University

ABSTRACT

In this study, a reversible image steganographic scheme that is able to restore original cover images after retrieving secret information is developed based on the optimal pixel-based pixel value ordering (PPVO) by Weng et al. (2017) and the improved PPVO by Wu et al. (2020). By this method, secret information is embedded in grayscale images and only causes minor changes imperceptible to human vision. The quality of stego images is enhanced in the case that a certain low amount of information is embedded. The cover image is split into two parts interleavedly in a checkerboard-like manner, and each part performs the embedding process independently. With different pixel complexity, the size of context pixels is adaptively modified. During experimental verification, the images in BOSSBase dataset are adopted as the cover images. The quality of stego images is generally better by using this method, compared with the one by Weng et al. (2017) and Wu et al. (2020). This method can also prevent the stego images from being detected by the regular-singular (RS) analysis and the pixel difference histogram (PDH) analysis, and performs well during generalized benchmarking.

Keywords: steganography, reversible image steganography, adaptive pixel-modification, PPVO

文稿收件日期 111. 06. 22; 文稿修正後接受日期 111. 10. 21; *通訊作者
Manuscript received Jun. 22, 2022; revised Oct. 21, 2022; * Corresponding author

一、前　言

隨著近年來網際網路的快速發展，大量資訊以網路為媒介快速流通，使得資訊的傳遞更為便利。然而，未經保護的明文資料，在傳遞過程中會暴露在安全威脅之下。因此，如何安全地傳遞資料，在現階段是一個重要的課題。其中被廣泛研究的領域包含：密碼學（Cryptography）、數位浮水印、資訊隱藏（Information Hiding）等。資訊隱藏著重於不可察覺性，亦即將秘密訊息以不可察覺的方式隱藏於載體中，以本研究為例，載體即為影像，可用於隱藏版權資訊、內容檢查，以及軍事上的秘密通訊等等。

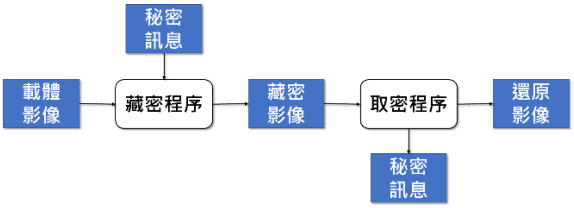


圖 1. 可回復式之影像藏密與取密基本流程圖

傳統的影像藏密技術，通常會在載體影像上產生不可恢復的失真，因此無法利用於某些需要維持載體完整性的場域，如醫學影像標記及軍用影像處理，於是就有了可回復式資料隱藏技術（Reversible Data Hiding, RDH）的出現。如圖 1 所示，可回復性，亦即從藏密影像（Stego Image）取出秘密訊息（Secret Text）的同時，可將影像恢復成原始的載體影像。為了確保可回復性，任何會導致永久失真的技術都不能運用於可回復式影像藏密技術。根據 Kaur 等人[1]的整理，先前的研究者所開發出的幾種無損的藏密方式（如圖 2 所示）包含：無損壓縮（Lossless Compression, LC）[2]、差異擴張（Difference Expansion, DE）[3]、直方圖位移（Histogram Shifting, HS）[4]、預測誤差擴張（Prediction-Error Expansion, PEE）[5]、像素排序（Pixel Value Ordering, PVO）[6]等方式。

無損壓縮法，是以不損毀影像資訊的方式壓縮載體影像的一部分，來騰出空間，並將秘密訊息嵌入此間隙中。由於無損壓縮演算法的壓縮率較低，採用此方法的缺點是可供嵌入的容量（Embedding Capacity, EC）較小。差異擴張法則是以兩個像素為一組編碼單元，每個單元至多可藏入 1 位元，最終的藏密率至多接近 0.5 位元/像素（bits per pixel, bpp）。直方圖位移法，可將秘密訊息的位元串流，嵌入影像中最頻繁出現的像素，亦即直方圖的峰值。此方法會改變該像素在直方圖中的位置，而其餘沒有參與藏密的像素，也必須各自位移 1，以利於在影像的直方圖中為待嵌入像素騰出空間，來維持取密時的可回復性。

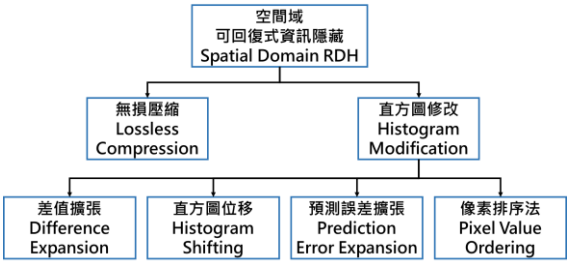


圖 2. 空間域之可回復式資訊隱藏

差異擴張法中，每個待嵌入像素，都由相鄰的一個像素做預測。而有鑑於 Tian[3]的差異擴張法具有較差的預測效果，Thodi 等人[5]提出的預測誤差擴張法（Prediction Error Expansion, PEE），以臨近的 3 個像素預測待嵌入像素，並擴張預測值和原值之間的差值，藏入 1 位元的秘密訊息。其中直方圖位移法所採用的壓縮溢位映射表（Location Map, LM）的流程也被 Thodi 和 Rodriguez[5]採納，用來記錄可能會在藏密過程中溢位的區塊所在。

接下來是 Li 等人[6]所提出基於像素排序法（PVO）之方式。此類方法在不需太高嵌入容量的情況下，可達到極高的藏密影像品質。它是由 PEE 方法延伸而來，並使用 PVO 做為預測方式。待編碼區塊的像素以遞增方式重新排序，並以次大（次小）的像素預測最大（最小）的像素，並以預測誤差（Prediction Error, PE）的絕對值生成直方圖。PVO 方法產生的預測誤差，相較於其他 PEE 衍伸的方法，在低嵌入容量的前提下可以生成更正確的預測結果。後續再以此預測誤差直方圖（Prediction Error Histogram, PEH）透過類似直方圖位移法的方式進行藏密。其中預測誤差為 1（|PE|=1）的區塊通常是 PEH 中的峰值（bin 1），可用於嵌入秘密訊息，而剩餘的區塊則位移 1，以騰出可藏密空間。

依照 Li 等人[6]的方式，會排除預測誤差為 0（即|PE|=0、bin 0）的部分，因而不可用於藏密，但通常此區塊在 PEH 中是次多的，如果也可用於藏密，則可以增加藏密容量。為了改進這一點，Peng 等人[7]提出了改良式 PVO（Improved PVO, IPVO），將次大及次小的預測誤差部分也用於藏密，當然也使用到了預測誤差為 0 的部分。而為了更有效利用這個部分，Ou 等人[8]提出 PVO-k 方法，其中 k 代表最大（最小）的 k 個像素，在此方法中會同時被修改，用來藏密。

後續基於 PVO 的方式，主要包含：變動區塊 PVO（Variable-size Partitioning PVO）[9]、基於像素之 PVO（Pixel-based PVO, PPVO）[10]、像素對 PVO（Pairwise PVO）[11]、最佳化 PPVO（Optimal PPVO）[12]、改良式 PPVO（Improved PPVO）[13]。而 Kaur 等學者[1]曾於 2020 年針對 PVO 相關方法進行比較（如表 1 所示）。

表 1. PVO 相關方法比較

提出者	劣勢
Li 等人[6]	個別區塊內至多只利用 2 個像素進行嵌入，且只有 PEH 的 bin 0 用於嵌入。
Qu 等人[10]	以像素為藏密單位，使用參照區塊計算 PE 和複雜度，可提供比 Li 等人[6]之方法更大的容量和較佳的藏密品質，但藏密也只使用 bin 0 參與嵌入。
Weng 等人[12]	和 Qu 等人[10]的方法相比，參照區塊更完整地包圍待嵌入像素，並可根據影像品質決定用於藏密的 PEH bin，但參照區塊大小在整張影像中是固定的。
Wu 等人[13]	和 Qu 等人[10]的方法相比，參照區塊更完整地包圍待嵌入像素，並且可隨著影響內容調整區塊大小，但藏密也只使用 bin 0 參與嵌入。

PPVO 的主要概念，是以排序後的參照區塊（Context Pixels）來預測待編碼像素。具體而言，給定待編碼像素（即待預測像素 x），其參照區塊為 n 個右方及下方的最近相鄰像素（編碼方向由左至右，由上而下），以遞增方式排序之。在參照區塊中的所有像素皆大於（或小於）像素 x 的前提下，其中的最大（或最小）像素則可用於預測像素 x。此方式幾乎可以預測所有載體影像上的像素，同時也因為可得到的預測誤差（PE）數量更多，所以其中可用於藏密的部分，也會較其他方式多。在較光滑的影像中，PPVO 可達成比傳統 PVO 更高的嵌入容量（EC）。

Weng 等人[12]提出的最佳化 PPVO，則是運用更緊湊的參照區塊，使用了區塊內全部像素的平均值來進行預測。而由於參照區塊和待編碼像素排列得更緊密，和原先的 PPVO 相比，能更準確地做預測，因此在相同藏密量下可以達到更好的品質。此外，原始的 PVO 及 PPVO 方法固定使用 PEH 中的 bin 0 嵌入秘密訊息，雖然容量較高，但藏密後的品質不一定較好，於是作者在此方法中加入一種機制，使其在嵌入時可以在保證容量足夠的前提下，動態調整所使用的 PEH bin。根據 Kaur 等人[1]的比較結果，指出該方法運用在粗糙的載體影像（如：USC-SIPI 影像資料庫中之 Baboon 載體），可以達到比原始 PPVO 更好的品質，但是運用在相對平滑的載體影像時，其品質並沒有明顯提升。

Wu 等人[13]提出的改良式 PPVO，也改善了一些原始 PPVO[10]的不足。例如原始 PPVO 在參照區塊的選取上，並沒有利用待編碼像素的左側，而此方法則可同時利用待編碼像素的左下、下方及右下方的臨近像素做為參照區塊。然而，此方法和原始 PPVO 相同，僅參考該區塊的最大及最小值，而且在粗糙區塊中，藏密影像之品質沒有明顯變化。

本研究的主要目的，是在達到某種藏密容量的前提下，提昇藏密後的影像品質，並維持影像結構。其中影像品質以峰值訊號雜訊比（Peak Signal-Noise Ratio, PSNR）做為比較基準。本研究將以 Weng 等人[12]和 Wu 等人[13]的方法為基礎，針對預測參照區塊範圍及預測方式進行改良，以提升藏密後影像之品質，同時能維持 Weng 等人[12]所提出之方法針對複雜圖片的預測效果。

二、文獻探討

2.1 像素排序法（PVO）

像素排序法（PVO）的概念，最初是由 Li 等人於 2013 年[6]提出。此方法的優點是可維

持極高的藏密影像品質，因而吸引了大量學者的興趣。以 8 位元灰階影像為例，每一像素以無符號 8 位元整數表示，範圍介於 0 (00000000_2) 到 255 (11111111_2) 之間，用來代表該像素之亮度，0 為黑色，而 255 為白色。

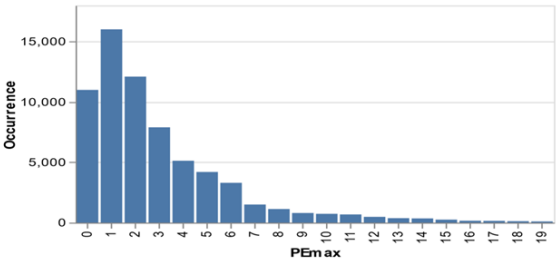


圖 3. 一般影像的 PEH[6]

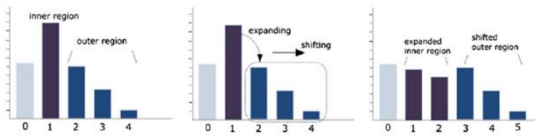


圖 4. 嵌入預測誤差直方圖（PEH）的變化[6]

根據 Li 等人[6]的方法，首先，將載體影像切割成不重疊的相等大小區塊$(n_1 \times n_2)$，並將該區塊內的所有像素以遞增排序。接著計算該區塊之複雜度，預先過濾掉不適合的區塊。此處所指之複雜度，為該區塊次大及次小像素的差值，如果超過設定之閾值，則認定此區塊相對粗糙，在計算預測誤差時通常無法將所預測之像素作為嵌入目標，而只能位移該像素。根據圖 3 的預測誤差直方圖（PEH），為了避免這類多餘的位移（N_E），作者將這類像素維持不變，在藏密前即跳過。後續根據編碼計算最小預測誤差PE_{min}及最大預測誤差PE_{max}。當$|PE_{\{min,max\}}|$為 1，即 PEH 中的 bin 1 (圖 4)。而依據待嵌入之秘密訊息位元是否為 1，調整$PE_{\{min,max\}}$成 1 或 2，而其他 bin $\{PE: PE > 1\}$則必須為 bin 1 騰出編碼空間，因此此類像素的 PE 需向外移動 1。若以$PE_{max} = 2$為例，則是調整成$PE'_{max} = 3$（此處PE'代表調整後之值）。計算出新的像素值後，再依排序前該像素的原始位置，將其置入藏密影像。最後嵌入還原資訊（Auxiliary information），生成藏密影像。其中，還原資訊主要包含藏密過程中所使用的參數，來提供後續取密演算法使用。

於取密的過程中，首先取出還原資訊，從中取得記錄的區塊大小及複雜度閾值等藏密時使用之參數。後續，依據記錄的區塊大小，將藏密影像切割成不重疊的相同大小區塊，再以增序的方式排列各區塊內部的像素。接著計算區塊複雜度，排除過於粗糙之區塊，並依照修改後的$PE'_{\{min,max\}}$為 1 或 2，取得該區塊嵌入之秘密訊息位元 b，再將$PE'_{\{min,max\}} \geq 2$的部分反向位移 1，還原成原始影像。當整張影像解密過一遍後，即可獲得完整的秘密訊息和還原後的原始影像。

此方法的主要缺點，在於計算複雜度和 PE'時，僅僅依據各區塊內部的像素進行預測，而各區塊周圍的像素則會被忽略掉，使得預測的準確度在區塊較小時會變得較不準確。此外，各區塊只有兩個像素參與藏密，導致容量上有所侷限。針對這些問題，Qu 與 Kim[10]提出了以像素為嵌入單位的 PPVO，來進行改善。

2.2 像素式像素排序法（PPVO）

由 Qu 和 Kim[10]提出的像素式像素排序法（PPVO），是改進自 Li 等人[6]的原始 PVO 方法。其主要區別，在於以滑動窗口（Sliding Window）的方式，對全圖像素逐一進行嵌入。而該滑動窗口，也稱為參照區塊（Context Pixels），用於複雜度篩選及預測誤差（PE）之計算，作為嵌入流程的一部分。其藏密步驟說明如下：

步驟 1：溢位處理

將載體影像中像素值為 0 及 255 的像素，分別修改為 1 及 254，並將修改的像素在溢位映射表（Location Map, LM）中設定為 $LM_{(i)}$=1，而其餘值介於 1 和 254 之間的像素，則設定為 $LM_{(i)}$=0，並將 LM 以算術編碼進行無損壓縮，得到 CLM，長度縮減為 l_{clm}，稍後和秘密訊息一併嵌入至載體影像中。

步驟 2：還原資訊處理

令 L 為藏有秘密訊息及 CLM 的最後一個像素位置，而 N 為影像像素總數。將載體影像最後一行前$16 + 2\lceil \log_2 N \rceil$個像素的最低有效位元（$LSBs$）原始值保存為 H_{LSB}，隨後和秘密訊息一併嵌入影像。而在後續的步驟中，會將還原資訊以 LSB 取代法，嵌入到上述這些像素中。

步驟 3：計算參照區塊大小

參照區塊 C 和目標像素x的相對位置關係，如圖 5 所示。其順序是以參照像素c_i和目標像素x之間的歐幾里得距離由近而遠，且由上而下排列。而採取之像素個數，以編碼時指定的參照像素數量c_{CN}為限。

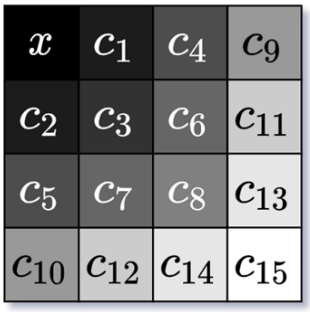


圖 5 PPVO 參照區塊樣式

步驟 4：過濾適合的目標像素

依據公式(1)，以參照區塊中最大像素值$\max(C)$減去最小像素值$\min(C)$，來計算區塊複雜度(NL)。對於複雜度超過閾值T的像素x，則不進行藏入或位移，避免造成非必要的位移。

$$NL = \max(C) - \min(C) \tag{1}$$

步驟 5：藏入秘密訊息

待嵌入像素 p 先以公式(2)和參照區塊 C 進行比較，以計算預測值$\hat{p}$，再以公式(3)計算原始像素 p 與預測值$\hat{p}$之差，獲得預測誤差 PE。後續，根據 PE 和預測值$\hat{p}$（公式(4)），來決定 p 需偏移 1，或嵌入一位元秘密訊息 b，以獲得修改後像素p'。運用相同方式逐行掃描，以由左至右、由上至下的順序進行嵌入，直到秘密訊息結束或容量不足，並記錄最後一個修改的像素位置 L，一併納入還原資訊中。

$$\hat{p} = \begin{cases} \max(C) & if\ \min(C) \neq \max(C)\ and\ \ p \geq \max(C) \\ \min(C) & if\ \min(C) \neq \max(C)\ and\ \ p \leq \min(C) \\ 254 & if\ \max(C) = \min(C) = VC = 254 \\ VC & if\ \max(C) = \min(C) = VC, 1 \leq VC \leq 253 \\ & otherwise\ \ skipped \end{cases} \tag{2}$$

$$PE = p - \hat{p} \tag{3}$$

$$p' = \begin{cases} p + b & if\ \ \hat{p} = \max(C)\ and\ \ PE = 0 \\ p + 1 & if\ \ \hat{p} = \max(C)\ and\ \ PE > 0 \\ p - b & if\ \ \hat{p} = \min(C)\ and\ \ PE = 0 \\ p - 1 & if\ \ \hat{p} = \min(C)\ and\ \ PE < 0 \\ & otherwise \rightarrow p \end{cases}$$
$$where\ max(C) \neq min(C) \tag{4}$$

$$p' = \begin{cases} p + b & if\ \ VC = 254\ \ and\ \ PE = 0 \\ p - b & if\ \ VC \neq 254\ \ and\ \ PE = 0 \\ p - 1 & if\ \ VC \neq 254\ \ and\ \ PE < 0 \\ & otherwise \rightarrow p \end{cases}$$
$$where\ max(C) = min(C) = VC$$

步驟 6：嵌入還原資訊

將還原資訊以 LSB 取代法，嵌入至載體影像的最後一行。其中，該行前$16 + 2\lceil \log_2 N \rceil$個像素 *LSBs* 的原始值（$H_{LSB}$），已於步驟 2 預先和秘密訊息一併嵌入載體影像。

本方法的取密步驟如下：

步驟 1：取出還原資訊

依 Qu 和 Kim[10]之方法，從藏密影像最後一行的前$16 + 2\lceil \log_2 N \rceil$個像素的 *LSBs* 中取出還原資訊，其中包含：壓縮溢位映射表長度l_{clm}、篩選閾值 T、區塊像素數量CN，以及修改的最後一個像素位置 L。

步驟 2：取密

由最後一個編碼像素的位置L開始，反向取出藏密影像所藏的秘密訊息，並還原該像素。首先以公式(1)計算區塊複雜度 NL，如 $NL > T$，則跳過該像素不做更動。隨後依照公式(2)計算預測值$\widehat{p'}$，並依公式(3)計算預測誤差 PE，作為偏移與取密的依據。最後由公式(5)取出秘密訊息位元，並還原影像。

$$p, b = \begin{cases} p', b = 0 & if\ \ \widehat{p'} = \max(C)\ and\ \ PE = 0 \\ p' - 1, b = 1 & if\ \ \widehat{p'} = \max(C)\ and\ \ PE = 1 \\ p' - 1, \text{no } b & if\ \ \widehat{p'} = \max(C)\ and\ \ PE > 1 \\ p', b = 0 & if\ \ \widehat{p'} = \min(C)\ and\ \ PE = 0 \\ p' + 1, b = 1 & if\ \ \widehat{p'} = \min(C)\ and\ \ PE = -1 \\ p' + 1, \text{no } b & if\ \ \widehat{p'} = \min(C)\ and\ \ PE < -1 \end{cases}$$
$$where \min(C) \neq \max(C) \tag{5}$$

$$p, b = \begin{cases} p', b = 0 & if\ \ VC = 254\ \ and\ \ PE = 0 \\ p' - 1, b = 1 & if\ \ VC = 254\ \ and\ \ PE = 1 \\ p', b = 0 & if\ \ VC \neq 254\ \ and\ \ PE = 0 \\ p' + 1, b = 1 & if\ \ VC \neq 254\ \ and\ \ PE = -1 \\ p' + 1, \text{no } b & if\ \ VC \neq 254\ \ and\ \ PE < 1 \end{cases}$$
$$where \max(C) = \min(C) = VC$$

步驟 3：溢位還原

解壓縮溢位映射表 LM，並將$LM_{(i)} = 1$的像素，分別由$(254,1)$調整回$(255,0)$。

步驟 4：還原影像末行

針對已於步驟 2 和秘密訊息一併取出的 H_{LSB}，運用 LSB 取代法，還原回影像末行的前$16 + 2 \times \lceil \log N \rceil$個像素。

2.3 最佳化 PPVO

Weng 等人[12]所提出的最佳化 PPVO，是建立在 Qu 等人[10]的 PPVO 基礎上，使參照區塊的像素更緊密地圍繞待嵌入像素 p，並可依照編碼後的失真情形，適應性地調整使用的 PEH bin。bin 0 通常是 PEH 的峰值，可能達到最高的藏密容量，但在容量足夠時，根據載體影像的不同，bin 0 不一定會是達到最高品質的最佳選擇，所以在保證足夠容量的前提下，可根據藏密後之品質來適應性調整所用之 PEH bin，以進行品質上的改善，使得此藏密方式就算在紋理相對粗糙的載體影像（如：USC-SIPI 影像資料庫中之 Baboon 載體），也能達到比其他 PPVO 相關的方法更好的效果。但後續根據 Kaur 等人[1]的比較，將此方法運用於在大部分圖片時，在品質上並沒有特別優異的表現。

此方法的藏密步驟說明如下：

步驟 1：拆分影像

根據 Weng 等人[12]的方法，為了維持演算法的可回復性，參照區塊I_{LSE}中不能包含目前編碼階段會更動到的像素，因此將待編碼像素，以棋盤式拆分成兩階段。如圖 6 所示，灰色部分像素為本階段將修改或嵌入的像素（集合I_A），其中 x 是當下正編碼的像素，同階段c_5、c_6、c_7、c_8及c_{13}為接下來陸續要編碼的像素。而黑色部分像素僅做為參照區塊使用，不進行藏密（集合I_B），如果需要額外容量，則將會於下一層編碼參與藏密。至於白色部分像素，則不採用。

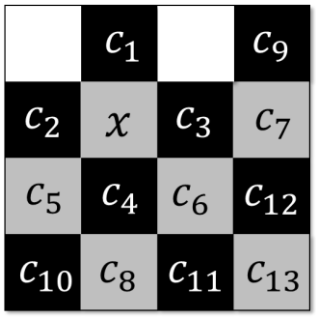


圖 6. 參照區塊及交錯嵌入的樣式

步驟 2：溢位處理

由於 PVO 的藏密方式最多只會改動像素 ± 1，此時須將溢位像素集合，$O_P = \{p \in \{0,255\}\}$的像素記錄於溢位映射表 LM，排除於藏密過程中，並將 LM 以算術編碼壓縮後，記錄於還原資訊。

步驟 3：篩選像素

區塊複雜度Δ可以透過公式(6)計算獲得，其值代表參照區塊的標準差，而篩選閾值為vT。後續將不會溢位的待嵌入像素D_P所在的參照區塊像素做篩選，當$\Delta \leq vT$時，視為平滑像素S_P，否則視為粗糙像素C_P，會跳過而不進行藏密。n為參照區塊取用之像素數量，介於 4 至 18 之間。如圖 7 所示，下限為$n = 4$時涵蓋了周圍上下左右最近的 4 個像素。

$$\Delta = \sigma_{LSE} = \sqrt{\frac{\sum_{t\in\{1,n\}}(c_t - \mu_{LSE})^2}{n}} \tag{6}$$

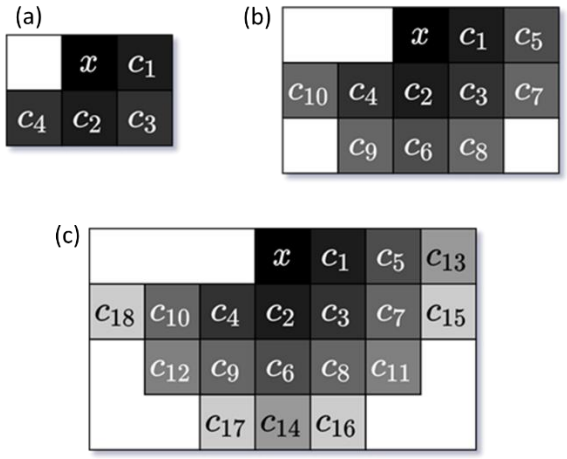


圖 7. 不同n下的參照區塊C_n

步驟 4：藏密

預測誤差直方圖 PEH 中所選定 bin 的最大藏密量$H(k,t)$可運用公式(7)，根據所選用之預測誤差（$k = P_e$）和複雜度閾值t計算產生，其中#為符合條件的像素數量，而N_S為平滑像素S_P的總數。運用公式(8)計算預測誤差 P_e，再以公式(9)嵌入各像素。

$$H(k,t) = \#\{1 \leq i \leq N_S : P_{e,i} = k, \Delta_i = t\} \tag{7}$$

$$P_e = \begin{cases} \begin{cases} p - c_{max} & if\ \ p \geq c_{max} \\ c_{min} - p & if\ \ p \leq c_{min} \\ skipped & if\ \ c_{min} < p < c_{max} \end{cases} & and\ \ c_{max} \neq c_{min} \\ \begin{cases} p - c_{max} & if\ \ p \geq c_{max} + 1 \\ c_{max} - p & if\ \ p \leq c_{max} \end{cases} & and\ \ c_{max} = c_{min} \end{cases} \tag{8}$$

$$p', P_e' = \begin{cases} \begin{cases} p+b, P_e+b & if\ \ P_e = p^*_{e,opt} \\ p+1, P_e+1 & if\ \ P_e > p^*_{e,opt} \quad and\ \ p \geq c_{max} \\ p, P_e & if\ \ P_e < p^*_{e,opt} \end{cases} \\ \begin{cases} p-b, P_e-b & if\ \ P_e = p^*_{e,opt} \\ p-1, P_e-1 & if\ \ P_e > p^*_{e,opt} \quad and\ \ p \leq c_{min} \\ p, P_e & if\ \ P_e < p^*_{e,opt} \end{cases} \\ skipped \qquad\qquad if\ \ c_{min} < p < c_{max} \end{cases}$$

$$where\ \ c_{max} \neq c_{min} \tag{9}$$

$$p', P_e' = \begin{cases} \begin{cases} p+b, P_e+b & if\ \ P_e = p^*_{e,opt} \\ p+1, P_e+1 & if\ \ P_e > p^*_{e,opt} \quad and\ \ p > c_{max} \\ p, P_e & if\ \ P_e < p^*_{e,opt} \end{cases} \\ \begin{cases} p-b, P_e-b & if\ \ P_e = p^*_{e,opt} \\ p-1, P_e-1 & if\ \ P_e > p^*_{e,opt} \quad and\ \ p \leq c_{min} \\ p, P_e & if\ \ P_e < p^*_{e,opt} \end{cases} \end{cases}$$

$$where\ \ c_{max} = c_{min}$$

2.4 改良式 PPVO

相較於 Weng 等人[12]與 Qu 和 Kim[10]提出的 PPVO，Wu 等人[13]的改良式 PPVO 將參考區塊擴大（如圖 7 所示），可以在低容量時更全面地參考周圍的像素，使預測結果更為精確，以降低不必要的位移。並且因為能更準確地計算預測誤差（PE），使得其直方圖（PEH）更為銳利，可以在 Top bin 嵌入更多的秘密訊息，換言之，可在品質不變的狀況下增加容量。此方法還可以依據像素所在區塊之複雜度，適時調整實際進行藏密時使用的參照區塊大小。

作者將像素以編碼時使用之參照區塊大小，區分為3類：$\{C_4, C_{10}, C_{18}\}$（如圖 7 所示），亦即 3 類所使用之像素數量，分別為：$(n_1, n_2, n_3) = (4,10,18)$。並以多個閾值$T = (t_1, t_2, t_3)$，依照所在之區塊複雜度 LC 加以區分，其中 T 滿足$0 \leq t_1 \leq t_2 \leq t_3$。區塊複雜度 LC 是依照$[t_{i-1}, t_i)$的範圍歸入第 i 類，並採取C_i作為嵌入時的參照區塊，而$LC > t_3$的像素則完全捨棄，從藏密過程忽略。採取 3 種參照區塊大小，對於嵌入效果而言已經相當足夠，如果取到 4 種以上參照區塊大小，對於藏密後的影像品質提昇並沒有明顯的幫助。而若以窮舉的方式，來尋找閾值t_i的時間開銷，將隨著 k 的增加呈平方增長，即便能提昇影像品質，卻並不見得划得來。

三、研究方法

3.1 藏密程序

本研究所提出之方法，主要是結合 Weng 等人[12]提出的最佳化 PPVO 方法（採用環繞交錯式藏密方式），以及 Wu 等人[13]提出之改良式 PPVO 方法（採用擴大參照區塊的方式）兩者之優點，目標在提升藏密後影像之品質，以減低失真程度（以 PSNR 作為比較基準），同時能維持 Weng 等人[12]所提出之方法對於較粗糙圖片（如：USC-SIPI 影像資料庫中之 Baboon 載體）的預測效果。

具體的藏密流程（如圖 8 及圖 9 所示），各步驟說明如下：

步驟 1：拆分影像

參考 Weng 等人[12]的方法，先將載體影像I_C以類似西洋棋盤的方式，拆分成I_A和I_B兩組交錯的像素集合（其排列樣式如圖 10 所示）。

步驟 2：複雜度計算及參照區塊大小調整

假設此時正在參與藏密的集合為I_A，則以待編碼像素（即待預測像素 x）為基準，如圖 11 所示，依由近而遠、由左而右、由上而下的順序選取參照區塊所用的像素，再以參照區塊的最大可能的像素個數CN（此處固定$CN = 22$），根據公式(10)計算像素的區塊複雜度Δ，其中 $\mu_{LSE} = I_{LSE}$ 的平均值，$\sigma_{LSE} = I_{LSE}$ 的標準差，$I_{LSE} = n$ 個臨近像素的集合。

$$\Delta = \sigma_{LSE} = \sqrt{\frac{\sum_{t \in \{1,n\}} (c_t - \mu_{LSE})^2}{CN}} \tag{10}$$

參考 Wu 等人[13]的方法，採用m個閾值$T = (t_1, t_2, \ldots, t_m)$來篩選各區塊複雜度Δ（此處設定$m = 2$，則獲得 2 個閾值$(t_1, t_2)$，且$0 \leq t_1 \leq t_2$）。後續依照這些閾值，來判斷各個參照區塊藏密時實際使用的大小。各區塊分別使用CN_i個像素，並歸類為第 i 類的參照區塊像素集合 A_i，其中$i \in [1, m]$。本研究所使用之參照區塊大小，如圖 12 所示（分別為$CN_1 = 4$ 與 $CN_2 = 11$）。

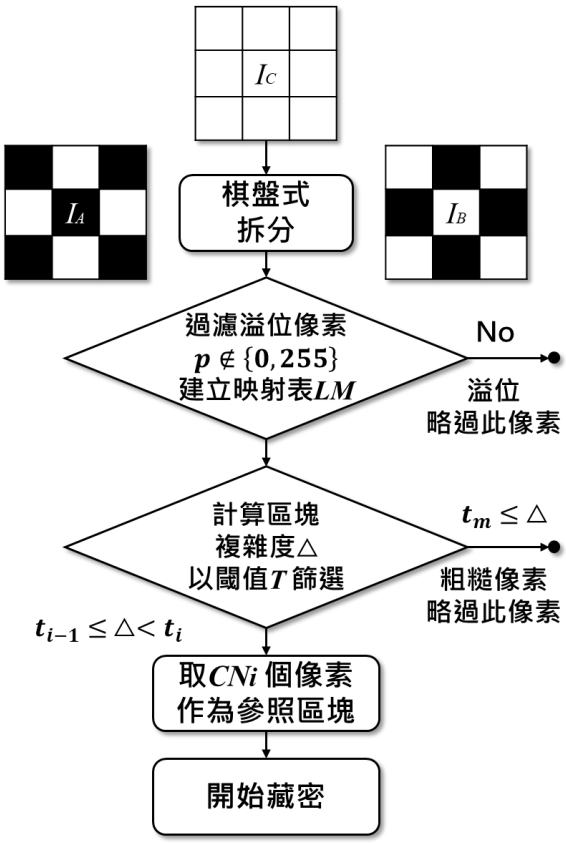


圖 8. 載體影像拆分與參照區塊調整

步驟 3：由預測誤差直方圖 PEH 中選擇最佳 bin 並藏密

各像素可根據所在區塊的複雜度Δ，透過所對應的閾值t_i，將其歸屬於第 i 類的像素集合，於藏密時將使用不同大小的參照區塊CN_i，其中$i \leq m$。

嵌入前，需先判斷I_A的 PEH 中預測誤差$P_{e,opt}$ bins 的預估容量$H\left(P_{e,opt}^{i}, t_i\right)$，並選擇滿足公式(11)的 bin。嵌入時，則各自根據 PEH bin 的品質（藏密前後之方均差 MSE），動態調整使用之 PEH bin，直到找到最適合的$P_{e,opt}^{*}$，完成此層的嵌入。

$$H(k,t) = \#\{1 \leq i \leq N_S : k = P_e, t = T\}$$
$$H\left(P_{e,opt}, T\right) \geq \text{剩餘密文長度} \qquad (11)$$

#為集合內元素數量，N_S為S_p的像素總數

步驟 4：還原資訊

當嵌入到各層的第L_{CLM}個像素後，先分別將I_A、I_B子圖起始的L^{Σ}個像素的 *LSBs* 保存成位元流 C，再以步驟三的方式接續進行嵌入（流程如圖 9）。而位元流 C 嵌入完畢後，再繼續嵌入秘密訊息，直到訊息結束或是容量不足為止。最後再分別於I_A、I_B子圖起始的L_{CLM}個像素，以 LSB 取代法嵌入還原訊息。

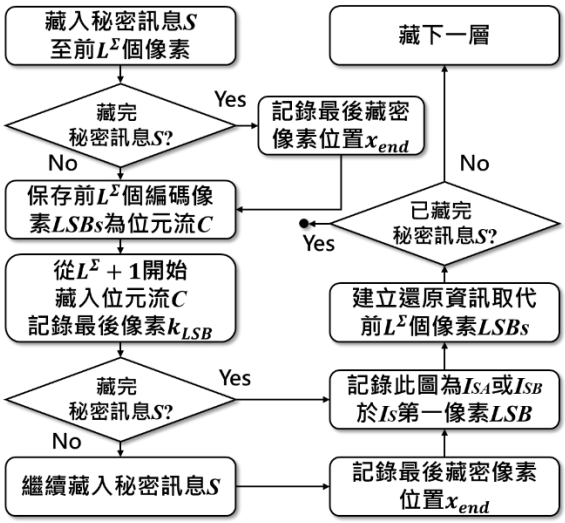


圖 9. 藏密流程

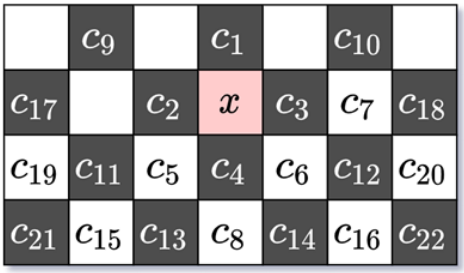


圖 10. 取樣區塊範圍及交錯樣式(完整的 *CN* = 22)

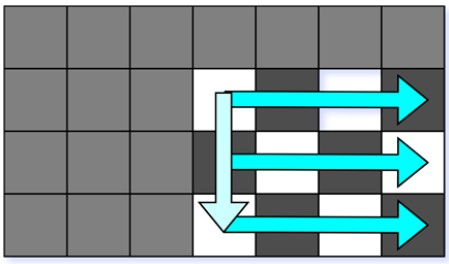


圖 11. 參照區塊像素的選取順序

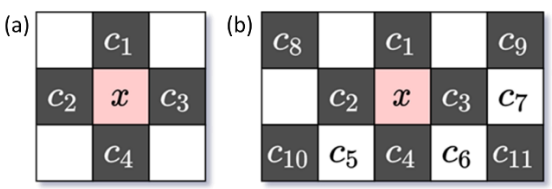


圖 12. 不同之參照區塊大小 *CN*（左圖為 *CN* = 4，右圖為 *CN* = 11）

個別像素的藏密流程如下：

步驟 1：根據溢位映射表LM，跳過溢位標記為 True 的像素。

步驟 2：以公式(10)計算像素 p 的區塊複雜度 Δ，根據篩選閾值t_i，決定所用參照區塊的大小 CN_i。

步驟 3：根據所設定之參照區塊大小，及所採用之 PEH bin，依據公式(12)計算預測差值P_e，並以公式(13)嵌入秘密訊息位元 b，得到修改後的像素值p'。

$$P_e = \begin{cases} \begin{cases} p - c_{max} & if\ \ p \geq c_{max} \\ c_{min} - p & if\ \ p \leq c_{min} \\ skipped & if\ \ c_{min} < p < c_{max} \end{cases} & and\ \ c_{max} \neq c_{min} \\ \begin{cases} p - c_{max} & if\ \ p \geq c_{max} + 1 \\ c_{max} - p & if\ \ p \leq c_{max} \end{cases} & and\ \ c_{max} = c_{min} \end{cases} \tag{12}$$

$$p', P_e' = \begin{cases} \begin{cases} p + b, P_e + b & if\ \ P_e = p_{e,opt}^* \\ p + 1, P_e + 1 & if\ \ P_e > p_{e,opt}^* \\ p, P_e & if\ \ P_e < p_{e,opt}^* \end{cases} & and\ \ p \geq c_{max} \\ \begin{cases} p - b, P_e - b & if\ \ P_e = p_{e,opt}^* \\ p - 1, P_e - 1 & if\ \ P_e > p_{e,opt}^* \\ p, P_e & if\ \ P_e < p_{e,opt}^* \end{cases} & and\ \ p \leq c_{min} \\ skipped & if\ \ c_{min} < p < c_{max} \end{cases} \quad where\ \ c_{max} \neq c_{min} \tag{13}$$

$$p', P_e' = \begin{cases} \begin{cases} p + b, P_e + b & if\ \ P_e = p_{e,opt}^* \\ p + 1, P_e + 1 & if\ \ P_e > p_{e,opt}^* \\ p, P_e & if\ \ P_e < p_{e,opt}^* \end{cases} & and\ \ p > c_{max} \\ \begin{cases} p - b, P_e - b & if\ \ P_e = p_{e,opt}^* \\ p - 1, P_e - 1 & if\ \ P_e > p_{e,opt}^* \\ p, P_e & if\ \ P_e < p_{e,opt}^* \end{cases} & and\ \ p \leq c_{min} \end{cases} \quad where\ \ c_{max} = c_{min}$$

3.2 取密程序

取密程序（如圖 13 與圖 14 所示），各步驟說明如下：

步驟 1：拆分子圖

首先，以西洋棋盤的樣式為基準，交錯拆分藏密影像I_S，成為I_{SA}、I_{SB}兩層次的子圖。從第一個像素 *LSB* 是否為 0，來判斷此層子圖所屬為I_{SA}或I_{SB}，再取出對應子圖各像素的 *LSBs*，來重建還原訊息，而得到溢位映射表和各個藏密參數。

步驟 2：取密

由第x_{end}個像素，反向取出並還原秘密訊息（取密順序示意，如圖 15），取出至第L^{Σ}個像素時，則暫時停止。所取出的這L^{Σ}個位元，即包含原始影像前端 *LSBs* 所組成的位元串流 C，故運用 LSB 取代法，對影像前端像素進行還原。接著從先前的位置繼續取密，直到第一個像素還原完成後結束。

步驟 3：決定是否繼續下一層取密

根據目前記錄的層數 *MT*，來判斷是否繼續往下一層進行取密。如果 *MT* 不為 0，即代表還有剩下的秘密訊息，則需回到步驟 1，繼續進行取密。

個別像素的取密流程如下：

步驟 1：根據取出的溢位映射表LM，跳過溢位標記為 True 的像素。

步驟 2：以公式(10)計算像素 p 的區塊複雜度 Δ，並根據篩選閾值t_i，來決定所使用的參照區塊大小CN_i。

步驟 3：根據所用之參照區塊大小，及所採用之 PEH bin，依據公式(14)計算預測差值P_e'，並以公式(15)取出秘密訊息位元 b，以還原原始像素值 p。

$$P_e' = \begin{cases} \begin{cases} p' - c_{max} & if\ \ p' \geq c_{max} \\ c_{min} - p' & if\ \ p' \leq c_{min} \\ skipped & if\ \ c_{min} < p' < c_{max} \end{cases} & where\ \ c_{max} \neq c_{min} \\ \begin{cases} p' - c_{max} & if\ \ p' \geq c_{max} + 1 \\ c_{max} - p' & if\ \ p' \leq c_{max} \end{cases} & where\ \ c_{max} = c_{min} \end{cases} \tag{14}$$

$$p, b = \begin{cases} \begin{cases} p', b = 0 & if\ \ P_e' = P_{e,opt} \\ p' - 1, b = 1 & if\ \ P_e' = P_{e,opt} + 1 \\ p' - 1, no\ \ data & if\ \ P_e' > P_{e,opt} \\ p', no\ \ data & if\ \ P_e' < P_{e,opt} \end{cases} & and\ \ p' \geq c_{max} \\ \begin{cases} p', b = 0 & if\ \ P_e' = P_{e,opt} \\ p' + 1, b = 1 & if\ \ P_e' = P_{e,opt} - 1 \\ p' + 1, no\ \ data & if\ \ P_e' > P_{e,opt} \\ p', no\ \ data & if\ \ P_e' < P_{e,opt} \end{cases} & and\ \ p' \leq c_{min} \end{cases} \quad where\ \ c_{max} \neq c_{min} \tag{15}$$

$$p, b = \begin{cases} \begin{cases} p', b = 0 & if\ \ P_e' = P_{e,opt} \\ p' - 1, b = 1 & if\ \ P_e' = P_{e,opt} + 1 \\ p' - 1, no\ \ data & if\ \ P_e' > P_{e,opt} \\ p', no\ \ data & if\ \ P_e' < P_{e,opt} \end{cases} & and\ \ p' > c_{max} \\ \begin{cases} p', b = 0 & if\ \ P_e' = P_{e,opt} \\ p' + 1, b = 1 & if\ \ P_e' = P_{e,opt} - 1 \\ p' + 1, no\ \ data & if\ \ P_e' > P_{e,opt} \\ p', no\ \ data & if\ \ P_e' < P_{e,opt} \end{cases} & and\ \ p' \leq c_{ax} \end{cases} \quad where\ \ c_{max} = c_{min}$$

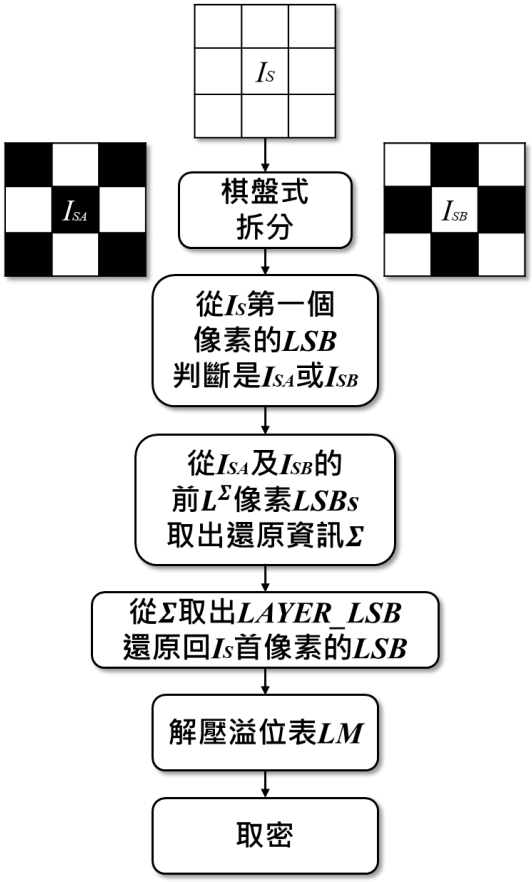


圖 13. 取密流程(一)

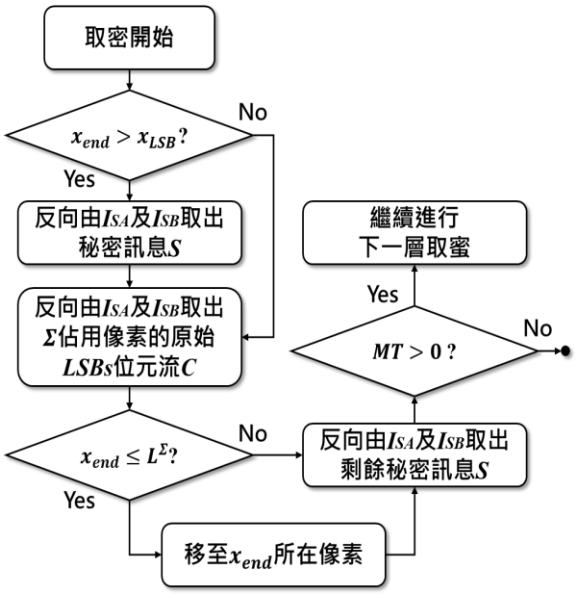


圖 14. 取密流程(二)

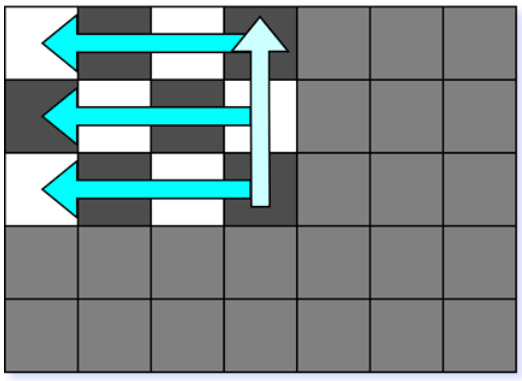


圖 15. 像素的取密順序

3.3 藏密、取密範例說明

令區塊複雜度的閾值T分別為 $(t_1, t_2) = (5,10)$，參照區塊大小CN分別為$(CN_1, CN_2) = (11, 22)$，而密文位元為$b = 1$。

假設待編碼像素 p=177，則其周邊的像素如圖 16 所示。除待編碼像素外，圖中標示像素值之其餘 22 個像素，將根據公式(10)，用於計算區塊複雜度 Δ。其中，完整參照區塊 I_{LSE}=(179,179,…,179,178)，而參照區塊平均值 μ_{LSE}=178.86，計算出本像素的區塊複雜度為 Δ=1.66。由於 $0 \leq \Delta \leq t_1$，故本像素屬於第 1 類像素，應使用 CN_1=11 來進行藏密、預測差值計算及嵌入（如圖 16 紅色粗框部分）。

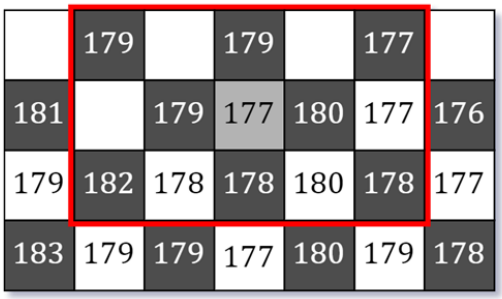


圖 16. 藏密前測試區塊

於參照區塊 c 之中，$c_{max} = 182$，而 $c_{min} = 177$，則根據公式(12)，$c_{max} \neq c_{min}$且 $p \leq c_{min}$，計算出預測差值為$P_e = c_{min} - p = 177 - 177 = 0$。假設此時此層$P_{e,opt}^{1*} = 0$，則根據公式(13)，$P_e = P_{e,opt}^{*}$且$p \leq c_{min}$，計算出 $p' = p - b = 177 - 1 = 176$。此時結果如圖 17 所示，待嵌入像素變為 176。

最後將所使用之(t_1, t_2)、(CN_1, CN_2)、$P_{e,opt}^{*}$皆嵌入載體影像後，則完成藏密。

	179		179		177	
181		179	176	180	177	176
179	182	178	178	180	178	177
183	179	179	177	180	179	178

圖 17. 藏密後測試區塊

取密流程也以相同區塊來進行說明。首先，從還原資訊取出所使用的參數(t_1, t_2)、(CN_1, CN_2)、$P^*_{e,opt}$。

於圖 17 中，根據公式(10)來計算區塊複雜度為Δ。其中，$I_{LSE} = (179,179,\ldots,179,178)$，而$\mu_{LSE} = 178.86$，故本像素的區塊複雜度為$\Delta = 1.66$。而由於$0 \le \Delta \le t_1$，則判斷本像素屬於第 1 類像素，應使用$CN_1 = 11$來進行取密（如圖 17 之紅色粗框部分）。

於參照區塊 c 之中，$c_{max} = 182$，而$c_{min} = 177$，則根據公式(14)，$c_{max} \neq c_{min}$且$p' \le c_{min}$，計算出預測差值為$P'_e = c_{min} - p' = 177 - 176 = 1$。最後根據公式(15)，$P'_e = P^{1*}_{e,opt} - 1$且$p' \le c_{min}$，計算出$(p, b) = (p' + 1,1) = (176 + 1,1) = (177,1)$，故秘密訊息位元$b = 1$，而原始像素$p = 177$，完成取密。

四、實驗結果

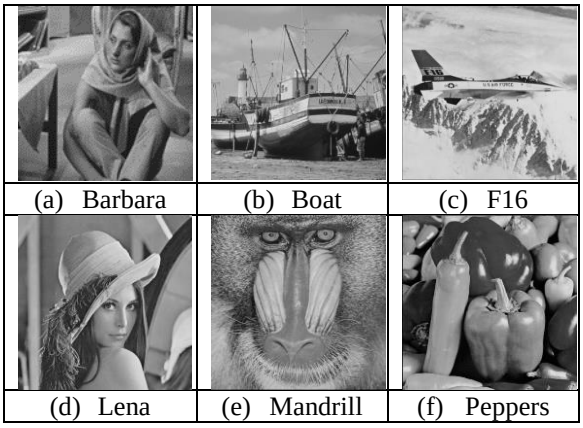
本節將針對本研究提出的基於像素式像素排序法之可回復式資訊隱藏技術進行分析，並藉由得到的實驗數據，來驗證本方法之可行性。此處針對 5,000-15,000 位元範圍下的低藏密量，進行固定容量藏密，並以峰值訊號雜訊比（PSNR）為指標，和其他學者提出方法之藏密影像進行品質比較。以下內容，首先陳述本研究進行之實驗環境，接續為實驗結果分析。

4.1 實驗環境

本研究採用之相關軟、硬體環境，簡述如下：

- 硬體環境：所使用之筆記型電腦，其中央處理器為具備四核心之 Core i7 6700HQ 2.6 GHz，並搭配 24GB 記憶體。主要用來進行藏密實驗程式開發及影像編碼。
- 軟體環境：所採用之作業系統為 Ubuntu 20.04，並運用 Visual Studio Code 整合開發環境，以 GCC 作為編譯器，以 C++進行程式編寫，並藉由 OpenCV 及 Eigen 所提供之函數，來實作原始 PPVO 藏密法及本研究提出之藏密方法，以進行藏密、取密及影像還原比較。

本研究使用影像處理領域常用的 USC-SIPI 影像資料庫[14]中之 Boat 等 6 張512 × 512灰階影像（如圖 18），來作為初步測試和比對目標。後續，以 BOSSBase 自然影像資料庫[15]中之影像為主，來進行廣泛性測試。其中包含具備複雜紋理的影像，也有相對平滑的影像，並於分類上包含人像、物體、風景等，可用來代表常見數位影像的內容。而在影像藏密後品質上的衡量標準，則主要是以峰值訊號雜訊比（PSNR）值作為比較依據。



(a) Barbara (b) Boat (c) F16
(d) Lena (e) Mandrill (f) Peppers

圖 18. 本研究使用之測試影像

4.2 結果分析

依照本研究提出之方法（藏密步驟如 3.1 節所述），採用 USC-SIPI 影像資料庫[14]中之 Boat 等 6 張 512x512 灰階影像（如圖 18），就藏密後之影像品質，與 Li 等人[6]的原始 PVO、Qu 和 Kim[10]的原始 PPVO、Weng 等人[12]的固定大小交錯式 PPVO，以及 Wu 等人[13]的變動參照區塊大小 PPVO 等方法進行初步比較，所獲得之數據（PSNR）依不同藏密量與載體影像區分，如表 2 至表 6 所示。另外，整體平均表現如表 7 與圖 19 所示。

由此初步實驗結果得知，和 Li 等人[6]、Qu 和 Kim[10]、Weng 等人[12]，以及 Wu 等人[13]的方法相比，本研究方法在藏密量約為 10000 位元，且估計是在第一層藏密即將結束時，具有最好的表現。而至於其他藏密量區間，則穩定保持與 Weng 等人[12]的方法有相當的品質表現。

表 2. 採用 USC-SIPI 影像資料庫 6 張標準影像於 5000 位元藏密量之 PSNR（dB）表現比較

作者 / 圖片	Li 等人[6]	Qu 和 Kim [10]	Weng 等人 [12]	Wu 等人 [13]	本研究
Barbara	N/A	64.0	65.0	64.0	64.3
Boat	61.7	62.5	62.5	63.0	65.2
F16	65.4	67.0	66.2	68.0	65.8
Lena	63.4	64.0	64.5	65.0	64.0
Mandrill	58.6	59.0	59.9	59.0	58.7
Peppers	62.3	62.5	63.0	63.0	62.9
平均	62.3	63.2	63.5	63.7	63.5

表 3. 採用 USC-SIPI 影像資料庫 6 張標準影像於 7500 位元藏密量之 PSNR（dB）表現比較

作者 / 圖片	Li 等人[6]	Qu 和 Kim [10]	Weng 等人 [12]	Wu 等人 [13]	本研究
Barbara	N/A	62.0	63.1	62.3	62.4
Boat	59.45	60.4	60.8	60.9	63.2
F16	63.1	65.3	65.1	66.0	64.4
Lena	61.3	62.2	62.7	63.0	61.8
Mandrill	55.8	56.6	57.7	56.7	56.4
Peppers	60.2	60.8	61.3	61.3	60.9
平均	60.0	61.2	61.8	61.7	61.5

表 4. 採用 USC-SIPI 影像資料庫 6 張標準影像於 10000 位元藏密量之 PSNR（dB）表現比較

作者 / 圖片	Li 等人[6]	Qu 和 Kim [10]	Weng 等人 [12]	Wu 等人 [13]	本研究
Barbara	N/A	60.0	61.1	60.5	60.7
Boat	58.0	58.2	59.0	58.8	61.7
F16	61.7	63.5	64.0	64.0	62.8
Lena	59.9	60.4	60.9	61.0	60.4
Mandrill	53.6	54.2	55.4	54.4	54.7
Peppers	58.5	59.0	59.5	59.5	59.4
平均	58.3	59.2	60.0	59.7	60.0

表 5. 採用 USC-SIPI 影像資料庫 6 張標準影像於 12500 位元藏密量之 PSNR（dB）表現比較

作者 / 圖片	Li 等人[6]	Qu 和 Kim [10]	Weng 等人 [12]	Wu 等人 [13]	本研究
Barbara	N/A	58.7	59.8	59.3	59.6
Boat	56.4	57.2	57.8	57.5	60.5
F16	60.8	62.7	63.0	63.2	61.8
Lena	58.6	59.4	59.9	60.1	59.4
Mandrill	51.8	52.3	54.0	Out of space	52.9
Peppers	57.4	57.8	58.3	58.3	58.4
平均	57.0	58.0	58.8	59.6	58.8

表 6. 採用 USC-SIPI 影像資料庫 6 張標準影像於 15000 位元藏密量之 PSNR（dB）表現比較

作者 / 圖片	Li 等人[6]	Qu 和 Kim [10]	Weng 等人 [12]	Wu 等人 [13]	本研究
Barbara	N/A	57.7	58.6	58.5	58.6
Boat	55.4	56.0	56.5	56.2	59.5
F16	59.4	61.8	62.0	62.3	60.8
Lena	57.6	58.3	58.7	59.1	58.4
Mandrill	Out of space	50.7	53.0	Out of space	51.0
Peppers	56.2	56.4	57.1	57.1	57.3
平均	57.1	56.8	57.7	58.6	57.6

表 7. USC-SIPI 資料庫 6 張標準影像於 5000-15000 位元藏密量之平均 PSNR（dB）表現比較

作者 / 藏密量	Li 等人[6]	Qu 和 Kim [10]	Weng 等人 [12]	Wu 等人 [13]	本研究
5000	62.3	63.2	63.5	63.7	63.5
7500	60.0	61.2	61.8	61.7	61.5
10000	58.3	59.2	60.0	59.7	60.0
12500	57.0	58.0	58.8	59.6	58.8
15000	57.1	56.8	57.7	58.6	57.6

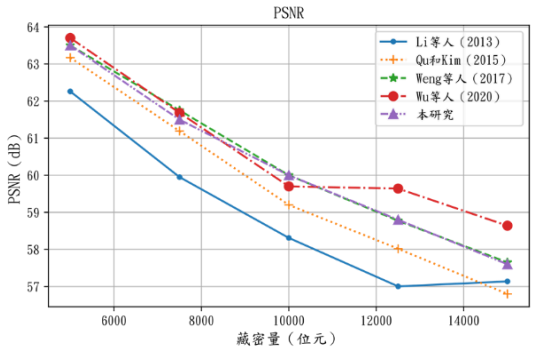


圖 19.採用 USC-SIPI 影像資料庫 6 張標準影像於 5000-15000 位元藏密量之平均 PSNR（dB）表現比較

本研究另外採用了大量影像，來進行廣泛性測試，並與其他學者所提出之方法進行比較。所使用之影像，是從知名的 BOSSBase 自然影像資料庫中，隨機選取的 3850 張 512x512 灰階影像。測試結果如圖 20 所示，相較於 Li 等人[6]、Qu 和 Kim[10]、Weng 等人[12]，以及 Wu 等人[13]的方法所產出之結果，普遍有較好的品質表現。於本測試中，由於缺少 Wu 等人[13]和 Weng 等人[12]兩種方法的原始程式碼，而依據其演算法所自行實作之程式，並沒有辦法達到與其文獻呈現完全相同效果，故本研究針對這兩種方法之結果，是直接採用其文獻中所宣稱之最佳數據，來進行比較。而由於本研究所採取之圖片數量更多，且種類範圍更廣，故可合理認定本研究進行之廣泛性測試，所產出之結果更為客觀，且更具參考價值。且即便廣泛測試之結果，相較於只採用少數影像測試之最佳結果，可能會有低估性能之情形，然而在此情形下，本研究之結果仍能優於這兩種方法所提出之數據。

另外，本研究方法與 Li 等人[6]及 Qu 和 Kim[10]方法之比較結果，如圖 21、表 8 及表 9 所示。就峰值訊號訊比（PSNR）而言，本研究方法於 5000-15000 位元藏密量下，相較於 Li 等人[6]之方法，分別提升了 6.34%、6.09%、6.05%、6.10%、6.16%，而相較於 Qu 和 Kim[10]之方法，則分別提升了 7.12%、6.40%、5.90%、5.51%、5.16%。就結構相似性（SSIM）而言，相較於 Li 等人[6]之方法，分別提升了 0.0122%、0.0135%、0.0148%、0.0158%、0.0168%，而相較於 Qu 和 Kim[10]之方法，則分別提升了 0.0307%、0.0321%、0.0333%、0.0339%、0.0343%。

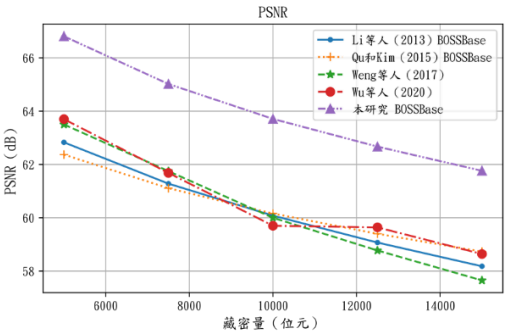


圖 20.採用 BOSSBase 影像資料庫 3850 張隨機影像於 5000-15000 位元藏密量之平均 PSNR（dB）表現比較

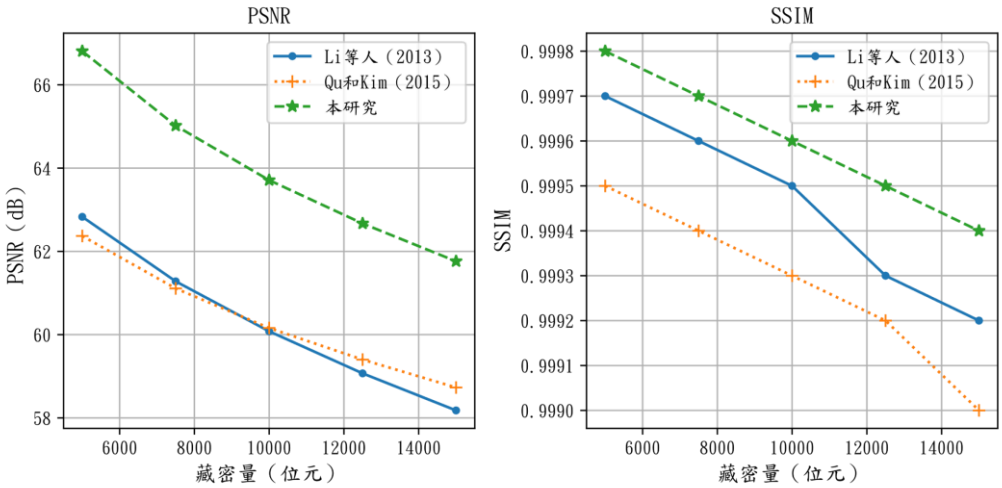


圖 21.採用 BOSSBase 3850 張隨機影像於 5000-15000 位元藏密量之平均 PSNR 與 SSIM

表 8.採用 BOSSBase 3850 張隨機影像於 5000-15000 位元藏密量之平均 PSNR（dB）結果比較

作者 / 藏密量	本研究	Li 等人（2013）	Qu 和 Kim（2015）
5000	66.81	62.83	62.37
7500	65.02	61.28	61.11
10000	63.71	60.08	60.16
12500	62.67	59.07	59.40
15000	61.76	58.18	58.73

表 9.採用 BOSSBase 3850 張隨機影像於 5000-15000 位元藏密量之 SSIM 結果比較

作者 / 藏密量	本研究	Li 等人（2013）	Qu 和 Kim（2015）
5000	0.9998	0.9997	0.9995
7500	0.9997	0.9996	0.9994
10000	0.9996	0.9995	0.9993
12500	0.9995	0.9993	0.9992
15000	0.9994	0.9992	0.9990

針對 BOSSBase 自然影像資料庫隨機 3850 張影像與 USC-SIPI 影像資料庫 6 張標準影像（如圖 18），運用本研究方法所進行之藏密影像品質比較，結果如圖 22 所示。顯示本研究方法在 5000-15000 位元藏密容量下，廣泛測試的平均表現，優於其他文獻中所特別挑選出的 6 張影像（在各藏密容量下，影像品質分別有 5.210%、5.721%、6.183%、6.581%、7.227%的提升）。也代表本方法雖與文獻中其他方法在相對侷限的 6 張標準影像上的表現相當，然而在模擬實際運用的廣泛測試下，能有更良好穩定的表現。

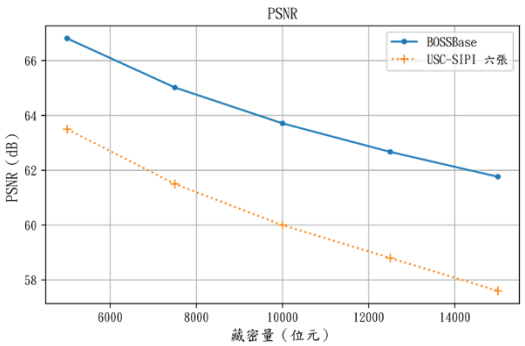


圖 22.本研究方法對 BOSSBase 資料庫 3850 張隨機影像與 USC-SIPI 資料庫 6 張標準影像於 5000-15000 位元藏密量下之結果比較

4.3 安全性分析

資訊隱藏的根本目標，是完全不被察覺。而除了不被輕易從肉眼觀察到失真與不自然的改變之外，也要能夠預防被常見的藏密分析技術檢查出異常狀況。本節主要探討本研究提出的藏密方法，是否能有效抵抗藏密分析技術的偵測，故採用了兩種具有代表性的空間域藏密分析方法，分別為正規/特異（RS）藏密分析，以及像素差值直方圖（PDH）藏密分析，來進行藏密影像偵測。

RS 藏密分析技術是由 Fridrich、Goljan 與 Du 三位學者於 2001 年提出。於本安全性分析中，採用了 RS 藏密分析技術，來針對 Boat 原始載體影像進行分析，所獲得之參考結果如圖 23 (a)所示。後續，分別針對使用 1-bit LSB 取代法及 3-bit LSB 取代法藏密的 Lena 影像進行分析，其結果如圖 23 (b)與(c)所示。最後，針對本研究藏密方法進行藏密的 Boat 影像（15,000 位元藏密量）進行分析，其結果如圖 23 (d)所示。由 RS 分析結果，指出各影像的藏密比率（Embedding Rate）分別為(a) -0.03、(b) 0.94、(c) 1.1 及(d) -0.03，由此判定(b)與(c)的影像有藏密，而(a)與(d)的影像無藏密。由此安全性分析結果，可得知 RS 偵測技術可有效偵測使用 LSB 取代法藏密之影像，但無法偵測本研究藏密方法產生的藏密影像，代表本研究藏密法可有效抵抗 RS 藏密分析技術之偵測。

一般灰階影像像素是以 8 個位元來表示，使得其值介於 0 到 255 的範圍，以表示灰色的深淺。像素差值直方圖（PDH），是用來統計影像相鄰像素間各種差值出現的次數。一般載體影像之 PDH 會接近常態分佈，如圖 24 (a)所示，其中橫軸為像素差值，而縱軸為像素差值的統計量（次數）。而另一方面，影像若經藏密演算法藏密後，則會改變像素間彼此的相關性，將使得其 PDH 會呈現不正常的分佈（例如：階梯狀），而不再接近常態分佈，如圖 24 (b)(c)(d)所示，此顯著的特徵容易成為藏密分析的偵測目標，用來判定該影像內確實隱藏秘密訊息。本研究針對原始像素差值（Pixel-Value Differencing, PVD）藏密法、3-bit LSB 取代法結合 PVD 藏密法、整合式 PVD/LSB 藏密法，以及本研究所提出之藏密方法（藏密量分別為 5,000 至 15,000 位元），分別進行藏密影像 PDH 安全性分析。

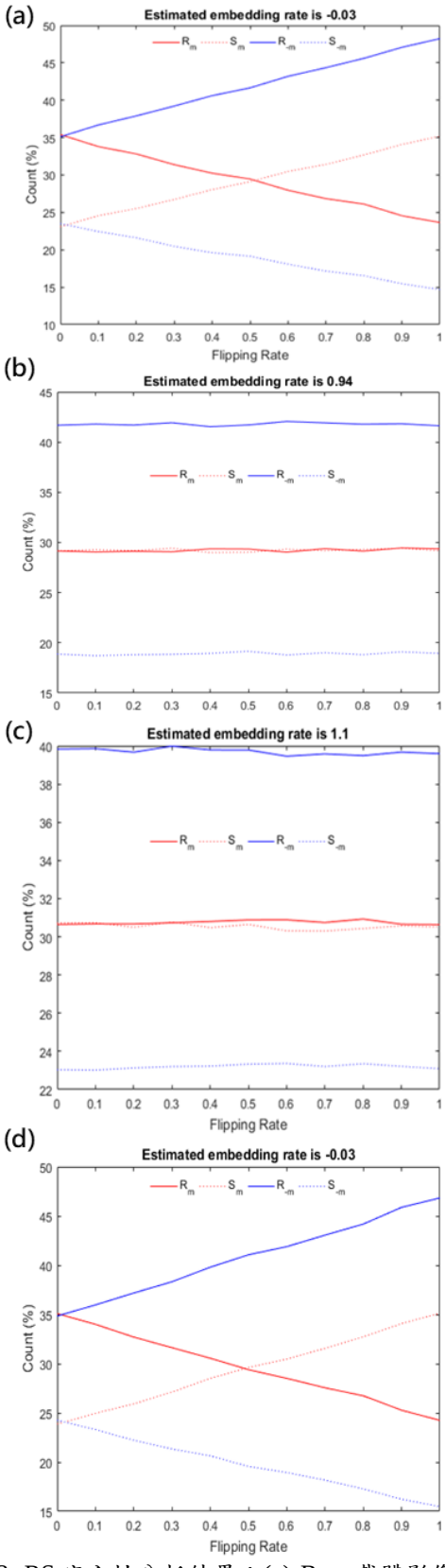


圖 23. RS 安全性分析結果：(a) Boat 載體影像 (b) 1-bit LSB 取代法 (c) 3-bit LSB 取代法 (d) 本研究藏密法

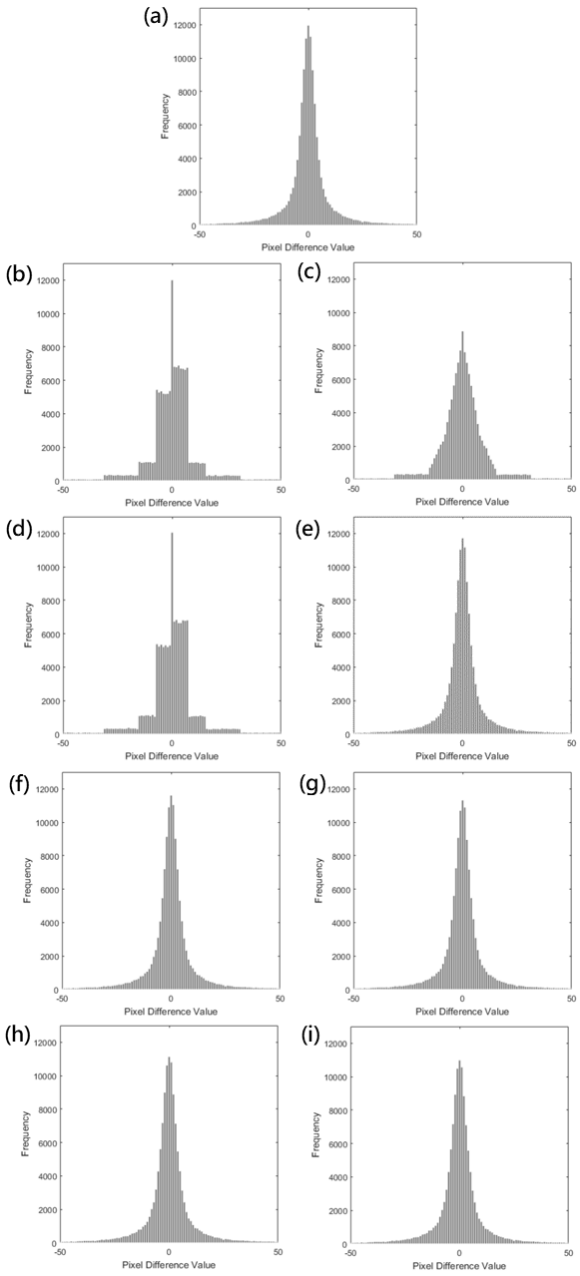


圖24.像素差值直方圖分析結果：(a) 載體影像 (b) 原始PVD藏密法 (c) 3-bit LSB取代法結合PVD藏密法 (d) 整合式PVD/LSB藏密法 (e) 本研究方法藏入5,000位元 (f) 本研究方法藏入7,500位元 (g) 本研究方法藏入10,000位元 (h) 本研究方法藏入12,500位元 (i) 本研究方法藏入15,000位元

此分析以 Boat 影像為例，載體影像的 PDH 如圖 24(a)所示，而運用前述各種方法藏密後，所產生之 PDHs 如圖 24 (b)~(i)所示。除了使用本研究所提出之藏密方法所產生之 PDHs 仍呈現常態分佈外（如圖 24 (e)~(i)），其餘皆明顯呈現不正常的階梯狀分佈，而這類特徵容易遭藏密分析方法成功偵測。由此可知，本研究所提出之藏密方法，能有效抵禦像素差值直方圖藏密分析偵測。

五、結論

本研究的目的，在於提升基於像素之像素排序法（Pixel-based Pixel Value Ordering, PPVO）於低容量下（5000~15000 位元）的藏密品質。此處以低容量為前提，主要是因為在確保圖片可還原的情形下，需保持圖片的冗餘特性，使得其藏密量和不可回復式藏密方法相比，具有先天的劣勢，故以低藏密量下的品質改進做為探討方向。

而依據此目的，主要的貢獻，是以 Weng 等學者於 2017 年提出的最佳化 PPVO 及 Wu 等學者於 2020 年提出的改良式 PPVO 為基礎，並結合兩者的優點，來建構出一套可回復式資訊隱藏方法。在測試與驗證上，先使用影像處理領域常用的 USC-SIPI 影像資料庫中之 Boat 等 6 張512 × 512灰階影像，來作為初步測試和比對目標，而後續再以 BOSSBase 自然影像資料庫中之影像為主，來進行廣泛性測試。由測試結果，說明了採用本方法所獲得之藏密影像品質，可普遍優於前述兩種方法，以及其他更原始之 PPVO 方法。其性能表現，也已透過各項測試來進行展示與驗證。

於現階段，本研究藏密方法已可提供動態選擇藏密預測誤差、依區塊複雜度調整參照區塊大小，以及調整複雜度閾值區間等功能。後續，可採用其他做法，如：動態更動參照區塊的樣式，甚至依照不同大小之參照區塊，來自動調整並採用更緊密的像素排列方式，同時考量其他可能有效提升影像品質之方式，來做為未來探究的方向。另外，本研究於有限期程內，並未針對所採用方法的回復性進行測試，將規劃納入後續的研究中做探討。

如果將本研究的藏密方法，結合公開金鑰密碼系統等方式保護秘密訊息，整合成完整的秘密通訊系統，可更有效達成可回復式資訊隱藏的隱蔽性。

參考文獻

[1] Kaur, G., Singh, S., Rani, R., and Kumar, R., “A Comprehensive Study of Reversible Data Hiding (RDH) Schemes Based on Pixel Value Ordering (PVO),” Archives of

Computational Methods in Engineering, Vol. 28, No. 5, pp. 3517-3568, 2020.

[2] Fridrich, J., Goljan, M., and Du, R., "Lossless Data Embedding—New Paradigm in Digital Watermarking," EURASIP Journal on Advances in Signal Processing, Vol. 2002, No. 2, 2002.

[3] Tian, J., "Reversible data embedding using a difference expansion," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 8, pp. 890-896, 2003.

[4] Ni, Z. C., Shi, Y. Q., Ansari, N., and Su, W., "Reversible data hiding," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 3, pp. 354-362, 2006.

[5] Thodi, D., and Rodriguez, J., "Expansion Embedding Techniques for Reversible Watermarking," IEEE Transactions on Image Processing, Vol. 16, No. 3, pp. 721-730, 2007.

[6] Li, X., Li, J., Li, B., and Yang, B., "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," Signal Processing, Vol. 93, No. 1, pp. 198-205, 2013.

[7] Peng, F., Li, X., and Yang, B., "Improved PVO-based reversible data hiding," Digital Signal Processing, Vol. 25, pp. 255-265, 2014.

[8] Ou, B., Li, X., Zhao, Y., and Ni, R., "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," Signal Processing: Image Communication, Vol. 29, No. 7, pp. 760-772, 2014.

[9] Wang, X., Ding, J., and Pei, Q., "A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition," Information Sciences, Vol. 310, pp. 16-35, 2015.

[10] Qu, X., and Kim, H., "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," Signal Processing, Vol. 111, pp. 249-260, 2015.

[11] Ou, B., Li, X., and Wang, J., "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion," Journal of Visual Communication and Image Representation, Vol. 39, pp. 12-23, 2016.

[12] Weng, S., Zhang, G., Pan, J., and Zhou, Z., "Optimal PPVO-based reversible data hiding," Journal of Visual Communication and Image Representation, Vol. 48, pp. 317-328, 2017.

[13] Wu, H., Li, X., Zhao, Y., and Ni, R., "Improved PPVO-based high-fidelity reversible data hiding," Signal Processing, Vol. 167, 107264, 2020.

[14] SIPI Image Database Volume 3: Miscellaneous. Available online at https://sipi.usc.edu/database/database.php?volume=misc [retrieved on April 5, 2022].

[15] BOSS Break Our Steganographic System, BOSSBases (v0.93). available online at http://agents.fel.cvut.cz/boss/index.php?mode=VIEW&tmpl = materials [retrieved on January 4,2022].