使用音訊檔隱藏社交圖像

姜博原 黃貞瑛*

輔仁大學資訊工程系

摘　要

由於資訊技術的進步，社交媒體使用率也大增，其中傳遞圖像更是常見的社交行動，不過在傳遞圖像時，其隱私性、保密性並不易保持。為了解決此一問題，本文提出以音訊檔做為掩護，將圖像壓縮並嵌入至音訊檔中，形成偽裝音訊檔，還原時只需解壓縮即可。圖像隱藏至音訊檔可分為兩大階段，第一階段為嵌入音訊檔。在嵌入音訊檔前，會對多張圖像作重組或並排處理，並且為了減少需要嵌入的位元數，會先對圖像做失真壓縮處理後，再嵌入音訊檔。第二階段為萃取回復圖像，會將隱藏位元取出，重組並解壓縮後便可回復圖像。實驗結果顯示，本論文提出的圖像合併方式，適合不同系列的多張圖像嵌入，並且在萃取回復圖像時也保有不錯的品質。

1. 緒論

隨著資訊技術的進步，傳遞社交媒體圖像之行為越來越普遍，但也由於駭客技術的提升，傳遞圖像時其私密性也不易被保持；舉例來說：想要透過網路將出遊的照片與好朋友分享時，在傳遞圖像的過程中可能就會被有心者竊取，雖然不是軍事、醫療等具有高度機密性的資料，一般人在傳遞資料的過程中也希望擁有一定程度的隱私性、安全性及保密性。

本文因此想要找出一種傳輸方法，使得在傳遞社交圖像的過程中，也可以保持其隱私性。而近年來由於智慧型裝置及多媒體檔案的普及，許多人會從網路上將音樂下載，然後儲存至智慧型裝置中收聽；也因為音訊檔容量較大，可以一次隱藏多張圖像而不易被發覺，所以我們想要以音訊檔做為掩護，將想要隱藏的圖像嵌入至音訊檔中，再將音訊檔傳遞給接收者，即使在傳遞的過程中被有心者竊取，也不易被察覺有圖像隱藏在其中，音訊檔成功傳遞給接收者時，再由接收者解壓縮並萃取出圖片。

在社交媒體的傳輸上，需要具有較高的傳輸速度，而一般社交圖像主要是分享生活圖像，因此在圖像品質之要求，只須符合人眼視覺即可。基於以上兩個特質：高傳輸速度及適度品質要求，因此在圖像傳輸部分採用了失真壓縮方法，藉以減少圖像大小，達到加快傳輸速度的目的。

* Corresponding author. Email: jihwang@csie.fju.edu.tw

本文以下各節分別為：第二節介紹音訊隱藏相關研究與音訊檔格式，第三節介紹研究方法，第四節為實驗方法、結果與討論，第五節為結論與未來展望。

2. 音訊處理與音訊檔

本文目的在於如何將隱私性高的個人相片隱藏至音訊檔中，避免在網際網路傳送與接收檔案時，被竊取的可能性。因此本節將先介紹音訊隱藏與處理，及 WAV 音訊檔。

2.1 音訊處理

資訊隱藏(Information Hiding)是一個通用術語，它涵蓋廣泛的以載體為媒介，將訊息內容嵌入某一掩護載體。早期資訊隱藏時會由圖片做為掩護，近年資訊的發達與進步，對於音訊處理的了解也更多[1]，使得音訊檔也逐漸可作為隱藏資訊的載體[2-12]。若以圖像與音訊分別作為載體時，圖像是由眼睛所觀察，而音訊是由耳朵所聆聽，後者被察覺有資訊隱藏的機率更小，且可隱藏空間也較大。近年來學者們為了使隱藏資料不被發現，便將資料隱藏至最低有效位元(Least Significant Bit, LSB)中[4-7][10,11]，Lavanya 等人將文字嵌入至音訊檔[8]，Zamani 等人則是在資料隱藏前以基因演算法運算後，降低隱藏資料被發現的風險[2]，而 Rimba 等人則是將浮水印嵌入，增強及保護音訊檔之所有權[12]。

常見的音訊檔案格式可分為三類：非壓縮音訊檔案格式，無失真壓縮音訊檔案格式及失真壓縮音訊檔案格式，以下分別描述其特性。

常見的非壓縮音訊檔格式計有 Waveform Audio Format(WAV) 及 Audio Interchange File Format(AIFF)。WAV 儲存的音訊未經過任何壓縮，因此在聲音方面不會出現失真的情況。也因為如此，檔案體積在眾多音訊格式中算得上是最大的，主要常見於 Windows 作業系統中。

無失真壓縮雖然縮小音訊檔的儲存容量，但可以保留原始檔案的所有資訊，在播放上與原始檔案沒有任何差別。壓縮技術可以從各方面評估，例如：壓縮速度、壓縮比率、解碼速度、軟體硬體支援、穩定性和出錯率。較為人所熟知的無失真壓縮音訊檔格式計有：Free Lossless Audio Codec (FLAC)、Monkey's audio (APE)、WavPack、Windows Media Audio Lossless (WMAL)、Apple Lossless Audio Codec (ALAC)、True Audio (TTA)、Tom's lossless Audio Kompressor (TAK)。其中市面上支援度最廣泛的無損音訊壓縮格式為 FLAC，因為 FLAC 的源碼是完全開放，也因此 FLAC 幾乎相容於所有作業系統平台。

失真壓縮音訊檔案主要將次要的資訊捨棄，犧牲部分品質以減少資料量，藉此提高壓縮比。MPEG-1 或 MPEG-2 Audio Layer III (MP3)為此種檔案格式中最廣為人知的格式。其目的是為了大幅降低資料量，使用了大量音訊刪減技術，其中包含利用心理聲學原理判斷何種音訊資料可以捨棄，將人耳不易察覺的聲音訊號移除，成功達成非常高的壓縮比例。標準 MPEG-1 或 MPEG-2 Audio Layer III

的副檔名是 .mp3。Windows Media Audio (WMA) 是微軟開發的數位音訊壓縮格式，WMA 格式屬於微軟私有，但是因為蘋果公司 iTunes 支援它，以致 WMA 格式目前成為 MP3 格式的競爭對手。

2.2 WAV 音訊檔

本文目的希望可以將多張圖像在不被發覺的情況下，由傳遞者傳送至接收者手中，因此先考慮的將會是載體的容量問題，並且希望能夠有工具對此音訊檔做處理，最後選擇的是 WAV 檔。而 WAV 檔除了檔案容量大以及有工具支援外，本身也屬於無失真壓縮的音訊檔，就算在音訊檔有被修改的情形下，修改過後的失真情況也不會比失真壓縮的音訊檔嚴重。嵌入過程中使用的方法，是可以將資料數據進行失真壓縮的離散餘弦變換，如此一來便能夠減少嵌入的數值量，達到增加嵌入圖片張數的目的。

WAV 檔不僅為無失真壓縮音訊檔，且本身的容量也夠大，也有 Python 支援處理。WAV 之中以數個 Chunk 組成，包括 RIFF Chunk、Format Chunk、Fact Chunk、Data Chunk 四種 Chunk，如圖一所示。

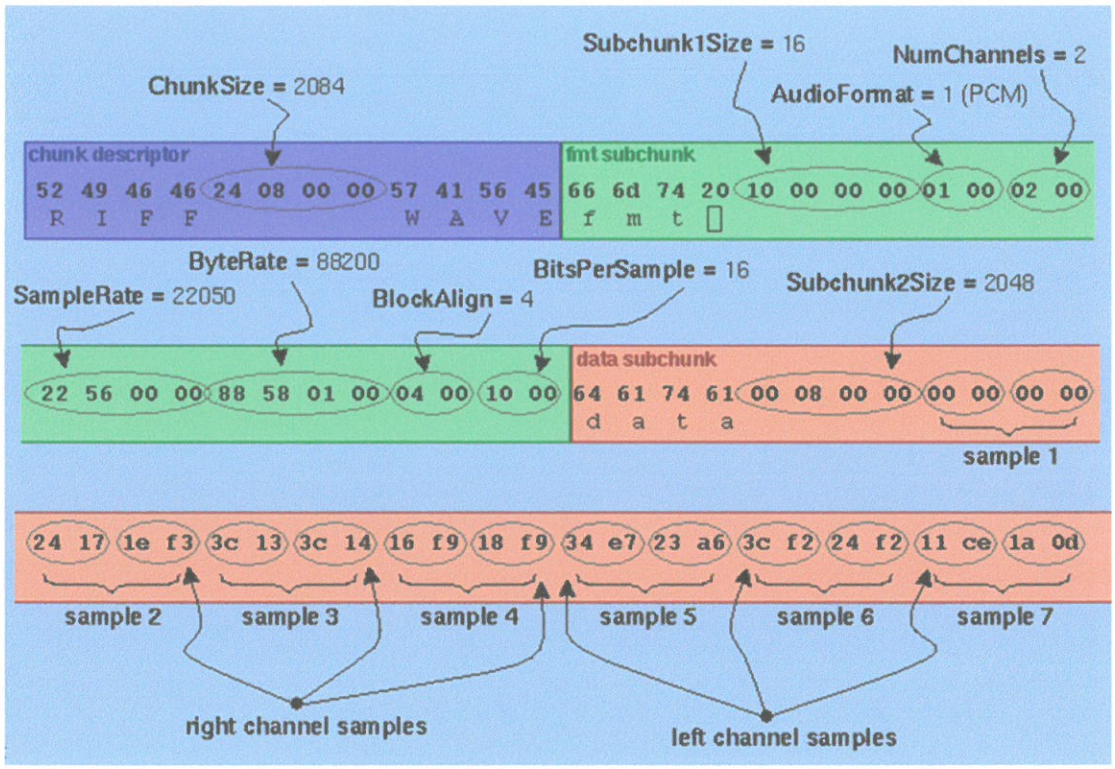


圖 一 WAV 音訊檔案規格 [13]

(A) RIFF Chunk：包含了 RIFF 標籤，Chunk 大小的資訊，WAV 標籤。

(B) Format Chunk：包含了音訊檔的各種屬性，首先是 fmt 標籤，Chunk 大小的資訊，音訊的編碼方式，單/雙聲道資訊，音訊採樣頻率，音訊傳送頻率，每次採樣的大小，每個聲道的採樣精確度。

(C) Fact Chunk：是用來記錄數據解壓縮後的大小(非紀錄檔案大小)，此 Chunk 的數量不一定，但非 PCM 格式的檔案至少會多加入一個 Fact Chunk，通常會加在 Data Chunk 的前面。

(D) Data Chunk：紀錄 WAV 實際資料的地方，一開始是 data 標籤，然後是音訊長度的大小，最後是音訊的數據，也是本論文嵌入方法會接觸到的部分。

3. 研究方法

本節將介紹如何使用音訊檔隱藏圖像，其過程可分為兩個階段，第一階段為嵌入，將圖像壓縮，並把壓縮位元嵌入至音訊檔中；第二階段萃取，將隱藏位元由音訊檔萃取後，重組萃取位元並解壓縮，便能夠得到回復圖像。本文於 3.1 節介紹單張圖像嵌入演算法，3.2 節介紹單張圖像萃取回復演算法，3.3 節介紹多張圖像嵌入與萃取演算法。

3.1 單張圖像嵌入演算法

在圖像隱藏至音訊檔之前，為減少隱藏容量，以加速嵌入時間，會先對圖像做壓縮。以下為嵌入之詳細步驟：

步驟一：轉換色彩空間

將表示圖像顏色的 RGB 格式轉為以亮度及色度的表現方式，由於人類眼睛的習性對於色度較不敏感，轉換後可減少色度的份量；以下 Y 代表亮度，Cb、Cr 則代表色度，以 ITU-R BT.601 格式轉換：

Y =16 + (66*R+129*G+25*B+128)/256

Cb = 128 + (-38*R-74*G+112*B+128)/256

Cr = 128 + (112*R – 94*G -18*B + 128)/256

步驟二：DCT 轉換

首先將圖像分成多個 8*8 區塊，以 8*8 矩陣儲存。而在將 8*8 矩陣代入離散餘弦變換(Discrete Cosine Transform, DCT)運算前，會先將每個數值減去 128，使其在代入 DCT 後的數字範圍會位於(-128~127)*8 之間，之後將圖像數值以 DCT 運算後以頻率的方式儲存，以下是 2D-DCT 的公式，$x_{n1,n2}$表示轉換前位於第n_1列第n_2行之原數值，$X_{k1,k2}$表示轉換後位於第k_1列第k_2行之頻率係數值。

$$X_{k1,k2} = \frac{1}{4}C(k_1)C(k_2)\sum_{n_1=0}^{7}\sum_{n_2=0}^{7} x_{n1,n2}\cos\frac{(2n_1+1)k_1\pi}{16}\cos\frac{(2n_2+1)k_2\pi}{16}$$

其中 $C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & otherwise \end{cases}$

步驟三：量化

經過 2D-DCT 後，數值便會以頻率係數儲存。最後會進行量化，將頻率係數除以量化矩陣的值，Y 的 8*8 矩陣除以亮度量化表(見表一)，Cb、Cr 的 8*8 矩陣則除以色度量化表(見表二)，取與商數最近的整數，目的是將頻率係數由浮點數轉為整數，以下是亮度量化表及色度量化表：

表 一 亮度量化表

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

表 二 色度量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

亮度與色度量化表是依據心理視覺製作，以濾除高頻能量，由於人眼對於低頻敏感度遠高於高頻，所以處理後的視覺損失不大。

步驟四：ZigZag 排序

量化後的矩陣將會出現大量的 0，而非 0 的數字則會集中在矩陣左上角，若是將矩陣以逐列的方式讀取數字，那麼可能會存取到大量的 0，為了減低儲存所需要的空間，會以 ZigZag 編碼法來儲存數字。表三 為 ZigZag 編碼方式，表中之數值表示拜訪順序：

表 三 ZigZag 拜訪表

1	3	4	10	11	21	22	36
2	5	9	12	20	23	35	37
6	8	13	19	24	34	38	49
7	14	18	25	33	39	48	50
15	17	26	32	40	47	51	58
16	27	31	41	46	52	57	59
28	30	42	45	53	56	60	63
29	43	44	54	55	61	62	64

本方法以 ZigZag 順序拜訪時，若是拜訪到連續兩排為 0 的數字，那麼拜訪便會停在最後一排不為 0 的位置，例如：若第四排及第五排之對角線數值全為 0，即表中之拜訪順序為第 7 至第 15 之值均為 0(見表四之陰影部分)，最後只會儲存非零排數 3，及拜訪順序第 1 第至 6 的數值，以一維陣列的方式表示，稱為 Z 數列。至此，嵌入前的壓縮已經完成，接下來便會將 Z 數列嵌入至音訊檔中。

表 四 ZigZag 拜訪範例圖

1	3	4	10	11	21	22	36
2	5	9	12	20	23	35	37
6	8	13	19	24	34	38	49
7	14	18	25	33	39	48	50
15	17	26	32	40	47	51	58
16	27	31	41	46	52	57	59
28	30	42	45	53	56	60	63
29	43	44	54	55	61	62	64

步驟五：嵌入

本實驗使用的音訊檔為雙聲道，採樣數每秒 44100 次，每個採樣為 16 位元的音訊檔，另外每個樣本擁有左聲道及右聲道，且左聲道及右聲道大小各為 16 位元，本文將以 $S_0,S_1,\ldots,S_{p-1}$ 表示各採樣時間點資料，p 為總採樣數，每一樣本點 S_i，i=0,1,…,p-1，又可分為左聲道樣本 $S_{i,0}$ 及右聲道樣本 $S_{i,1}$，每樣本之左右聲道各含 16 位元，以 $S_{i,0} = (s_{i,0}^{15}s_{i,0}^{14}\cdots s_{i,0}^{2}s_{i,0}^{1}s_{i,0}^{0})_2$ 及 $S_{i,1} = (s_{i,1}^{15}s_{i,1}^{14}\cdots s_{i,1}^{2}s_{i,1}^{1}s_{i,1}^{0})_2$ 表示。嵌入音訊檔的第一步，會先將圖像的長 H、寬 W 與維度 D 分別嵌入於 $S_{0,0}$，$S_{0,1}$ 及 $S_{1,0}$ 並分別取代為 $S'_{0,0}$，$S'_{0,1}$ 及 $S'_{1,0}$。換言之，$S'_{0,0}$，$S'_{0,1}$ 及 $S'_{1,0}$ 所儲存值為 H、W 與 D，而非原聲道樣本值。接著將 Z 數列的每個元素逐一取出，分解成位元並嵌入於音訊檔聲道樣本的最低兩位有效位元(Least Significant Bit, LSB)。例如：第一個數值 Z_0，其值為區塊非零排數，將其以二進位表示為 $Z_0 = (z_0^3z_0^2z_0^1z_0^0)_2$，依每兩個位元分段可將 Z_0 分為 $(z_0^3z_0^2)_2$、$(z_0^1z_0^0)_2$ 兩組，分別取代 $S_{2,0}$、$S_{2,1}$ 的兩位

LSB， 成為 $S'_{2,0}$、$S'_{2,1}$：

$S'_{2,0} = (s_{2,0}^{15} s_{2,0}^{14} \cdots s_{2,0}^{3} s_{2,0}^{2} z_0^3 z_0^2)_2$

$S'_{2,1} = (s_{2,1}^{15} s_{2,1}^{14} \cdots s_{2,1}^{3} s_{2,1}^{2} z_0^1 z_0^0)_2$

其餘 Z 數列的元素，即由 ZigZag 順序拜訪的儲存值，則以總長 12 位元，每兩個位元嵌入。而每個樣本的一個聲道可嵌入兩位元，又每一樣本點有左右聲道，因此每三個樣本點即可完整嵌入一個 Z 數列元素。待 Z 數列中所有元素數值全嵌入完畢時，便完成一次區塊的嵌入。

當每一區塊全數嵌入完畢後，便會再重新提取下一個 8*8 區塊，重複步驟二至步驟五運算。直到所有區塊都被嵌入時，便完成圖像隱藏。圖二(a)為嵌入流程圖。

3.2 單張圖像萃取回復演算法

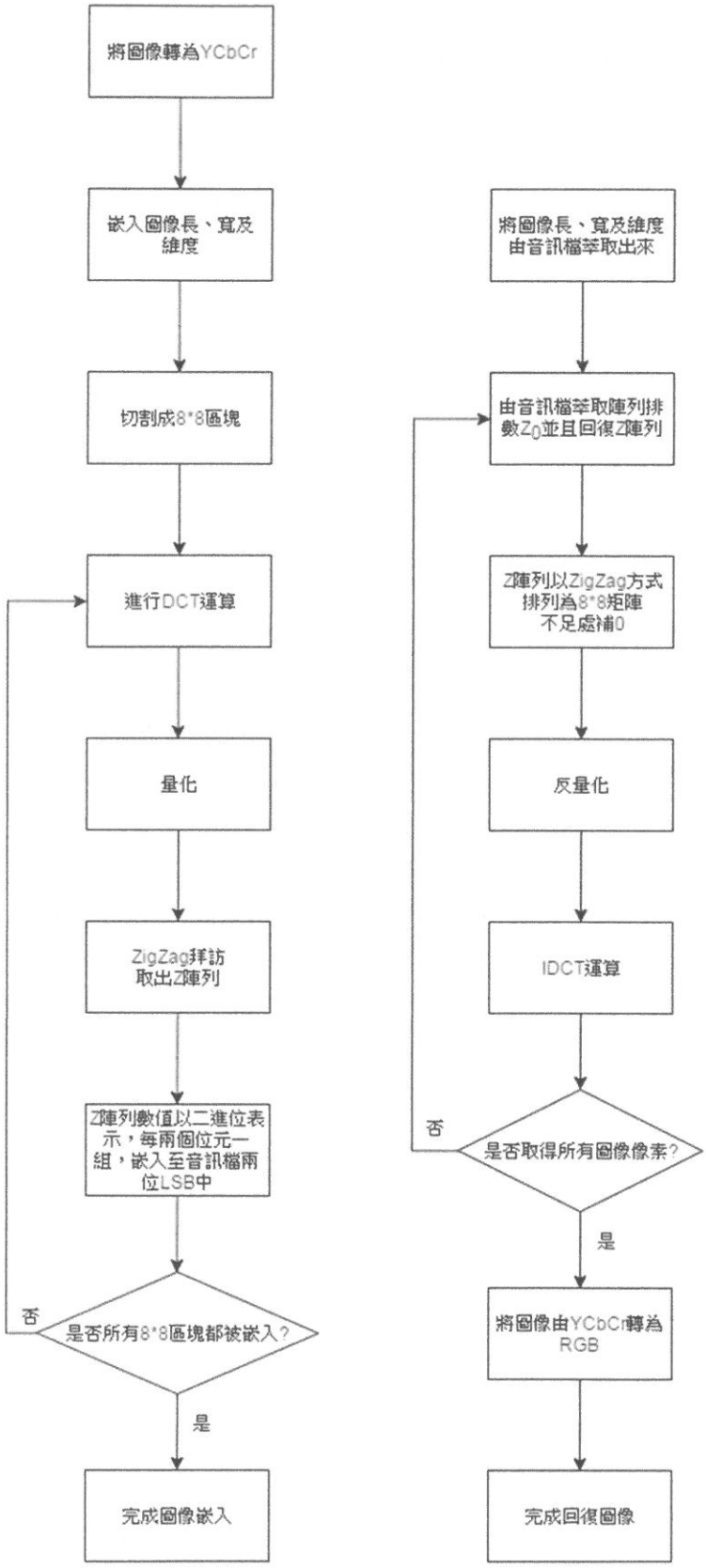
欲回復圖像時，第一步會將音訊檔樣本 $S'_{0,0}$，$S'_{0,1}$ 及 $S'_{1,0}$ 中的圖像長、寬、維度取出，以此數值可得知欲回復圖像的尺寸，接下來從 $S'_{2,0}$、$S'_{2,1}$ 各取出最低兩個位元重新組合便得到非零排數 $Z_0 = (z_0^3 z_0^2 z_0^1 z_0^0)_2$，再依照此數值，推算欲取出的 Z 數列在 8*8 區塊中 ZigZag 順序拜訪所儲存的個數，進而回復 Z 數列所有元素，並將此 Z 數列以 ZigZag 方式填入 8*8 矩陣，不足的部分補 0；得到此矩陣後，乘上亮度或色度量化表，再以 DCT 逆運算，即 IDCT，最後加上 128 可得到圖像的 YCbCr 值。當所有的 8*8 矩陣都被萃取、運算完全時，進行 YCbCr 轉 RGB 的無失真回復：

R = Y + 1.402*(Cr - 128)

G = Y - 0.34414*(Cb - 128) – 0.71414*(Cr-128)

B = Y + 1.722*(Cb - 128)

成功轉為 RGB 格式後，便完成圖像回復。圖二(b)為萃取流程圖。



(a) 嵌入流程圖　　(b) 萃取流程圖

圖 二 嵌入、萃取流程圖

3.3 多張圖像嵌入、萃取演算法

在社交媒體傳送圖像時，多張圖像傳遞的機會很高，因此在圖像隱藏至音訊檔時，可將多張圖像嵌入至音訊檔中，使音訊檔能夠攜帶的圖像數增加，除了將多張圖像分別嵌入音訊檔，也可以在運算時將多張圖像合併。因此我們提出兩種合併方式：

方法一：重組圖像

此方法是對圖像進行重新排列，方法是將 N 張圖像同一位置的像素水平相鄰排列。若以 $A_1,A_2,\cdots,A_N$ 表 N 張均為 H*W 大小之圖像，其中 H 為圖像長，W 為圖像寬，先排 n 張圖像的第一列，將第一位置 $A_1(0,0)$, $A_2(0,0)$, …, $A_N(0,0)$ 相鄰排列，再排第二位置 $A_1(1,0)$, $A_2(1,0)$, …, $A_N(1,0)$ ，以此類推。第二列至最後一列如同第一列排列方式，如表五。

表 五 重組圖像示意圖

$A_1(0,0)$	$A_2(0,0)$	…	$A_N(0,0)$	$A_1(1,0)$	…	$A_N(1,0)$	…	$A_N(W-1,0)$
$A_1(0,1)$	$A_2(0,1)$	…	$A_N(0,1)$	$A_1(1,1)$	…	$A_N(1,1)$	…	$A_N(W-1,1)$
…								…
$A_1(0,H-1)$	$A_2(0,H-1)$	…	$A_N(0,H-1)$	$A_1(1,H-1)$	…	$A_N(1,H-1)$	…	$A_N(W-1,H-1)$

實驗結果顯示此排列方式對於相似度極高的多張圖像，在還原回復圖像時的效果較佳，若是圖像相似度不高，使用此方法可能效果不佳。

方法二：並排圖像

此方法是直接將 N 張圖像依序水平排列，排列方式如表六：

表 六 並排圖像示意圖

$A_1(0,0)$	$A_1(1,0)$	…	$A_1(W-1,0)$	$A_2(0,0)$	…	$A_2(W-1,0)$	…	$A_N(W-1,0)$
$A_1(0,1)$	$A_1(1,1)$		$A_1(W-1,1)$	$A_2(0,1)$		$A_2(W-1,1)$		$A_N(W-1,1)$
…								…
$A_1(0,H-1)$	$A_1(1,H-1)$		$A_1(W-1,H-1)$	$A_2(0,H-1)$		$A_2(W-1,H-1)$		$A_N(W-1,H-1)$

不論方法一或方法二之合併圖像，長與寬皆為 H 及 N*W，將 H、W、D 嵌入 $S_{0,0}$、$S_{0,1}$、$S_{1,0}$，並且將圖像張數 N 嵌入至 $S_{1,1}$ 中，之後再以此新矩陣開始 3.1 節之步驟二至步驟五運算，直到所有切割區塊都被嵌入為止。

萃取階段取出合併陣列後，經過反量化及 IDCT 運算，將單張圖像像素值由合併陣列取出，當單張圖像像素值被完整取出時，進行 YCbCr 轉 RGB 格式，便能夠得到回復圖像。

4. 實驗

本文之目的在於提出具有保護隱私之圖像傳輸方法，以供社交媒體之應用。因此將圖像隱藏至音訊檔中，防止被他人所窺探。本節將實作所提出隱藏圖像於音訊檔方法，驗證且評估其成效。因此於 4.1 節介紹實驗評估方式，4.2 節呈現此實驗結果。

4.1 實驗評估

由於社交媒體傳輸著重於速度，且一般社交圖像主要在於分享，因此對於圖像品質之要求，只需滿足人眼視覺即可；基於以上兩個特質，所以本文在圖像隱藏的過程中採用了失真壓縮，並將此圖像失真壓縮資訊，隱藏於掩護音訊檔，成為偽裝音訊檔。一旦偽裝音訊檔傳輸予接受者後，再萃取圖像隱藏資訊，然後解壓縮回復圖像。因此。本社交圖像隱藏法，可分為兩方向評估：第一個方向為音訊檔部分，第二個方向為圖像部分。

有關音訊檔評估，本方法隱藏方式是直接將圖像資訊隱藏至 WAV 音訊檔中的 Data Chunk，即儲存樂曲音訊數據所在的地方，本文以 SNR(Signal-to-Noise Ratio)評估偽裝音訊檔之品質[4]，其公式如下：

$$SNR = 10 \times log_{10} \frac{\sum_n x^2(n)}{\sum_n [x(n) - y(n)]^2}$$

其中 x(n)為原始音訊檔之振幅，而 y(n)為偽裝音訊檔之振幅，SNR 值越大表示其品質越佳。

有關圖像評估，包含演算法時間與圖像壓縮比及圖像品質。基於社交媒體傳輸訴求，送出端的隱藏嵌入圖像時間，與接收端的萃取回復圖像時間，均不宜過長。因此，實驗數據將分別呈現嵌入與萃取階段的時間。另外，由於圖像採用失真壓縮，因此壓縮比是一項值得觀察的數據，所謂壓縮比即為原圖像與回復圖像儲存容量的比值。最後，因為圖像採用失真壓縮，因此回復圖像與原圖像並不相同，其回復圖像品質以峰值信噪比 PSNR(Peak Signal-to-Noise Ratio)做為評估標準，其公式如下：

$$PSNR = 10 \times \log(\frac{255^2}{MSE})$$

其中 MSE(Mean Squares Error)是均方差，其定義為：

$$MSE = \frac{\sum_{n=1}^{FrameSize} (I_n - P_n)^2}{FrameSize}$$

FrameSize 為 H*W*Channel，H、W、Channel 分別為圖像的長、寬、與通道數(灰

階圖像為 1，彩色圖像為 3)，I_n 是原圖像的像素值，P_n 是回復圖像的像素值。如果原圖像與回覆圖像相差越多，那麼得到的 PSNR 值也會越小。

4.2 實驗結果

本節實驗將會以音訊檔評估及圖像評估兩部分進行。圖像部分則分為單張圖像及多張圖像兩個部分。

音訊檔部分將嵌入不同圖像的偽裝音訊檔與原始音訊檔做比較，SNR 值越大則偽裝音訊檔品質越好。圖三與圖五為各單張與多張實驗圖像，表七為不同方法嵌入不同圖像的偽裝音訊檔之 SNR 值，可由單張嵌入觀察到，圖像尺寸大小會明顯的影響偽裝音訊檔之 SNR 值，換言之，圖像嵌入位元的數量將會影響偽裝音訊檔之 SNR 值，且隱藏量越多將會使 SNR 值越低。而多張嵌入部分，由於方法的不同造成嵌入量的不同，SNR 值的差異也越明顯。

表七 音訊評估

嵌入方法	嵌入圖像	檔案格式	圖像尺寸	圖像張數	SNR(dB)
單張嵌入	lena	png	256*256	1	90.7088
	flower	png	256*256	1	91.5035
	lena	jpg	256*256	1	90.8179
	camera	jpg	960*635	1	79.2356
多張重組合併	Sea	jpg	225*150	2	88.2856
	Animal	jpg	640*425	4	72.6784
	Family	jpg	225*150	5	81.0135
	Cloud	jpg	640*360	8	70.7969
多張並排合併	Sea	jpg	225*150	2	89.4553
	Animal	jpg	640*425	4	73.8495
	Family	jpg	225*150	5	81.1783
	Cloud	jpg	640*360	8	71.3275

單張圖像部分使用了多張不同尺寸大小或不同檔案格式分別測試，並比較其嵌入、萃取時間，壓縮比與 PSNR 值。圖三為各單張實驗圖像，表八為其實驗結果。

由表八可以觀察到若干現象，本方法在處理 png 圖像格式時與 jpg 圖像格式相比，壓縮比明顯較高，這是因為 png 本身為不失真壓縮的圖像格式，原圖像儲存容量高，因此我們提出的失真壓縮方式，可以達到較高的壓縮比。另外顯而易知的現象，是嵌入時間會隨著原圖像的大小而增加或減少。而壓縮比的大小，也會根據原圖像的色彩分布而有所改變，例如比較 lena.png 與 flower.png 兩張圖像，lena.png 在色彩的分布上較為平滑，而 flower.png 的色彩分布則較為跳躍，此色彩分布便會影響 DCT 運算時的頻率分布，造成壓縮比的高低差。在 jpg 格式方面，由於原圖檔已經是壓縮過的檔案格式，因此以本文方法壓縮的壓縮比便不高。

而在 PSNR 值方面，由於採用了失真壓縮，因此 PSNR 值大約在 29 左右，雖然人眼可看出與原圖的差異，不過還在人眼可接受範圍。例如：圖四為 Lena.png 之實驗結果，原圖像(左圖)與回復圖像(右圖)對比圖。

(a) lena.png (b) flower.png (c) lena.jpg

(d) camera.jpg

圖 三 單張實驗用圖像

表 八 單張圖像實驗結果

圖像	尺寸	時間		原檔大小 (KB)	回復圖檔大小 (KB)	壓縮比	PSNR
		嵌入(s)	萃取(s)				
lena.png	256*256	2.00	1.21	177	9.18	19.28	29.67
flower.png	256*256	1.99	1.13	61.9	6.12	10.11	29.26
lena.jpg	256*256	2.00	1.23	12.1	8.88	1.36	29.80
camera.jpg	960*635	16.61	11.56	241	95.7	2.51	28.98

圖 四 lena.png 單張實驗結果對比圖

有關多張圖像實驗部分，採用實驗的是尺寸大小相同場景，相似的多張圖像，這是因為社交媒體所傳輸的圖像，常具有多張與相似場景的特性。圖五為多張實驗用圖像，表九為多張重組圖像法實驗結果。

在多張重組圖像法實驗時，由表九可看出，若是同組圖像在同樣位置之顏色分布較相似，其壓縮比結果會比顏色分布差異大的群組高。而回復圖像品質數據，不論圖像群組的顏色分布差異是較大或較小，圖像之 PSNR 值皆位於 28.00~31.41 間，但若以人眼觀察為評估，可以輕易察覺出顏色分布差異較大的圖像群組，其回復圖像與原圖有較大差異。例如：Sea 群組與 Family 群組，同組圖片拍攝角度不同，造成圖像在同樣位置之顏色分布不相似，因此還原回復圖像時的效果較差。圖六為 Sea 群組實驗結果對比圖，左圖為原圖像，右圖為回復圖像。

(a) Sea

(b) Animal

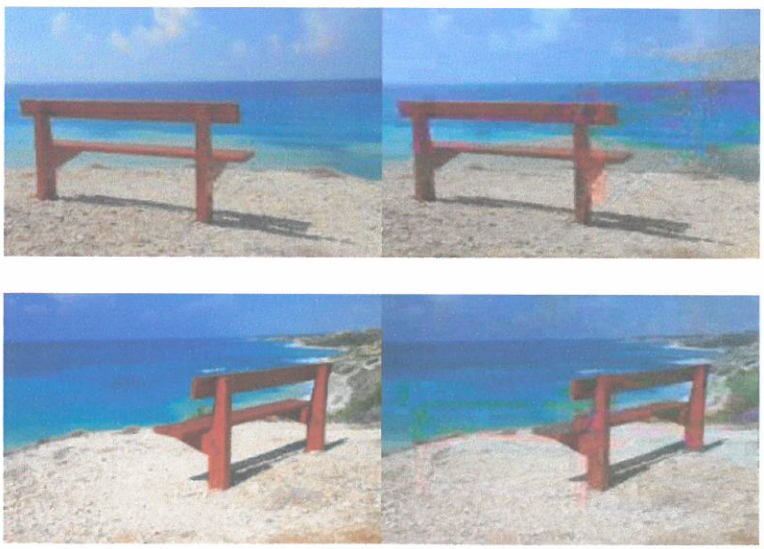
(c) Family

(d) Cloud

圖 五 多張實驗用圖像

表 九 多張重組圖像法實驗結果

實驗群組	圖 (jpg 檔)	尺寸	時間		原檔大小 (KB)	回復圖檔大小 (KB)	壓縮比	PSNR
			嵌入 (s)	萃取 (s)				
Sea	Sea0	225*150	2.99	2.29	11.8	7.05	1.67	28.93
	Sea1				13	7.02	1.85	28.73
Animal	animal0	640*425	44.00	36.62	60.2	47.8	1.25	29.61
	animal1				109	55.6	1.96	29.26
	animal2				104	55.1	1.88	29.55
	animal3				112	59.2	1.89	28.98
Family	Family0	225*150	7.61	5.83	12.1	7.49	1.61	28.35
	Family1				10.8	7.03	1.53	28.16
	Family2				7.7	6.51	1.18	28.21
	Family3				9.3	6.81	1.36	28.00
	Family4				9.92	7.07	1.40	28.00
Cloud	Cloud 1	640*360	75.62	64.21	45.1	20.6	2.19	30.88
	Cloud 2				44.4	20.3	2.19	30.65
	Cloud 3				44.8	20.3	2.21	30.57
	Cloud 4				46.5	20.7	2.25	30.65
	Cloud 5				46.3	20.7	2.24	30.73
	Cloud 6				46.5	20.6	2.26	30.81
	Cloud 7				46.7	20.4	2.29	30.88
	Cloud 8				46.5	20.7	2.25	31.41



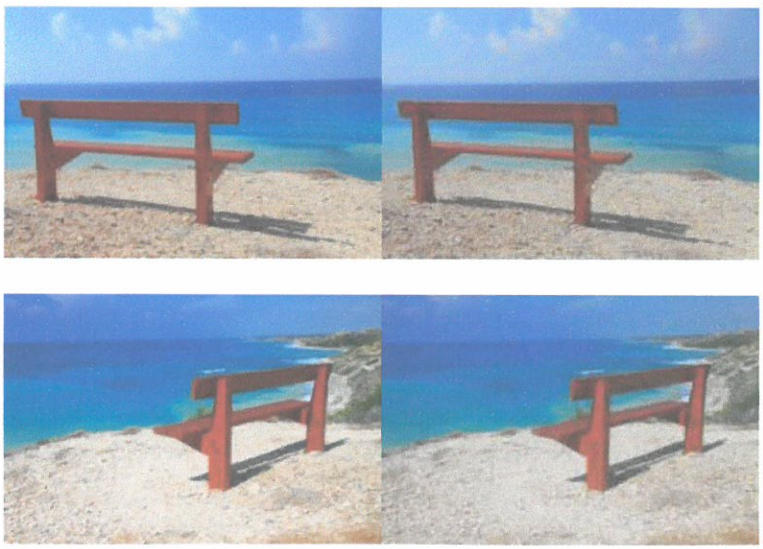
Sea

圖 六 多張重組圖像法，Sea 群組實驗結果對比圖

在多張並排圖像時，參見表十，其 PSNR 值位於 29.09~32.00 間。較多張重組圖像高，人眼也較不容易察覺回復圖像與原圖之差異。例如：圖七為 Sea 群組並排法實驗結果對比圖，左圖為原圖像，右圖為回復圖像；其視覺回復效果較圖六重組法佳。

表 十 多張並排圖像法實驗結果

實驗群組	圖 (jpg 檔)	尺寸	時間		原檔大小 (KB)	回復圖檔大小 (KB)	壓縮比	PSNR
			嵌入 (s)	萃取 (s)				
Sea	Sea0	225*150	2.63	1.68	11.8	6.25	1.88	29.09
	Sea1				13	6.75	1.92	29.20
Animal	animal0	640*425	43.49	28.91	60.2	27	2.22	31.31
	animal1				109	53.8	2.02	30.13
	animal2				104	51.0	2.03	30.27
	animal3				112	57.2	1.95	29.61
Family	Family0	225*150	7.26	5.02	12.1	6.51	1.85	28.35
	Family1				10.8	6.04	1.78	27.87
	Family2				7.7	4.56	1.68	27.75
	Family3				9.3	5.32	1.74	27.71
	Family4				9.92	5.21	1.90	27.71
Cloud	Cloud 1	640*360	69.80	48.93	45.1	15.4	2.93	31.60
	Cloud 2				44.4	15.4	2.88	31.60
	Cloud 3				44.8	15.5	2.89	31.60
	Cloud 4				46.5	15.7	2.96	31.69
	Cloud 5				46.3	15.9	2.91	31.80
	Cloud 6				46.5	16	2.91	31.90
	Cloud 7				46.7	16.1	2.90	32.00
	Cloud 8				46.5	15.9	2.92	32.00



Sea

圖 七 多張並排圖像法，Sea 群組實驗結果對比圖

5. 結論與未來展望

本文提出了圖像嵌入音訊檔之方法，在傳輸社交圖像之情形下，可維持其私密性不被有心者察覺。並且為了加速圖像嵌入的速度，對圖像做失真壓縮處理，其過程涵蓋 DCT 運算，量化及 ZigZag 順序拜訪，使得同一圖像需要嵌入的位元變少，可嵌入至音訊檔的圖像張數也增加。而在圖像嵌入至音訊檔時，只會更改音訊檔最低兩位有效位元，因此在音訊檔品質方面，也不易被察覺音訊檔中含有隱藏圖像。

本文不僅提出單張圖像的隱藏方法，也由於社交媒體所傳輸的圖像，常具有多張與相似場景的特性，而提出兩種多張圖像的隱藏方法，兩種方法均將多張圖像合併成一張圖像，再利用單張圖像隱藏法，一次嵌入多張圖像於音訊檔。兩種方法的差異性主要在於合併方式，方法一為重組圖像，將多張圖像依照圖像同一像素位置水平相鄰排列；方法二為並排圖像，依照圖像輸入順序，逐一水平排列各圖像。從兩種方法的實驗數據比較下，發現多張圖像壓縮前的排列方式，對於回復圖像之品質有很大的影響。

因此，未來的研究方向，我們希望利用社交媒體傳輸圖像的特性，即多張與相似場景，設計更適合同系列圖像的排列與嵌入隱藏方式，進而減少其嵌入、萃取時間，提高壓縮比並且保持回復圖像的品質。

參考文獻

[1] Oppenheim, Alan V., and Ronald W. Schafer. Discrete-time signal processing. Pearson Higher Education, 2010.

[2] Zamani, Mazdak, et al. "A genetic-algorithm-based approach for audio steganography." World Academy of Science, Engineering and Technology 54 (2009): 360-363.

[3] Dutta, Poulami, Debnath Bhattacharyya, and Tai-hoon Kim. "Data hiding in audio signal: A review." International journal of database theory and application 2.2 (2009): 1-8.

[4] Cvejic, Nedeljko, and Tapio Seppanen. "Increasing the capacity of LSB-based audio steganography." Multimedia Signal Processing, 2002 IEEE Workshop on. IEEE, 2002.

[5] Singh, Pradeep Kumar, and R. K. Aggrawal. "Enhancement of LSB based Steganography for Hiding Image in Audio." International Journal on Computer Science and Engineering 2.05 (2010): 1652-P1658.

[6] Asad, Muhammad, Junaid Gilani, and Adnan Khalid. "An enhanced least significant bit modification technique for audio steganography." Computer Networks and Information Technology (ICCNIT), 2011 International Conference on. IEEE, 2011.

[7] Gupta, Neha, and Nidhi Sharma. "Hiding Image in Audio using DWT and LSB." International Journal of Computer Applications 81.2 (2013).

[8] Lavanya, Budda, Yangala Smruthi, and Srinivasa Rao Elisala. "Data hiding in audio by using image steganography technique." IJETTCS Volume2 6 (2013).

[9] Jayaram, P., H. R. Ranganatha, and H. S. Anupama. "Information hiding using audio steganography–a survey." The International Journal of Multimedia & Its Applications (IJMA) Vol 3 (2011): 86-96.

[10] Wakiyama, Masahiro, Yasunobu Hidaka, and Koichi Nozaki. "An audio steganography by a low-bit coding method with wave files." Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on. IEEE, 2010.

[11] Chandrakar, Pooja, Minu Choudhary, and Chandrakant Badgaiyan. "Enhancement in Security of LSB based Audio Steganography using Multiple Files." International Journal of Computer Applications 73.7 (2013).

[12] Ciptasari, Rimba Whidiana, Kyung-Hyune Rhee, and Kouichi Sakurai. "An enhanced audio ownership protection scheme based on visual cryptography."EURASIP Journal on Information Security 2014.1 (2014): 1.

[13] http://soundfile.sapp.org/doc/WaveFormat/

A Method for Hiding Images in Audio Signals for Social Media

Po-Yuan Chiang and Jen-Ing G. Hwang
Department of Computer Science and Information Engineering
Fu Jen Catholic University
Taipei, Taiwan 242, R.O.C

Abstract

With the advancement of information technology, social media has become an important part of people's lives. Particularly, sending images to share personal photos is becoming a popular activity. However, this rarely entails the use of security measures. To address this problem, this paper presents a data hiding method that uses audio signals as cover objects to carry hidden images. The proposed approach has two main parts: the image embedding phase and the image extracting phase. During embedding, we combine several images into one image, and embed this combined image in an audio file. However, a technique of lossy compression that is a preprocessing task is used to reduce the data size of the secret image and to store or transmit the hidden data in an efficient form. During extracting, the embedded data can be obtained from the stego audio, and hidden images are restructured and revealed from the embedded data by applying the inverse data compression. Stego audio can be defined as the cover audio combined with the embedded hidden data. Experimental results showed that the proposed method attained acceptable performance on the measures of embedding time, extracting time and image quality. This confirmed the efficiency and effectiveness of the proposed method.

Key words: social image, social media, embedding, extracting