DSP Final Project Team 9

Image in Audio Steganography

組員: 徐浩宇、黃友廷

I. 問題定義/應用場景

本次的 Final Project 我們會去實作聲音隱寫術的功能，主要內容是透過資料轉換把想要傳送的圖片隱藏在一段音檔內，達到資訊安全的目的。

有別於傳統常見的隱寫術，我們獨創了一個新的隱寫方法，把圖片視為是一個 spectrogram，透過 STFT (Short-time Fourier Transform)轉換成聲音訊號過後，再用 key 加密並隱藏在載體音檔內部。

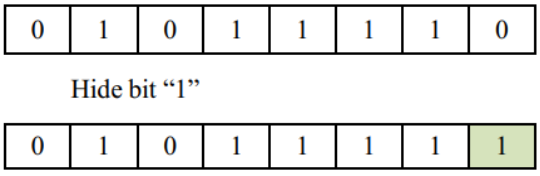
在這次的報告中，除了探討自創隱寫方法外，我們也自己刻了一些傳統的隱寫方法去做比較。實驗結果發現，自創隱寫方法成功解決了傳統隱寫方法不同面向上的問題，對於圖片的還原效果也相當好。

II. 問題分析

傳統隱寫術的方法介紹如下:

A. Least significant bit (LSB) coding:

將 secret message 的 bits 覆寫掉每個聲音訊號中最低位的 bit (LSB)。



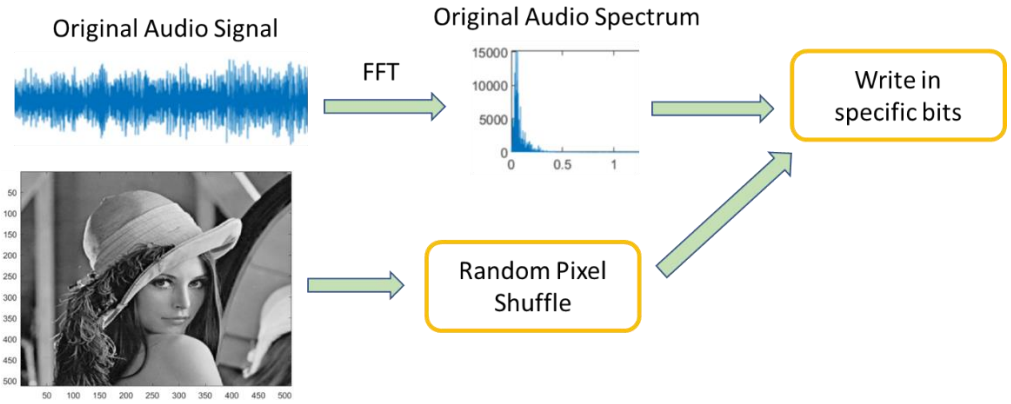
優點: 實現方便且快速，需要極少的計算量。

缺點: 容易被傳統訊號處理的方法攻擊。

B. Spread Spectrum coding:

將 secret message 的 bits 隨機散佈在聲音訊號的頻譜，寫在某一個 bit 上。

Note: 這邊必須成對填寫，以維持實數訊號的 magnitude 偶對稱性質。



優點: 比較不容易被訊號處理方法攻擊

缺點: 更改頻譜的數值容易造成音檔的 noise 增加，且轉換過程的精度差會造成圖片還原的完整度下降。

C. Phase coding:

(1) 將音檔分割成數個 segments，每個 segments 的大小為 encoded message 的長度。

(2) 對每一個 segment 去做 Discrete Fourier Transform (DFT)

(3) 計算相鄰 segment 之間的 phase 差距(phase difference)

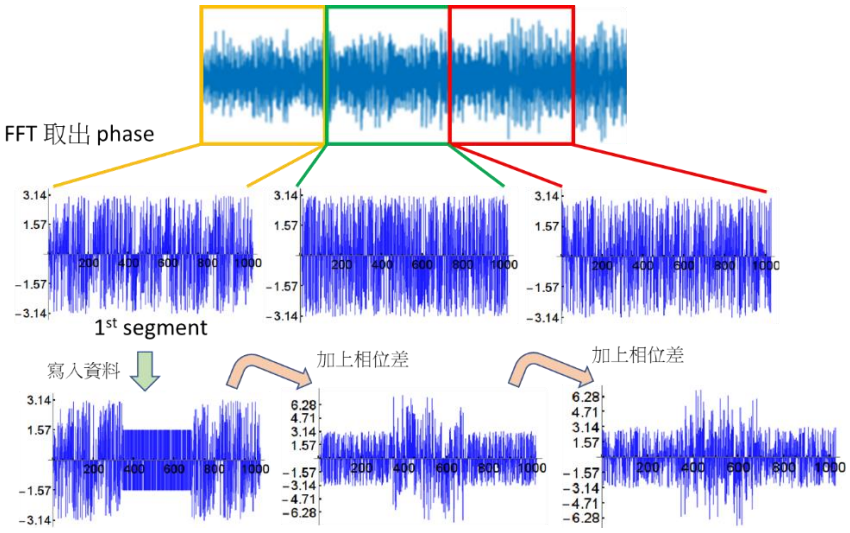
(4) 根據以下規則去修改第一段 segment 裡面的 phase :

$$phase\_new = \begin{cases} \pi/2 & if \ \ message \ \ bit = 0 \\ -\pi/2 & if \ \ message \ \ bit = 1 \end{cases}$$

Note: 這邊必須成對填寫，以維持實數訊號的 phase 奇對稱性質。

(5) 利用前面計算過的 phase difference 去更新除了第一段外的所有 phase

Note: 透過加上相位差去保持原本的相對相位，維持音檔的連續性。



優點: 更動 phase 相較更動 magnitude 不容易被人耳觀察出來，能夠解決前面方法會產生 noise 的問題。

缺點: 只能寫在第一個 segment，能夠隱寫的範圍較小，需要更大音檔長度去容納相同的圖片大小。

D. Our Proposed Method :

我們的方法是將圖片用 STFT 轉成音檔後，透過直接寫掉音檔高頻的部份去隱藏資料，這樣做能夠改善的部分如下:

(1) 不需要把圖片轉成 binary 的形式，大大減少隱藏需要的音檔長度。

ex: image(64x64) → 我們的方法需要 4096 (64x64)個點

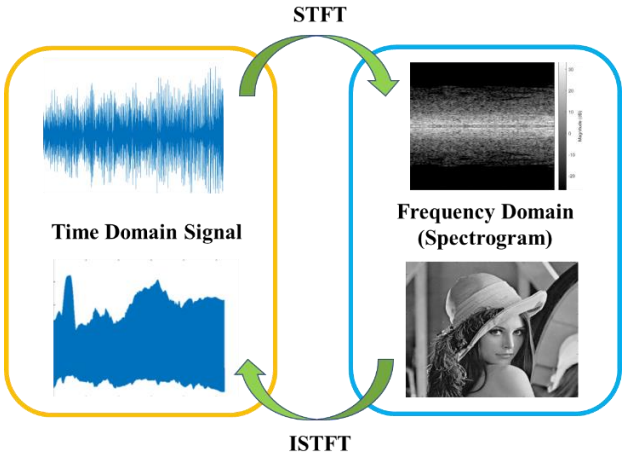
其他方法需要 32768 (64x64x8)個點

(2) 覆寫在音檔高頻的部分，能夠減少寫在 LSB 會產生 noise 的問題。

(3) 隱寫後的訊號難以察覺是圖片訊息，也因此比較難去攻擊、解密。

III. 定義方法

核心概念就是假定存在一段音檔 S，把 S 拿去做 Short-time Fourier Transform 後，產生的 spectrogram = STFT(S)畫出來會跟目標圖片一樣。反之，我們有一個圖片為 Img，將圖片做 Inverse Short-time Fourier Transform 過後，產生的 S = ISTFT(Img)即為要藏入的聲音訊號。示意圖如下:



首先，我們將 Lena 圖片透過 flip 變成上下對稱的圖片後，透過 ISTFT 轉回假定的音檔 S。會需要上下對稱的原因在於，如果把 Lena 圖片直接透過 STFT 轉回聲音訊號，會發現其為複數(complex)訊號，而不是一般音訊的實數(real)訊號，這樣在隱寫進入音訊上會有很大的問題。

實數訊號的頻域有 magnitude 偶對稱的性質(Figure 1)，所以我們要透過 flip 把頻譜變成偶對稱(Figure 2)，這樣才能轉回去實數訊號的音訊。

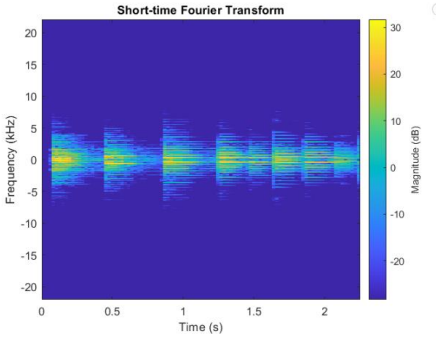


Figure 1

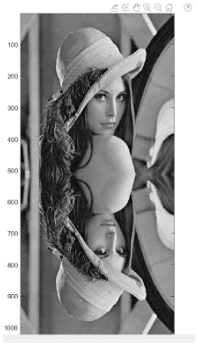


Figure 2

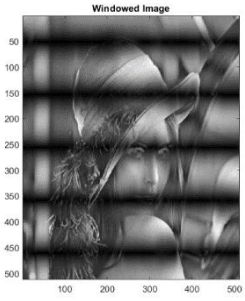


Figure 3

STFT 參數造成的影響:

```
stft(x,fs,'Window',,'OverlapLength', ,'FFTLength',);
```

(1) OverlapLength:

我們要轉換的圖片並不符合現實中音頻的 spectrogram，現實中聲音訊號透過 overlap STFT 轉換的 spectrogram 前後時間點的頻譜會有 correlation，而圖片本身不具有這個特性。

如果設定成有 overlap 去做轉換，就會無法成功恢復 (Figure 3)，因此我們設定 OverlapLength 為 0。

(2) Window:

根據 Short Time Fourier Transform 的公式:

$$X[k,n] = \sum_{m=0}^{L-1} w[m]x[n+m]e^{-j\left(\frac{2\pi k}{N}\right)m}$$

因為經過 flip 後產生的圖片大小為 1024 x 512 (1024: 頻譜點數、512: 時域點數)，為了維持轉換前後資訊的完整性，我們把 Window Length 設為 1024 以避免頻率(縱軸)上的資訊被省略。

我們選擇的 window 種類為 Kaiser Window，公式如下:

$$w[n] = \begin{cases} \dfrac{I_0\left[\beta\sqrt{1-[(n-\alpha)/\alpha]^2}\right]}{I_0(\beta)}, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

因為我們想要在 window 過後保留最大程度的資訊，避免 distortion。經過簡單測試後發現當 β 越小的時候，時域上的 window 變化越小，越能夠保留資訊，故選擇 $\beta = 1$。(Figure 4)

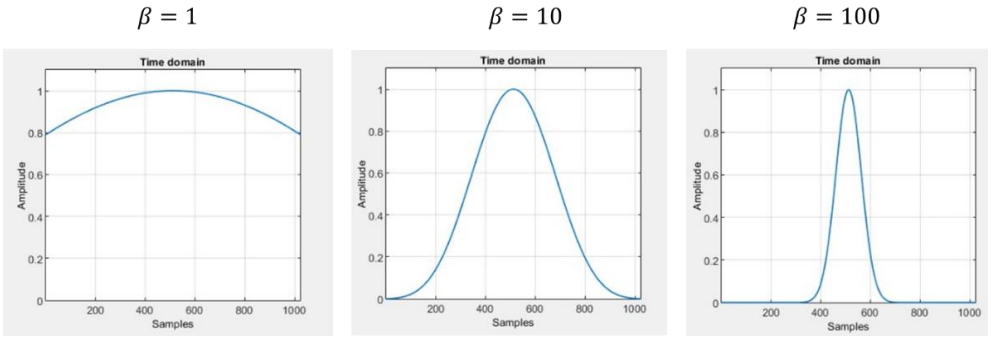
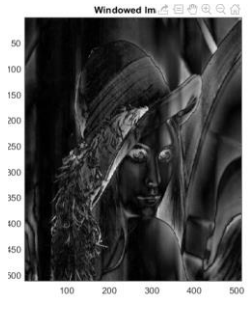
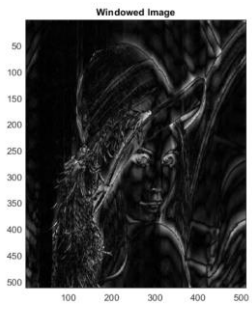


Figure 4

經過測試後發現 $\beta = 15$ 之後圖片的 distortion 會大大增加，見下表(Table 1):

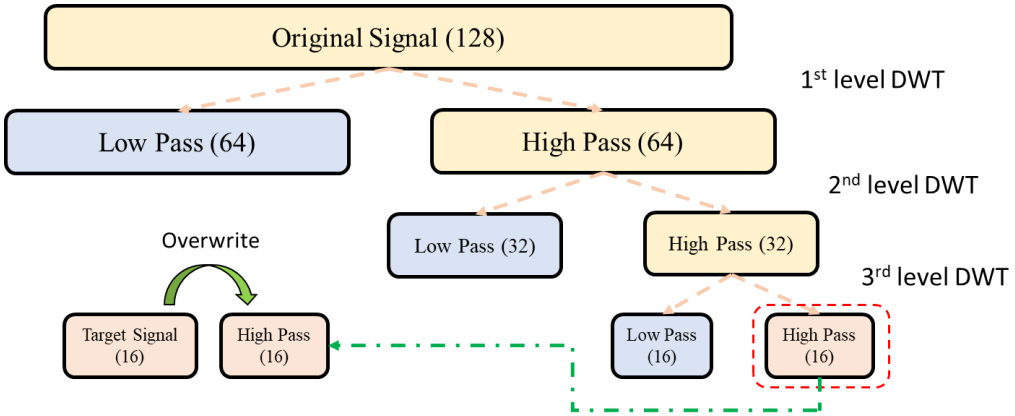
Table 1

				
β	1	15	18	20
PSNR	35.82	35.82	5.987	5.79
SNR	30.13	30.13	0.2965	0.0994

隱寫術設計:

根據上述的設定與操作，我們就能使用 ISTFT 將 Lena 圖片轉成一段 Target Signal，接著我們的目標是要將這段 Target Signal 隱寫到目標音檔的高頻區域，因為人耳對於高頻的音訊較為不敏感，所以覆寫在高頻區域較不易被察覺。

我們的實作方式就是使用 Discrete Wavelet Transform (DWT)，透過不斷切割高頻區域，找到大小和 Target Signal 相近的部分直接覆寫上去，實作流程如下圖:



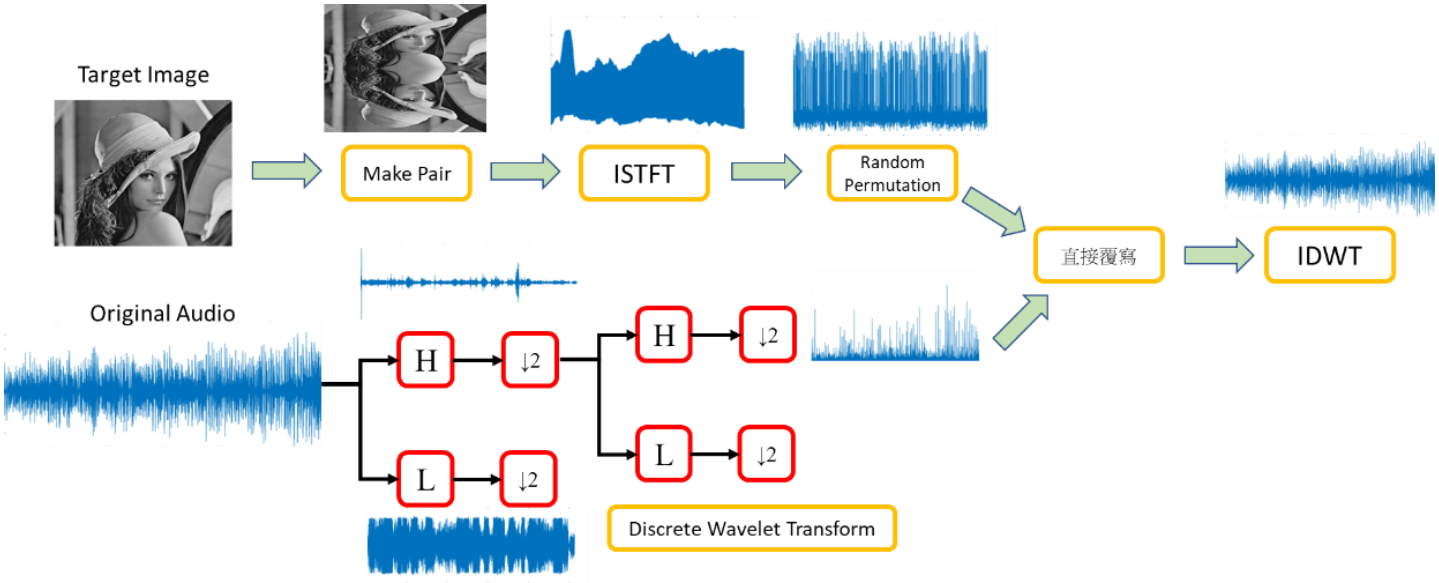
寫入過後再透過 Inverse Discrete Wavelet Transform (IDWT) 轉回去原本的音檔，達到隱寫的目的。在接收端也可以同樣透過 DWT 去對音檔做分解直到目標大小後，直接抓取 Target Signal 並做 STFT 得到目標圖片。

加密方法設計:

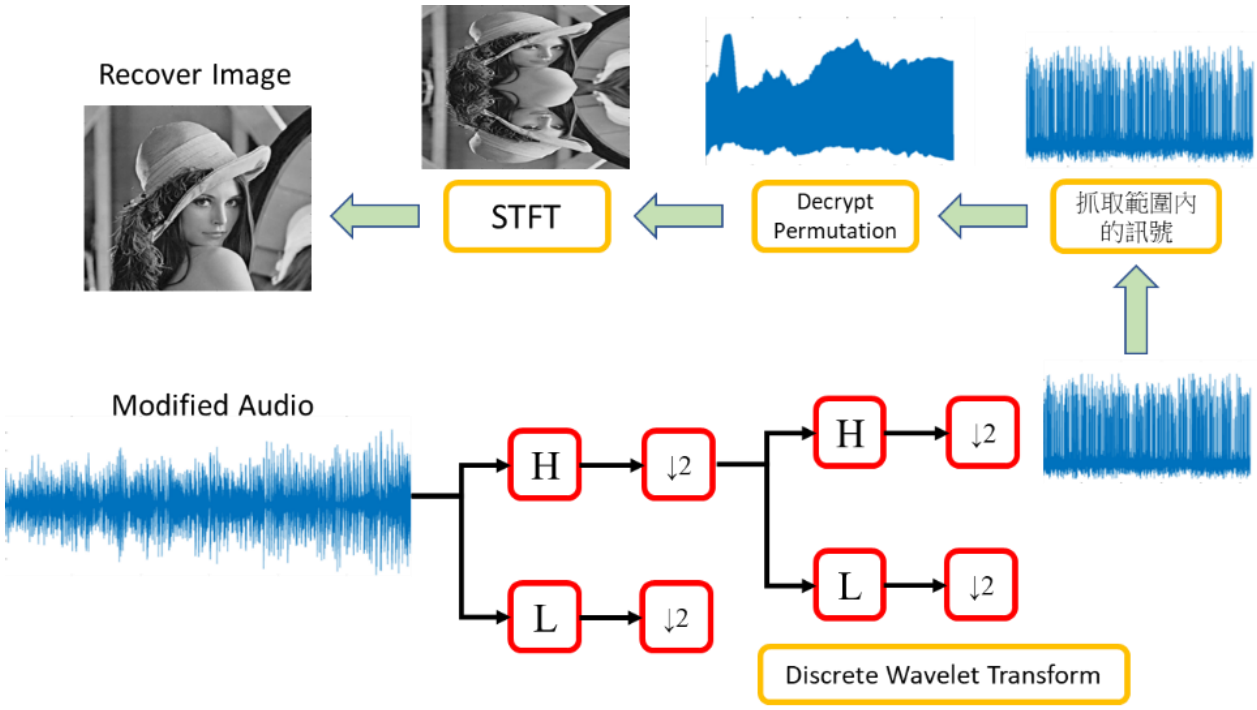
加密的方法是用在 Target Signal 隱寫到高頻音檔之前，透過 key (一個數字) 去設定 random seed 並產生特定的 random permutation 的方式去打亂 Target Signal。接收端同樣也可以透過 key 去產生相同的 permutation 去解出原本的 Target Signal，再經過 STFT 後得到目標圖片。

實作流程圖:

A. 隱寫圖片進入音訊的步驟:



B. 從隱寫音訊提取圖片出來的步驟:



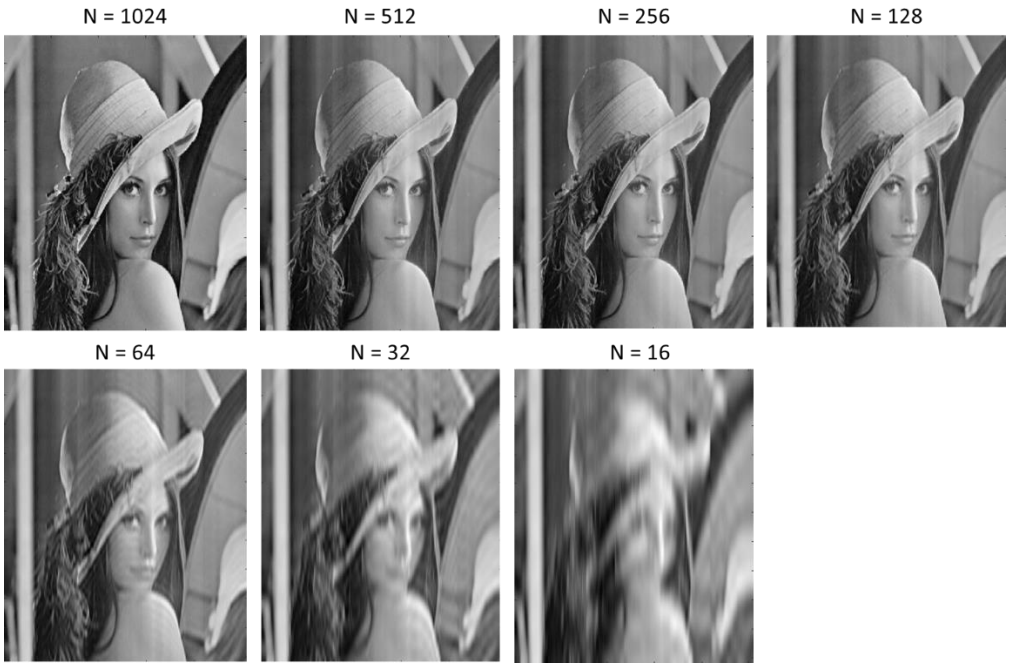
IV. 結果分析

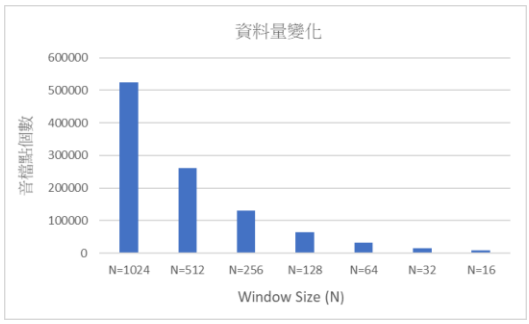
改變 STFT Window Length 比較表:

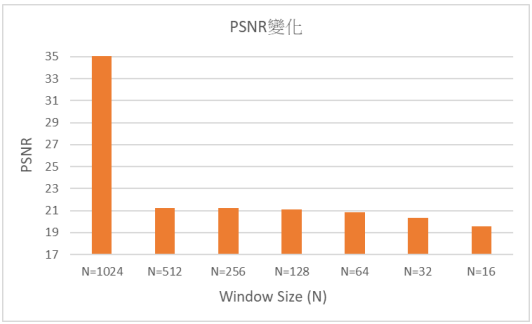
Table 2

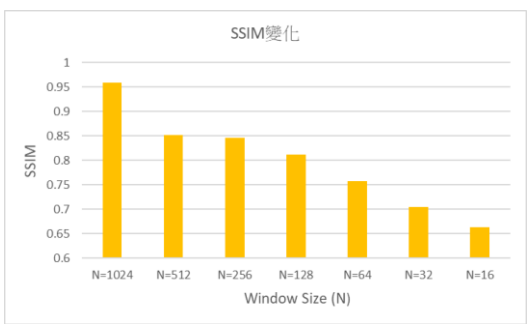
Window Length	N=1024	N=512	N=256	N=128	N=64	N=32	N=16
音檔長度	524288	262144	131072	65536	32768	16384	8192
PSNR	35.8288	21.2604	21.2481	21.1347	20.8425	20.3529	19.582
MSE	2.61e-04	7.5e-03	7.5e-03	7.7e-03	8.2e-03	9.2e-03	0.011
SSIM	0.9588	0.8522	0.8453	0.8116	0.7574	0.7045	0.6626

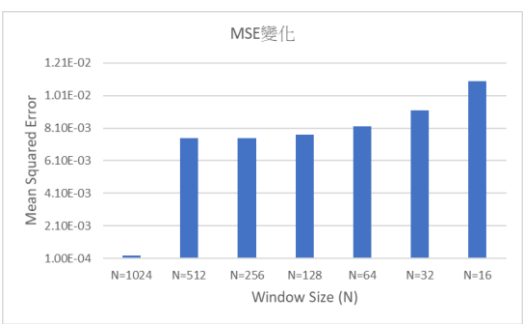
改變 STFT Window Size 比較圖:











由 Table 2 可以知道 window length (N)與資料點成正比，也與 PSNR 成正比。但仔細觀察 PSNR 與 window length 的關係圖可以發現 PSNR 的變化並不會是與 window length 成線性關係，而是從 N=1024 到 N=512 間有一個很大差距，之後的變化量比較少，主要是因為從 N=1024 到 N=512 資料量就已經掉了一半，導致 PSNR 在此區間有明顯的變化。此一現象可以從 PSNR 的公式來解釋:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

(1) N = 1024:

假設每一 pixel (範圍 0~256)之間差異極小(假設為 1)，可以大約估算

$$\text{MSE} = \frac{1}{1024 \times 512}(1^2) \times 512 \times 512 = \frac{1}{2}$$

$$\text{PSNR} = 10\log\left(\frac{\text{MAX}^2}{MSE}\right) = 10\log\left(\frac{\text{MAX}^2}{\frac{1}{2}}\right) = 10\log(2 \times MAX^2)$$

(2) N = 512:

因為資訊量減半，故假設有一半的 pixel 差異較大(取均值 128)，可以大約估算

$$\text{MSE} = \frac{1}{1024 \times 512}(128) \times 256 \times 512 = 64$$

$$\text{PSNR} = 10\log\left(\frac{\text{MAX}^2}{MSE}\right) = 10\log\left(\frac{\text{MAX}^2}{32}\right) = 10\log\left(\frac{1}{32} \times MAX^2\right)$$

(3) N = 256:

因為資訊量再減半，故假設有四分之三的 pixel 差異較大(取均值 128)

$$\text{MSE} = \frac{1}{1024 \times 512}(128) \times 384 \times 512 = 48$$

$$\text{PSNR} = 10\log\left(\frac{\text{MAX}^2}{MSE}\right) = 10\log\left(\frac{\text{MAX}^2}{48}\right) = 10\log\left(\frac{1}{48} \times MAX^2\right)$$

(4) 根據(1)、(2)可知，N=1024 與 N=512 在 PSNR 的差異是:

$10\log(2)+10\log(\text{MAX}^2)-10\log(1/32)- 10\log(\text{MAX}^2) = 10\log(2)+10\log(32) = 60\log(2)$

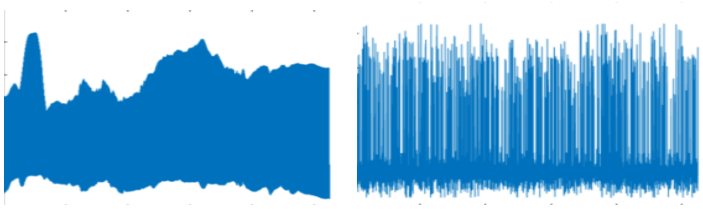
(5) 根據(2)、(3)可知，N=512 與 N=256 在 PSNR 的差異是:

$10\log(1/32)+10\log(\text{MAX}^2)-10\log(1/48)- 10\log(\text{MAX}^2) = -10\log(32)+10\log(48)$

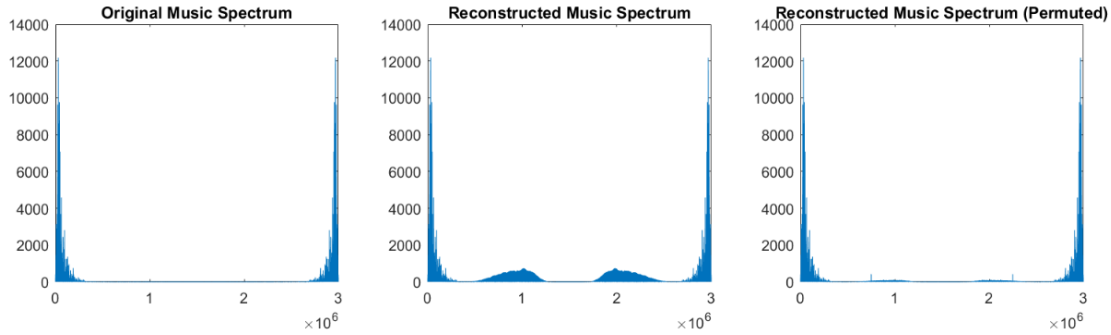
$= -50\log(2)+55.8\log(2) = 5.8\log(2)$

Discussion: (5)和前面(4)相比有著較少的差異，可以知道隨著 N 的遞減，PSNR decay 的速度也會遞減。因此，如果想要節省資料嵌入的數量，可以藉由調整 Window Length 的大小來進行選擇，以此去迎合不同目標之需求。

使用 Random Permutation 對於頻譜的影響:



(左圖) 發現圖片 ISTFT 產生的聲音訊號能量非常集中，故使用 Key 去產生一個 random permutation 去把該聲音訊號打散，順便一起做加密 (右圖)。



(中圖) 將沒打散的聲音訊號轉回去，發現他會在高頻產生能量。
(右圖) 將打散的聲音訊號轉回去，發現聲音訊號在高頻的能量小了不少。

	Permutated data	Original data
SNR	72.2578	68.2964
MSE	7.5152e-10	8.1572e-10

由上表可以發現嵌入 Permutated data 所得到的 SNR 值確實會比嵌入 Original data 來的好。

使用 random permutation 會將原本集中的頻譜分散，會使原本所對應的頻寬被放大，讓原本表現集中高頻的嵌入訊號，有著接近 white noise 的性質，使嵌入訊號造成的變異能夠平均分散。而根據公式:

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}} = \frac{A^2_{\mathrm{signal}}}{A^2_{\mathrm{noise}}}$$

例如，有一個 noise 的 amplitude 是 10，那麼此時的 SNR 值是$\frac{A^2_{signal}}{A^2_{noise}} = \frac{A^2_{signal}}{100}$。

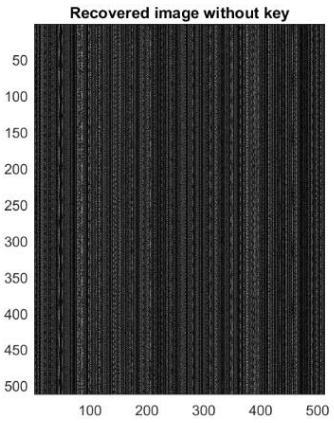
但如果是將此一 noise 的 amplitude 平均分散成 10 段，可推得:

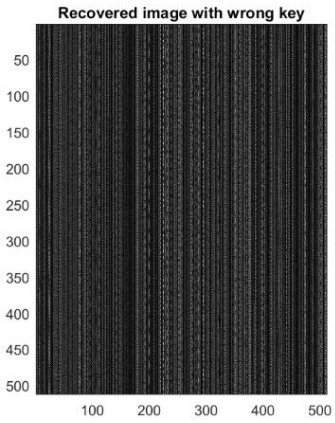
$$SNR = \frac{A^2_{signal}}{10 * A^2_{noise}} = \frac{A^2_{signal}}{10 * 1} = \frac{A^2_{signal}}{10}$$

很明顯看出 SNR 的值會提升 10 倍，將變異分散確實能夠使得 SNR 的數值上升。

雖然打散有助於減少高頻能量，聲音仍然會有刺耳的聲音，所以我們做了一個簡單的 scaling 去把高頻的能量直接降低到人耳聽不到的範圍。

使用 Key 做加密的效果:





和其他隱寫方法比較後的結果:

圖片	LSB coding	Phase coding	Spread Spectrum	Our Proposed Method
MSE	0	4.7876e-07	1.6909e-04	2.6138e-04
PSNR	Inf	63.1988	37.7189	35.8273
SSIM	1	0.9999	0.9739	0.9587
聲音	LSB coding	Phase coding	Spread Spectrum	Our Proposed Method
MSE	8.3212e-09	9.2808e-07	1.25e-13	7.5152e-10
SNR	61.2462	41.5323	110.39	72.2578

各方法在圖片及聲音的表現排行:

(1) 圖片: $LSB > Phase\ coding > Spread\ Spectrum > Our\ Proposed\ Method$

(2) 聲音: $Spread\ Spectrum > Our\ Proposed\ Method > LSB > Phase\ coding$

LSB coding 的方法是直接對時域訊號做隱寫，所以圖片恢復效果為完美，但會對於聲音產生一些 noise。

Phase coding 因為是直接去看 phase 是否為 $\frac{\pi}{2}\sim-\frac{\pi}{2}$ ，對於圖片恢復效果也極佳，但是更改 phase 會造成每段音檔之間不連續，使得聲音品質最差。

Spread Spectrum 是透過更改頻域的 LSB 去做隱寫，對於聲音的影響較低，但是因為轉換過程的精度差會造成圖片還原的效果較差。

Our Proposed Method 因為把圖片做了一些特殊轉換，故還原效果最差。但是其產生較少資料點的特性，有助於降低隱寫在高頻訊號時對於原本音檔的影響，聲音品質維持不錯。

V. 結論

我們設計了圖片隱寫到音訊內的方法，有效降低傳統方法對於音檔的影響。雖然圖片的恢復程度略差，但是其特殊的轉換特性能有效降低被發現和破解的機率，用較少資料點去表示也可以減少目標音訊所需的長度。

VI. 貢獻

A.方法貢獻:

自創了新的隱寫及加密方法，使用了 STFT、DWT、FFT 等課堂教到的概念。

B.實作貢獻:

使用 MATLAB:

(1)自創隱寫方法(encryptionDWT.m、decryptionDWT.m、STFT_Method.mlx)

(2) 2D 圖片加密(encryption2D.m、decyption2D.m)

(3) 手刻傳統隱寫方法(LSB_Method.mlx、Phase_Coding_Method.mlx、Spread_Spectrum_Method.mlx)

FFT、DWT、STFT、PSNR、SSIM 都是使用 MATLAB 內建的函式庫。

VII. 組內分工

我們做了相同的貢獻，題目發想、code 實作及報告撰寫都是共同完成的。

VIII. 文獻及資料參考

Paper:

[1] An Effective Technique for Hiding Image in Audio. Najiya Thasneem, Renjith V Ravi. IJSR 2013.

[2] An Improved Technique for Hiding Data in Audio. Huynh Ba Dieu, Nguyen Xuan Huy. IEEE 2014.

[3] Kriti Saroha, Pradeep Kumar Singh. “A Variant of LSB Steganography for Hiding Images in Audio”. International Journal of Computer Applications. Dec 2010.

[4] Dalal N. Hmood, Khamael A. Khudhiar, Mohammed S. Altaei. “ A New Steganographic Method for Embedded Image In Audio File.

Textbooks:

Ingemar J. Cox. “ Digital Watermarking and Steganography”. 2008.