R11945072 張栢彥 ADSP_HW3

Homework 3 (Due: May 3rd)

(1) Write a Matlab or Python code for the 4:2:0 image compression technique. write B = C420(A), A is the input color image and B is the reconstructed image. Just use the interpolation method for reconstruction. The code should be handed out by NTUCool. (Note: The command rgb2ycbcr cannot be used.)

(25 scores)

(2) Suppose that there is a multipath system $y[n] = x[n] + 0.3x[n-15] + 0.2x[n-25]$. (a) Find $p[n]$ such that $y[n] = x[n] * p[n]$. (b) Design the lifter to remove the effect of $p[n]$ and try to not destroy $x[n]$ as possible. (10 scores)

(3) Suppose that there are three vocal signals: (i) $\cos(300\pi t)$, (ii) $-\sin(1200\pi t)$, (iii) $\sin(6000\pi t)$. (a) Which voice sounds louder? (b) Which voice signal can be propagated to a longest distance? (10 scores)

low.f.

△. 高f?

same A.

Piano concept.

(4) Suppose that for a stringed instrument the frequency of Do is 240 Hz. (a) Determine the frequencies of Mi and So for the instrument. (b) Suppose that the rate of wave propagation is 340m/sec. Determine the lengths of the strings to generate Mi and So for the stringed instrument.   (10 scores)

(5) In addition to the DCT, which is adopted by MP3, write at least three possible ways that can compress a music signal more efficiently.

(10 scores)

(6) In the JPEG process, (a) why the DCT is used instead of the DFT for transformation?   Write at least two reasons. (b) Why the input image is separated into several 8x8 blocks before using the DCT? Write at least two reasons. (c) Why the DC difference is encoded instead of the original DC value? (d) Why zigzag is beneficial for AC term encoding?   (20 scores)

利於 low f, 高f×給齊

(*Continued*)

※ Poisson 机率分布 → entropy 算出, / log 2

(7) Suppose that $P(x = n) = e^{-\lambda}\lambda^n/(n!)$ for $n = 0, 1, 2, 3\ldots., 40$ where $\lambda = 0.97$. Also suppose that length(x) = 50000. Estimate the range of the total coding lengths in the binary system when using (i) the Huffman code and (ii) the arithmetic code. (15 scores)

stay

(Extra): Answer the questions according to your student ID number.
      (ended with 0, 1, 3, 4, 5, 6, 8, 9)

(2) $y[n] = x[n] + 0.3x[n-15] + 0.2x[n-25]$

(a) 使得 $y[n] = x[n] * p[n]$

① 其原式可視為 $x[n]$ (input) 及其延遲的線性組合

② 找到其系統脈衝信號響應, 記為 $\delta[n]$

$$\begin{cases} \delta[n] = 1 \\ \delta[n] = 0 \end{cases}, \text{ for } \begin{array}{l} n=0 \\ n \neq 0 \end{array}$$

③ 又可將 impulse response 之 $\delta[n]$ 換成 $x[n]$, 得:

$$y[n] = \delta[n] + 0.3\delta[n-15] + 0.2\delta[n-25]$$
$$= x[n] \cdot p[n]$$

④ 又 $x[n]$ 為 input, 故 impulse response 為 $p[n]$

$$p[n] = \delta[n] + 0.3\delta[n-15] + 0.2\delta[n-25] \quad \#$$

(b) 在此, 試著用 LMS 方法達成目標; 我們目標是找到脈衝響應 $q[n]$ 抵消 $p[n]$,

因 $y[n] = x[n] * p[n]$

$\Rightarrow x[n] = y[n] * q[n]$

運用 LMS:

① 隨機值初始化濾波器係數 $q[n]$

② 定義濾波長度 $L=30$ 作為起點, 因系統至多有 25 個樣本 delay

③ 設定步長參數 $\mu = 0.001$ (做收斂值收斂)

④ 每個樣本誤差 $e[n] = x[n] - y\_filtered[n]b$

→ 更新濾波器係數 $q[k] = q[k] + \mu^* e[n]^* y[n-k]$ / 對於 $k = 0, 1, \dots L-1$

⇒ $y[n]$ 以此得以經由一 filter 進行運算以抵消 $p[n]$ 之效力, 並不摧毀 $x[n]$

pseudo code :

```
def lms_adaptive_filter (x, y, L, mu):
        N = len(x)
        q = np.zeros(L)
        y_filtered = np.zero(N)
```

```
h = np. zeros (N)

for n in range (N):
    y_filtered[n] = np. dot (q, y[n-L+1 : n+1][::-1])
        ↳ # y[n] 輸出用現在 filter 係數

    e[n] = x[n] - y_filtered[n]
    ※ 計算 x[n], y_filter[n] 誤差

    for k in range (L):
            q[k] += mu* e[n] * y[n-k]
    return q
```

$x = np.\ random.\ randn\ (1000)$

$y = x + 0.3^* np.\ roll\ (x, -15) + 0.2^*np.\ roll\ (x, ->5)$

$L = 30$

$mu = 0.001$

$q = lms\_adaptive\_\ filter\ (x, y, L, mu)$

(3)

(a) loader 在客觀見上應考量振幅, 然而在此 振幅皆為1, 以為主觀考量,
雖無法知聽起來最大聲, 但 人耳於 1000-3000 Hz 最為敏感。

考量振幅同. 信號 $\sin(6000\pi t)$ 為 3000Hz. 在人耳聽覺中感知最大聲。(若不考慮此則一樣大聲)

(b) 通常較低頻率的較不易受傳播介質衰減或吸收, 較低頻率會傳得更遠。

故 $\cos(300\pi t)$, $f = 150Hz$ 、 預計會傳更遠。

(4)

(a)

$$D_0 \xrightarrow{7\text{个半音}} M_i \to S_0$$

$$D_0 \xrightarrow{4\text{个半音}} M_i$$

$D_0 : 240Hz$

每個半音 : $\sqrt[12]{2} = 1.0595$

∴ $M_i : 240Hz \times (1.0595)^4 \approx 302Hz$

$S_0 : 240Hz \times (1.0595)^7 \approx 360Hz$ #

(b) 若 $V = 340m/s$, 要確定弦的長度 ?

$$V = 2fL$$

$$Mi(320Hz): \quad L(Mi) = \frac{V}{2f} = \frac{340}{2(302)} \approx 0.56291m$$

$$So(360Hz): \quad L(So) = \frac{V}{2f} = \frac{340}{2(360)} \approx 0.47222m$$

(5) 除了 MP3 的 DCT 壓縮外，其它 技術 更有效 的壓縮音樂信號

① Linear Predictive Coding (LPC):
數字信號 數據壓縮網絡 的技術了。它如比的假設，即給定樣本可近似於
過去 sample 的線性組合。通過 LPC 對信號建模又需對 LPC 係數和殘餘誤差進
行編碼而實現壓縮。 ex. CELP. Speex

② Perceptual Audio Coding:
這是一種壓縮技術，它利用人類聽覺特性来移除或減少分配給音頻信號感知
不相關、次要部分的 bits。通過利用聽覺系統的掩蔽效應，可使不多增感知的方式實現
更高的壓縮比。包含 AAC、AC-3 等解碼器。

③ Lossless Compression Algorithms:
保留原始信號而不丟失任何信息。雖然它們比有損方法實現更低壓縮比，使
保持音頻質量時更有效。例如: FLAC、ALAC、APE

④ Wavelet Transform:
小波變換可用於分析、壓縮 時頻域 的技術。它使用縮放、移動之小波函數提
供信號之多辨率分析。小波變換對于穩信號特有效，已被用於各壓縮。ex: WavPack、
JPEG2000.

(6) 在 JPEG 過程，

(a) 用 DCT > DFT 作為圖片轉換？

① 實值: 因為圖片是複數，若要用 Fowler Transform 需要分別對於
實虛數做複虛實轉換並相加，而 DCT 則不用。

② 大部分 DFT 高頻部分能量少，若要先對高頻做刪節的轉換 則可用 KLT
⇒ DCT 有更好的能量壓縮特性: DCT中 大多變換圖像都集中在少數低數係數中

(b) 在用 DCT 前，將圖像分成 8×8塊的原因？

(可局部處理各個細節)

① 頻率分佈 隨空間分布不同，可不要全局考慮，透過 8×8切割 才看見到局部細節。

② 暫存記憶體所需更少：① 切成很多塊較小的，於是其實計算量更少
                        ② 所需儲存量較少，每次其實又要 64 個值儲儲存

③ DCT 的運算複雜度近似於 FT，切成 8×8 也可降低運算量。

ex: 原本 → $O(M*N\log N) + O(N*M\log M) = O(MN\log(MN))$
    改後 → $(M/8)(N/8) * 64 \log(64) = MN\log 64 → O(MN)$ #

(c) 對 DC 差異進行編碼更好，因其提供了更好的壓縮性能。由於空間兄餘
    圖像中相鄰的 8×8 塊通常具有相似 DC 值。因此若針對相鄰塊 DC 差異點編碼
    所需值 range 更小，更易於壓縮。

(d) Zigzag 掃瞄有利於 AC 編碼。通常因能量集中在低 f
    而 Q 越小則 f(敦字)越高，則四捨五入後 ~0 的機率越高 (EOB 特性)。
    使用 Zigzag 可依先處理低頻，是二維→一維的最好策略。
    Zigzag 以最大化零值係數表達行概率方式對係數排序。量化後，
    大部分高 f 交流係數 ~0 或幅度小。Zigzag 可將這些小係數組合在一起
    使編碼技術更有效壓縮數據，提高 JPEG 的整體壓縮效率。

7. 需先計算 Source 的熵，表無噪聲信道中每個符號的 bits

① $H(X) = -\sum P(x) * \log_2(P(x))$，其中總和取所有 x 可能值 (n=0, 1, 2, ... 40)

② 給定 $P(x=n) = e^{-\lambda} \times \lambda^n / n!$ 和 $\lambda = 0.97$

⇒ codes: $\lambda = 0.97$
           probabilities = [np.exp(λ)* (λ*n) /math. range for n in range (41)
           entropy = -sum ([p*np.log₂(p) for p in probabilities])

⇒ 熵 ≈ 1.658 bits/symbol

③ 計算兩者編碼長度 range:

(i) Huffman: 為每丁 symbol 生成整數個投，因此平均代碼長度
             介於 H(X) 和 H(X)+1 之間，給定 length = 50000；

Lower bound : $50000 \times H(x) = 50000 \times 1.658 \approx$ $\boxed{82900 \ bits}$

Upper bound : $50000 \times (H(x)+1) = 50000 \times 2.658 \approx$ $\boxed{132900 \ bits}$

故 以 Huffman code 的預估編碼長度, 介於 $82900 \sim 132900 \ bits$ 之間 #

(ii) *Arithmatic :* 因不需整數位數(對每了symbol). 故可實現更趨於熵H(x)的 編碼長度。

∴ *Total Coding Length* $\approx 50000 \times H(x) \approx$ $\boxed{82900 \ bits}$ #

=

(號碼尾數2, 無 *extra point !*)

R11945072.