

高等計算機圖學與應用 6650

Assignment 06: 3D Non-Equilateral Arnold Transform (3D-NEAT)

授課教師：王宗銘

2023/04/27

1. 請撰寫 2 個 python 程式，練習利用 3D Non-Equilateral Arnold Transform (3D-NEAT) 對灰階或彩色影像作 bit-level encryption and decryption。

(1) bit-level encryption 加密程式：

程式名稱 學號-06-3D-NEAT_enc.py。請使用以下矩陣做 NEAT 轉換，並在程式中給定參數之數值。只要更改參數數值，即可重新購建不同的矩陣，作 NEAT 轉換。座標 $[x_n, y_n, z_n]$ 經過 3D-NEAT 之轉換矩陣，依照轉換矩陣(1), (2), (3)之順序，可產出對應的座標 $[x_{n+1}, y_{n+1}, z_{n+1}]$ 。請注意：座標 $[x_n, y_n, z_n]$ 所儲存之內容為 1 個 bit 的數值，0 或 1。假設為 0，此數值需要取代新座標 $[x_{n+1}, y_{n+1}, z_{n+1}]$ 之原先儲存之 bit 內容。因此，3D Non-Equilateral Arnold Transform 是屬於 bit level transformation。非如 2D Arnold Transform，原像素數值取代轉換後座標的像素數值。

| | |
|--|-----|
| $\{S_z\}: \begin{bmatrix} x_z \\ y_z \\ z_z \end{bmatrix} = \begin{bmatrix} 1 & b_z & 0 \\ c_z & 1 + b_z c_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod \begin{matrix} N \\ M \\ K \end{matrix}$ | (1) |
|--|-----|

| | |
|--|-----|
| $\{S_x\}: \begin{bmatrix} x_x \\ y_x \\ z_x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b_x \\ 0 & c_x & 1 + b_x c_x \end{bmatrix} \begin{bmatrix} x_z \\ y_z \\ z_z \end{bmatrix} \bmod \begin{matrix} N \\ M \\ K \end{matrix}$ | (2) |
|--|-----|

| | |
|--|-----|
| $\{S_y\}: \begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} 1 + b_y c_y & 0 & c_y \\ 0 & 1 & 0 \\ b_y & 0 & 1 \end{bmatrix} \begin{bmatrix} x_x \\ y_x \\ z_x \end{bmatrix} \bmod \begin{matrix} N \\ M \\ K \end{matrix}$ | (3) |
|--|-----|

$$c_z = r_z * \frac{M}{\gcd(N, M)}, c_x = r_x * \frac{K}{\gcd(M, K)}, c_y = r_y * \frac{N}{\gcd(K, N)}$$

$\gcd(\cdot)$ 代表最大公因數， $b_x, b_y, b_z, r_x, r_y, r_z$ 為任意正整數。

請在程式內設定一個變數 L ($5 \leq L \leq 1000$)，代表加密的次數。

(2) bit-level encryption 解密程式：

程式名稱 學號-06-3D-NEAT_dec.py。請使用以下 NEAT encryption 的逆矩陣做 Inverse NEAT (INEAT)轉換，如 Eqs. (3), (4), (5)。參數之數值需與同於加密程式。解密時，需先將輸入的 $[x_{n+1}, y_{n+1}, z_{n+1}]$ 依照(4), (5), (6)順序解密成為 $[x_n, y_n, z_n]$ 。

| | |
|---|-----|
| $\{S_y^{-1}\}: \begin{cases} x_x = (x_{n+1} - c_y z_{n+1}) \bmod M \\ y_x = y_{n+1} \\ z_x = (z_{n+1} - b_x x_x) \bmod K \end{cases}$ | (4) |
|---|-----|

| | |
|---|-----|
| $\{S_x^{-1}\}: \begin{cases} x_z = x_x \\ y_z = (y_x - b_x z_z) \bmod N \\ z_z = (z_x - c_x y_x) \bmod K \end{cases}$ | (5) |
|---|-----|

| | |
|---|-----|
| $\{S_z^{-1}\}: \begin{cases} x_n = (x_z - b_z y_n) \bmod M \\ y_n = (y_z - c_z x_z) \bmod N \\ z_n = z_z \end{cases}$ | (6) |
|---|-----|

2. 請設定以下目錄

source 目錄，儲存欲加密之影像。

encrypt 目錄，儲存已加密之影像，並請在原始檔案名後加入 enc。

decrypt 目錄，儲存被解密之影像，並請將檔案名稱後加入 dec。

例如：欲加密影像為 Lena.png，存在 source image 內；加密後之影像為 Lena_enc.png，存在 encrypt 目錄內；解密後之影像為 Lena_dec.png，存在 decrypt 目錄內。

param 目錄，儲存使用的參數檔案。參數檔案名稱：parameter.txt，內容每行依序為 N, M, K, gcd(N, M), gcd(M, K), gcd(K, N), b_x , b_y , b_z , r_x , r_y , r_z , c_z , c_x , c_y 。請每行儲存一個數值，並請在數值前加入文字，已提供必要資訊，必免混淆。例如 N=512。最後請列出 3×3 的轉換矩陣 $[S_z]$, $[S_x]$, $[S_y]$ 。每個矩陣 3 行，空白間隔內部 3 元素，例如 2 1 0。

4. 撰寫之程式：

(1) 可以使用 openCV 套件。

(2) 請注意 python openCV 之頻道排列是 blue, green, red，非為 red, green, blue。請做向量處理。

(3) python 版本 ≥ 3.10 ，請確認程式在 IDLE python 64 bit 是可執行的。

(4) 提供 10 張 standard USC-SIPI 影像資料庫、Kodak 影像資料庫、HDR NCHU 影像資料庫經過色調轉換後之 png 影像。影像有可能是灰階，也有可能是彩色，影像也有可能是 equilateral。請利用 image.type() 或其他方法，自動判斷影像為灰階或是彩色型態。若是灰階，請設定 $K=8$ ，若是彩色，請設定 $K=24$ 。N 為影像之水平像素數量，M 為影像之垂直像素數量。

(5) 請將測試影像放在 source 目錄內。加密程式逐一將上述影像作加密處理。加密後，解密影像逐一將加密影像做解密處理，並放在對應的目錄內。

(6) 測試影像包含 1. AtriumNight.png, 2. Boat.png, 3. Desk_oBA2.png, 4. kodim04.png,

5. kodim07.png, 6. kodim19.png, 7. kodim22.png, 8. kodim23.png, 9. Peppers.png, 10. Tank.png。

5. 繳交：請繳交壓縮檔案，壓縮方式請選 zip 或 rar。

壓縮檔案名稱：學號-ass07.rar，包含下列 3 目錄

(1) 2 個 Python 程式，請放在與 source, encrypt, decrypt, parame 同層

加密程式：學號-06-3D-NEAT_enc.py

解密程式：學號-06-3D-NEAT_dec.py

(2) source 目錄：內含原始 10 張影像

(3) encrypt 目錄：內含已加密之 10 張影像

(4) decrypt 目錄：內含已解密之 10 張影像

(5) parame 目錄：內含 parameter.txt 檔案。

(7) readme.txt，請放在與 source, encrypt, decrypt, parame 同層，敘述如何執行 python 程式，載明是否需要額外的套件。