色彩科學導論與應用

Arnold Transformation

授課教師：王宗銘

2021/04/15

Assignment 8

The Arnold transform (ART), also known as 'cat mapping', was introduced by Vladimir I. Arnold in 1967 in the study of ergodic theory (遍歷性理論). It is later being used for image encryption and decryption to increase the image security. Please write two python programs to implement the Arnold transform.

The first program aims to encrypt images and the second one is for image decryption. The ART has shown to be periodic, which means that the original image will reappear after a certain number of iterations. Thus, we define the term "period value (P)" to be the *minimal* number of iterations for a given image which can reappear after P number of ART iterations. Your first python program needs to compute the period value (P) and append it to the corresponding line in the output file.

Program 1: Encrypting a number of images within the input file.

The first python program, "**學號-08-ART-ENC.py**, reads an input file, **ART-ENC-input08.txt**, and applies the ART or invert ART to the original image(s) listed within the input file. The output, **ART-ENC-output08.txt**, appends the period within the input file (see example).

Input: **ART-ENC-input08.txt**, which contains *N* lines (N=5 in this example) . In each line, there are three items separated by a blank:

1. An image name, say Alschari-1000.bmp, to be processed
2. A sign to indicate the ART or invert ART (IART), where "+" means the ART and "-" represents the IART. Note that:
   (1) the ART adopts the formula, where N represents the horizontal or vertical of a square image with the resolutions of $N \times N$.

   $(x', y') = \text{ART}[(x, y), N]$

   $x' = (x+y) \bmod N$

   $y' = (x+2y) \bmod N$

   In addition, we may express it as a matrix multiplication, as shown below:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N}, (x, y) \in N.$$

(2) Alternatively, the IART adopts the following formula

$(x',\ y') = \text{IART}[(x, y), N]$

$x' = (2x\text{-}y) \bmod N$

$y' = (\text{-}x\text{+}y) \bmod N$

In addition, we may express it as a matrix multiplication, as shown below:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N}, (x, y) \in N.$$

3. The period P needed for the test image.

Example of an Input file: **ART-ENC-input08.txt**

Alschair-1000.bmp + 20
Anturium-512.bmp + 187
BeaverDam-800.bmp - 27
Blackcat-1280.jpg + 741
Cars-1024.bmp - 45

Example of an output file: **ART-ENC-output08.txt**

ART+20_Alschair-1000.bmp + 20 600
ART+187_Anturium-512.bmp + 187 512
ART-27_BeaverDam-800.bmp - 27 480
ART+741_Blackcat-1280.jpg + 741 768
ART-45_Cars-1024.bmp - 45 1024

Program 2: Decrypting a number of encrypted images within the input file.

The second python program, "**學號-08-ART-DEC.py**, reads an input file, **ART-DEC-input08.txt**, and applies the ART or invert ART to decrypt image(s) listed within the input file. The output, **ART-DEC-output08.txt**, lists the images that have been processed. (see example).

Example of an Input file: **ART-DEC-input08.txt**

ART+20_Alschair-1000.bmp - 20 600

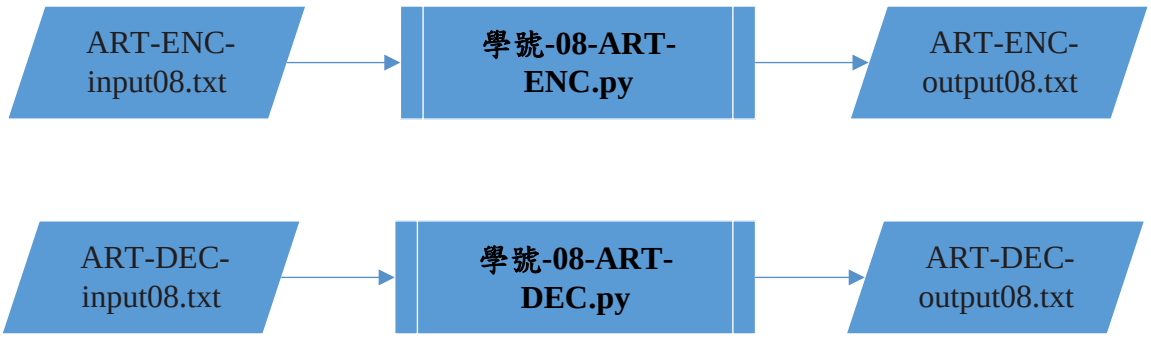
```
ART+187_Anturium-512.bmp - 187 512
ART-27_BeaverDam-800.bmp + 27 480
ART+741_Blackcat-1280.jpg - 741 768
ART-45_Cars-1024.bmp + 45 1024
```

Example of an output file: **ART-DEC-output08.txt**

```
Alschair-1000.bmp
Anturium-512.bmp
BeaverDam-800.bmp
Blackcat-1280.jpg
Cars-1024.bmp
```

Note:

1. A compression file, cat-master, is provided, which can be modified to write your own program.



Submission:

Please submit the following **SIX files.**學號-08-ART-ENC.py

1. 學號-08-ART-DEC.py
2. ART-ENC-input08.txt
3. ART-ENC-output08.txt
4. ART-DEC-input08.txt
5. ART-DEC-output08.txt

Please also submit the following **FIVE encrypted images** with proper names.

ART+20_Alschair-1000.bmp

ART+187_Anturium-512.bmp

ART-27_BeaverDam-800.bmp
ART+741_Blackcat-1280.jpg
ART-45_Cars-1024.bmp