

高等影像處理

作業#1

姓名：_____林郁庭

學號：_____110310238

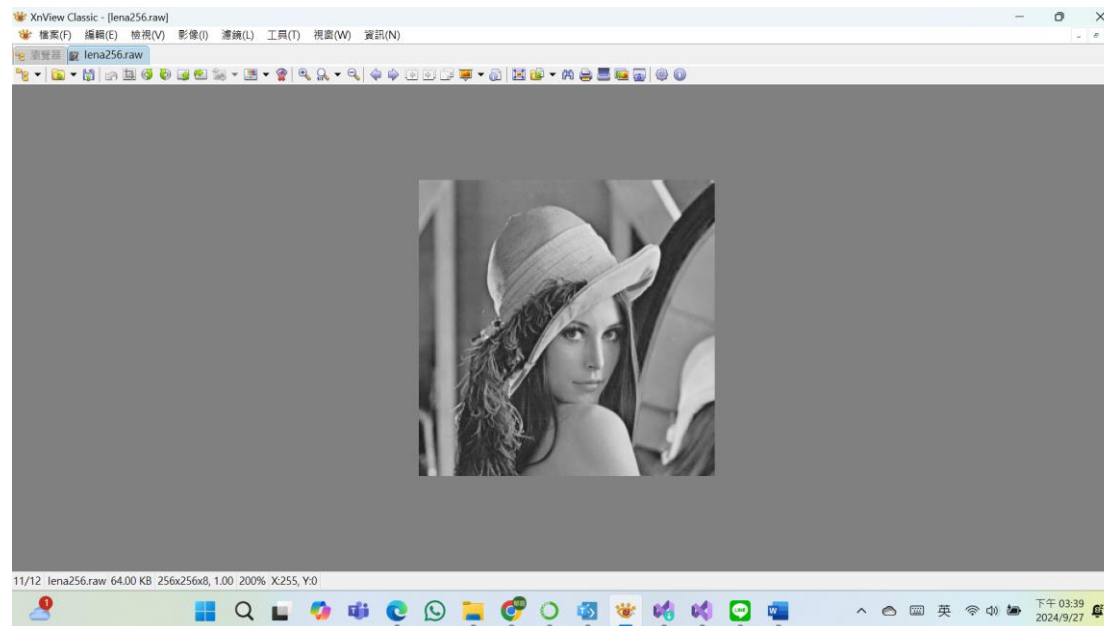
指導老師：_____ (你的指導老師)

(Note: 善用分頁符號)

1.1

b


Lena256.raw (標示圖片名稱)



Disgussion

我用 Xnview classic，一開始無法打開 raw 檔，後來進入到選項裡改變設定將 raw 勾選就可以看了。

1.2
a
<pre>1.2.a(1):Pixel value at (78, 199): 176 1.2.a(2):Pixel value at (16888-th): 157 1.2.a(2):Pixel coordinate: (65, 247) 1_3_b's random value:-33[INFO:0@0.054]</pre>
Disgussion
<p>(a) 在座標(78, 199)的值為 176。</p> <p>若從(0,0)開始算，座標(78, 199)是第 79 個 row 和 200 個 col，所以在一維陣列中的第 $78 * \text{width} + 200$ 個值即是我們要的像素值。</p> <p>(b) 在第 16888 個儲存的位置，也就是座標為(65, 247)的值為 157。</p> <p>我用「$\text{col_2} = \text{position} \% \text{width} - 1;$」和「$\text{row_2} = (\text{position} - \text{col_2}) / \text{width};$」算出第 16888 個儲存的位置為座標(65,247)，值為 157。</p>

1.2
b
<p>1_2_b_out.raw(1_2_b_out.jpg)</p> 
Disgussion
<p>在 IOTest 後，在 Xnview 中打開結果與 lena256.raw 相同</p>

1.2
c
1_2_c_out.raw(1_2_c_out.jpg)



Disgussion

將其分割為十六塊後就可以知道其移動位置和旋轉角度。我先創建一個新的一維陣列，在將 lena256 的各區塊一塊塊依照以下描述指定到新的陣列中。

(1) → (16)，無旋轉

$ad_img[(x + 192) * 256 + (y + 192)] = img_lena[x * 256 + y];$

(2) → (15)，無旋轉

$ad_img[(x + 192) * 256 + (y + 64)] = img_lena[x * 256 + y];$

(3) → (14)，逆旋轉 90 度

$ad_img[((191 - y) + 192) * 256 + (x + 64)] = img_lena[x * 256 + y];$

(4) → (13)，無旋轉

$ad_img[(x + 192) * 256 + (y - 192)] = img_lena[x * 256 + y];$

(5) → (12)，逆旋轉 90 度

$ad_img[(191 - y) * 256 + (x + 128)] = img_lena[x * 256 + y];$

(6) → (11)，無旋轉

$ad_img[(x + 64) * 256 + (y + 64)] = img_lena[x * 256 + y];$

(7) → (10)，正旋轉 90 度

$ad_img[(y) * 256 + (191 - x)] = img_lena[x * 256 + y];$

(8) → (9)，逆旋轉 90 度

$ad_img[(384 - y) * 256 + (x - 320)] = img_lena[x * 256 + y];$

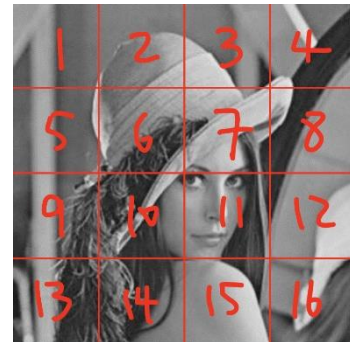
(9) → (8)，逆旋轉 90 度

$ad_img[(255 - y - 64 - 64) * 256 + (x + 192 - 128)] = img_lena[x * 256 + y];$


(10) → (7)，正旋轉 90 度

$ad_img[(x - 64) * 256 + (y + 64)] = img_lena[x * 256 + y];$

(11) → (6)，無旋轉



$\text{ad_img}[(255 - y) * 256 + (x - 64)] = \text{img_lena}[x * 256 + y];$
(12) → (5) , 正旋轉 90 度
$\text{ad_img}[(y - 128) * 256 + (191 - x)] = \text{img_lena}[x * 256 + y];$
(13) → (4) , 逆旋轉 90 度
$\text{ad_img}[(y) * 256 + (447 - x)] = \text{img_lena}[x * 256 + y];$
(14) → (3) , 無旋轉
$\text{ad_img}[(x - 192) * 256 + (y + 64)] = \text{img_lena}[x * 256 + y];$
(15) → (2) , 逆旋轉 90 度
$\text{ad_img}[(191 - y) * 256 + (x - 128)] = \text{img_lena}[x * 256 + y];$
(16) → (1) , 正旋轉 90 度
$\text{ad_img}[(y - 192) * 256 + (255 - x)] = \text{img_lena}[x * 256 + y];$

1.3
a
<p>1_3_a_out.raw(1_3_a_out.jpg)</p> 
Disgussion
<p>用雙層 for 迴圈一一處理每個值。在 if 中設定條件只要大於等於 179 就將值設定為 255，避免溢值變成黑色。</p>

1.3
b
1_3_b_out.raw(1_3_b_out.jpg)



Disgussion

設定隨機值範圍為 0~161，再減 81 將範圍移動到-80~80。用雙層 for 迴圈一一處理每個值。在 if 中設定條件只要像素值加上隨機值大於等於 255 就將值設定為 255，避免溢值變成黑色。設定條件只要像素值加上隨機值小於等於 0 就將值設定為 0，避免溢值變成白色。

1.3

c

Disgussion

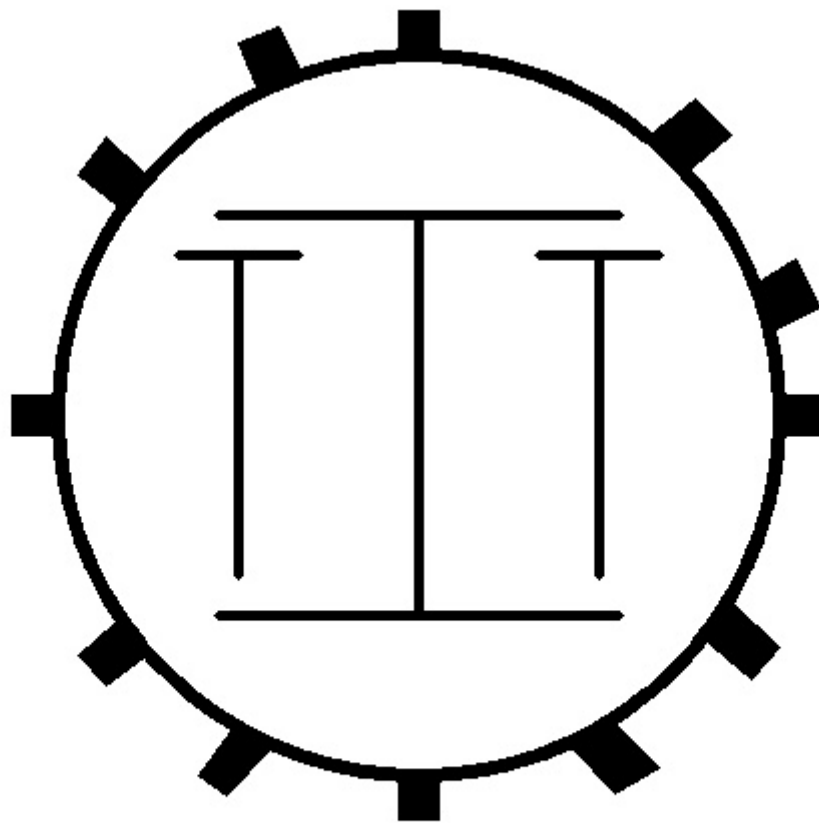
我們用 unsigned char 資料型態儲存圖片，其值範圍為 0~255，若溢值的話，可能從黑色變成白色或從白色變成黑色。

- (a) 用雙層 for 迴圈一一處理每個值。在 if 中設定條件只要大於等於 179 就將值設定為 255，避免溢值變成黑色。
- (b) 設定隨機值範圍為 0~161，再減 81 將範圍移動到-80~80。用雙層 for 迴圈一一處理每個值。在 if 中設定條件只要像素值加上隨機值大於等於 255 就將值設定為 255，避免隨機值為正數時發生溢值變成黑色。設定條件只要像素值加上隨機值小於等於 0 就將值設定為 0，避免隨機值為負數時發生溢值變成白色。

2

b

2_b.jpg



110310238

Disgussion

先創建 512x512 的白底圖片

//創建圓

```
cv::Point center(256, 220);
```

```
int radius = 180;
```

```
cv::Scalar color(0, 0, 0);
```

```
int thickness = 2;
```

```
cv::circle(image, center, radius, color, thickness+3);
```

//數字

```
std::string text = "110310238";
```

```
cv::Point org(160, 500);
```

// 文字的左下角位置

```
int fontFace = cv::FONT_HERSHEY_SIMPLEX;
```

// 字型

```
double fontScale = 1.0;
```

// 字型大小

```
cv::putText(image, text, org, fontFace, fontScale, color, thickness);
```

```
//依照下面程式碼一一找出線條兩邊座標
cv::Point startPoint(156, 120);    // 起點座標
cv::Point endPoint(356, 120);    // 終點座標
cv::line(image, startPoint, endPoint, color, thickness + 1);
//依照下面程式碼一一找出方形四個角座標
std::vector<cv::Point> points;
points.push_back(cv::Point(246, 18)); // 左上角
points.push_back(cv::Point(266, 18)); // 右上角（斜向右上）
points.push_back(cv::Point(266, 38)); // 右下角
points.push_back(cv::Point(246, 38)); // 左下角（斜向左下）
std::vector<std::vector<cv::Point>> fillPoints;
fillPoints.push_back(points);
//將所有方形印到圖片上
cv::fillPoly(image, fillPoints, color);
cv::imshow("Taipei Tech", image);
cv::imwrite("2_b.jpg", image);
```