**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 |
|  |
| Disgussion |
| 1. 在座標(78, 199)的值為176。   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\*width+200個值即是我們要的像素值。   1. 在第16888個儲存的位置，也就是座標為(65, 247)的值為157。   我用「col\_2 = position % width-1;」和「row\_2 = (position - col\_2) / width;」算出第16888個儲存的位置為座標(65,247)，值為157。 |

|  |
| --- |
| 1.2 |
| b |
| 1\_2\_b\_out.raw(1\_2\_b\_out.jpg) |
| Disgussion |
| 在IOTest後，在Xnview中打開結果與lena256.raw相同 |

|  |
| --- |
| 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];   1. 🡪 (15)，無旋轉   ad\_img[(x + 192) \* 256 + (y + 64)] = img\_lena[x \* 256 + y];   1. 🡪 (14)，逆旋轉90度   ad\_img[((191 - y) + 192) \* 256 + (x + 64)] = img\_lena[x \* 256 + y];   1. 🡪 (13)，無旋轉   ad\_img[(x + 192) \* 256 + (y - 192)] = img\_lena[x \* 256 + y];   1. 🡪 (12)，逆旋轉90度   ad\_img[(191 - y) \* 256 + (x + 128)] = img\_lena[x \* 256 + y];   1. 🡪 (11)，無旋轉   ad\_img[(x + 64) \* 256 + (y + 64)] = img\_lena[x \* 256 + y];   1. 🡪 (10)，正旋轉90度   ad\_img[(y) \* 256 + (191 - x)] = img\_lena[x \* 256 + y];   1. 🡪 (9)，逆旋轉90度   ad\_img[(384 - y) \* 256 + (x - 320)] = img\_lena[x \* 256 + y];   1. 🡪 (8)，逆旋轉90度   ad\_img[(255 - y - 64 - 64) \* 256 + (x + 192 - 128)] = img\_lena[x \* 256 + y];   1. 🡪 (7)，正旋轉90度   ad\_img[(x - 64) \* 256 + (y + 64)] = img\_lena[x \* 256 + y];   1. 🡪 (6)，無旋轉   ad\_img[(255 - y) \* 256 + (x - 64)] = img\_lena[x \* 256 + y];   1. 🡪 (5)，正旋轉90度   ad\_img[(y - 128) \* 256 + (191 - x)] = img\_lena[x \* 256 + y];   1. 🡪 (4)，逆旋轉90度   ad\_img[(y) \* 256 + (447 - x )] = img\_lena[x \* 256 + y];   1. 🡪 (3)，無旋轉   ad\_img[(x - 192) \* 256 + (y + 64)] = img\_lena[x \* 256 + y];   1. 🡪 (2)，逆旋轉90度   ad\_img[(191 - y) \* 256 + (x - 128)] = img\_lena[x \* 256 + y];   1. 🡪 (1)，正旋轉90度   ad\_img[(y - 192) \* 256 + (255 - x)] = img\_lena[x \* 256 + y]; |

|  |
| --- |
| 1.3 |
| a |
| 1\_3\_a\_out.raw(1\_3\_a\_out.jpg) |
| Disgussion |
| 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。 |

|  |
| --- |
| 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。   1. 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。 2. 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 |
| Disgussion |
| 先創建512x5122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); |