

# Advanced Computer Vision

## HOMEWORK 1

Lab : VPILab

Advisor : Cheng-Ming Huang(黃正民)

Student : Yu Cho(卓諭)

Student ID : 106318025

Data : 2017.10.07

## I. Problem

### A. Image Read/Write

- Write a program to implement "bmp" format image reading and writing.

### B. Image Rotation

- Do a 90-degree clockwise rotation over the input image to generate the output image.

### C. Image Channel Interchange

- Interchange the channels of the rotated image, i.e.,  $R \rightarrow G$ ,  $G \rightarrow B$ ,  $B \rightarrow R$ .

### D. Bonus

- Read and rotate the image with different size. (lena\_cropped.bmp)

## II. Method

### A. Image Read/Write

- C++ Program
  1. 利用 unsigned char 宣告一個一維大小為  $54 + 512 * 512 * 3$  的矩陣，使用 unsigned char 是因為儲存每一個像素點都是 1Byte(0~255bits)，且標頭檔以 Byte 計算。
  2. 利用 File 宣告輸入檔案指標，fopen()開啟二元檔(binary)供程式讀取，fread()將檔案指標內的二元資料讀進一維矩陣，完成讀檔。
  3. 利用 File 宣告輸出檔案指標，fopen()開啟二元檔(binary)供程式寫入，fwrite()將一維矩陣資料寫入輸出檔案指標內容中，完成寫檔。
- C++ Program using OpenCV
  1. 利用 imread() 將原圖讀入由 Mat 宣告的矩陣內，完成讀檔。
  2. 利用 imwrite()將 Mat 格式的矩陣資訊寫入一個自訂檔名的檔案中，完成寫檔。

## B. Image Rotation

### ● C++ Program(順時轉 90°)

1. 宣告三個一維空矩陣，分別為，標頭檔矩陣(54 Bytes)、讀取的原圖像資訊矩陣(512\*512\*3 Bytes)、旋轉後圖像資訊矩陣(512\*512\*3 Bytes)。
2. 將輸入檔案指標的內容分別以標頭檔(一維矩陣)、圖像資訊(一維矩陣)存入空矩陣中。(標頭檔 54 Bytes)
3. 建立三層 for 迴圈，外至內分別為，圖像列數、圖像行數、圖像深度(BGR 順序存入)。
4. 利用像素的取出與填入順序以得到列數、行數、深度的線性方程式。

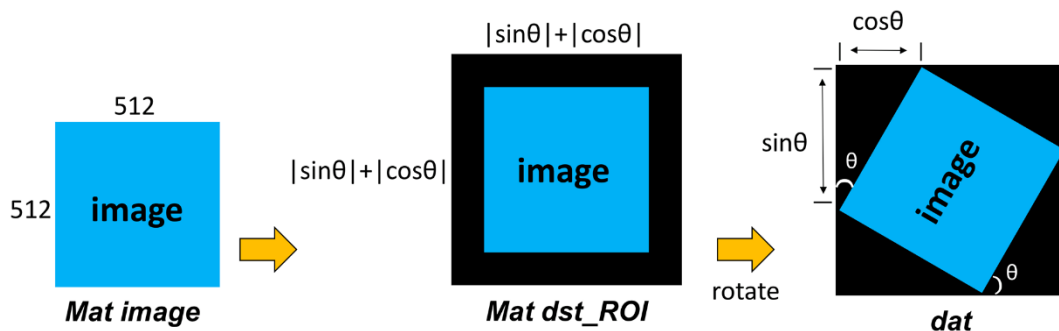
### ● C++ Program using OpenCV (動態旋轉、自訂角度旋轉)

#### 1. 動態旋轉

- i. 利用 Mat 宣告四個矩陣分別為，存放原圖像素資訊(image)、旋轉後應有的矩陣大小(dst)、可將原圖放置輸出矩陣中心的矩陣(dst\_ROI)、旋轉矩陣(rotate)。
- ii. 利用  $|\sin(\theta)| + |\cos(\theta)|$  計算出在相對應角度該有的矩陣長寬值。
- iii. 利用 copyTo() 結合 Rect() 將原圖放置在與 dst 大小相同的 dst\_ROI 中間(以利後續用來設定各個角度在不同大小矩陣中的旋轉中心)。
- iv. grtRotationMatrix2D() 去獲得一個旋轉矩陣
- v. warpAffine() 將影像經由轉矩陣轉換。
- vi. 利用一個 for 迴圈去得到各個旋轉角度的影像以及用無窮迴圈讓其不斷重複。

#### 2. 自訂角度旋轉

- i. 概念與動態旋轉相似，最大差異在於不需要 for 迴圈獲得各旋轉角度的影像。



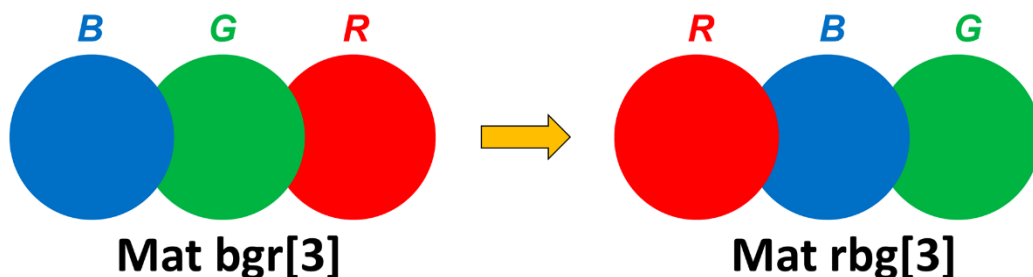
## C. Image Channel Interchange

- C++ Program

1. 宣告三個一維空矩陣，分別為，標頭檔矩陣(54 Bytes)、讀取的原圖像資訊矩陣(512\*512\*3 Bytes)、旋轉後圖像資訊矩陣(512\*512\*3 Bytes)。
2. 將輸入檔案指標的內容分別以標頭檔(一維矩陣)、圖像資訊(一維矩陣)存入空矩陣中。(標頭檔 54 Bytes)
3. 建立三層 for 迴圈，外至內分別為，圖像列數、圖像行數、圖像深度(BGR 順序存入)。
4. 因為此題要以 RBG 順序存入，故在圖像深度的迴圈內，增加一個判斷式，去將像素分別存入正確的通道內。

- C++ Program(using OpenCV)

1. Mat 宣告四個矩陣分別儲存，原圖資訊、BGR 順序圖像深度、RBG 順序圖像深度，影像通道轉換後影像。
2. 利用 spilt()分離原影像通道。
3. 將 BGR 順序換成 RBG 順序
4. 利用 merge()合併通道轉換後的三通道。



## D. Bonus

### ● C++ Program(Read/Write)

1. 利用 unsigned char 宣告一個一維大小為  $54 + 512 * 288 * 3$  的矩陣，使用 unsigned char 是因為儲存每一個像素點都是 1Byte(0~255bits)，且標頭檔以 Byte 計算。
2. 利用 File 宣告輸入檔案指標，fopen()開啟二元檔(binary)供程式讀取，fread()將檔案指標內的二元資料讀進一維矩陣，完成讀檔。
3. 利用 File 宣告輸出檔案指標，fopen()開啟二元檔(binary)供程式寫入，fwrite()將一維矩陣資料寫入輸出檔案指標內容中，完成寫檔。

### ● C++ Program(Rotation)

1. 宣告三個一維空矩陣，分別為，標頭檔矩陣(54 Bytes)、讀取的原圖像資訊矩陣( $512 * 288 * 3$  Bytes)、旋轉後圖像資訊矩陣( $288 * 512 * 3$  Bytes)。
2. 將輸入檔案指標的內容分別以標頭檔(一維矩陣)、圖像資訊(一維矩陣)存入空矩陣中。(標頭檔 54 Bytes)
3. 將標頭檔儲存寬高資訊的矩陣互換，才能儲存正確旋轉後影像。
4. 建立三層 for 迴圈，外至內分別為，圖像列數、圖像行數、圖像深度(BGR 順序存入)。
5. 利用像素的取出與填入順序以得到列數、行數、深度的線性方程式。

### ● C++ Program using OpenCV(Read/Write)

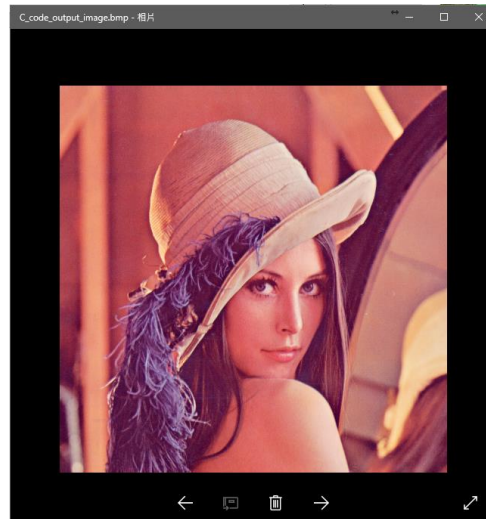
1. 利用 imread()將影像資訊存入由 Mat 宣告的矩陣內。
2. 利用 imwrite()將矩陣內的資料存成\*.bmp。
3. 完成寫檔讀檔。

### ● C++ Program using OpenCV(Rotation)

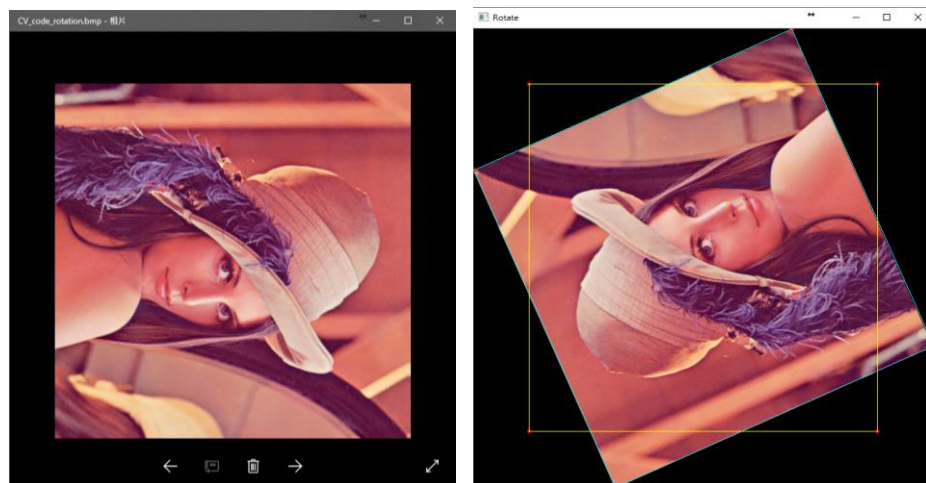
1. 宣告三個矩陣分別為，原影像、旋轉矩陣、旋轉後影像。
2. Point2f()宣告一個影像旋轉中心。
3. getRotationMatrix2D()獲得旋轉矩陣。
4. warpAffine()轉換影像資訊。

### III. Results

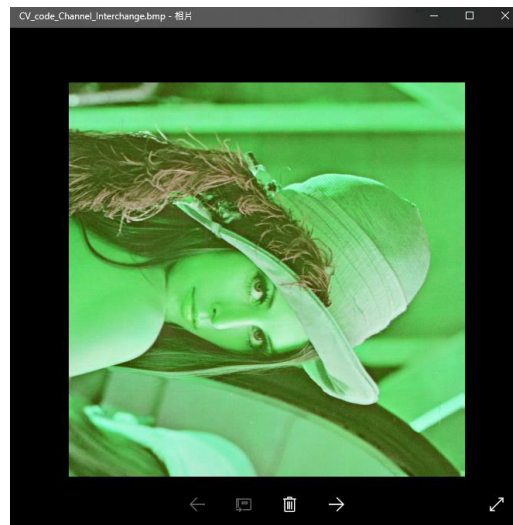
#### A. Image Read/Write



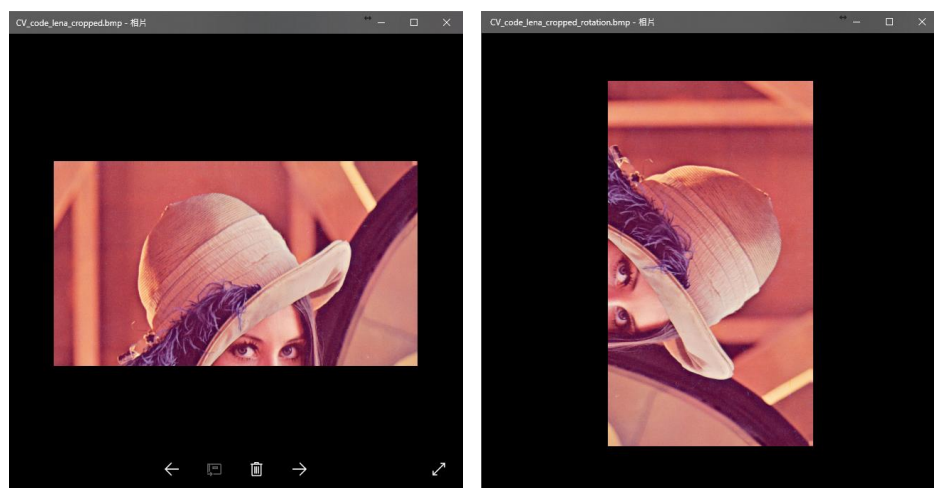
#### B. Image Rotation



## C. Image Channel Interchange



## D. Bonus



## IV. Discussion

### A. Image Rotation

- C++ Program

在做影像處理時，C++語言要把\*.bmp 檔的標頭檔以及影樣資訊分成兩個矩陣，單獨對影像資訊的矩陣處理。存出檔案時也必須將剛分離的標頭檔寫進輸出資料中。處理旋轉時只有在像素點的放置順序要注意，並寫出屬於該旋轉角度的方程式。

- C++ Program using OpenCV

相對於純 C++處理旋轉影像就簡單的多，只需要三行程式碼即可完成 C++需要三層 for 迴圈才能完成的目標。

### B. Image Channel Interchange

- C++ Program

因為通道上的變換，必須調換存取像素的順序(BGR 變成 RGB)，故要在第三層迴圈中下判斷式改變存取順序。

- C++ Program using OpenCV

OpenCV 的 split()、merge()可以將影像通道分離且合併，省去使用迴圈的記憶體空間成本。

### C. Bonus

- C++ Program

標頭檔在不同的圖像大小中，寬、高的資訊要改變否則無法正確顯示圖像。

- C++ Program using OpenCV

旋轉中心需要思考，必須讓旋轉後的圖像能夠在設定好的空矩陣中，若旋轉中心沒有設定正確會發生只顯示部分圖像的問題。