高等影像處理

Homework 1: image read/write and rotation書面報告

學號： 61247051S 姓名： 陳建壹

一、本作業所用之程式語言及編譯器

Language: C#

Compiler: Roslyn

二、程式功能

1. 讀取影像檔案並對影像進行向右、向左旋轉與存檔。
2. 本程式可讀入的影像檔格式包含JPG, PNG, BMP, PPM檔。
3. 本程式可輸出的影像檔格式則為JPG, PNG, BMP, PPM檔。

三、程式流程或演算法

|  |
| --- |
| Image RotateImage(Image input):  // swap w and h  output = new()  output.width = input.height  output.height = input.width    for (y : 0 -> input.height) {  for (x : 0 -> input.width) {  p = input.GetPixel(x,y);  nx = output.height - 1 - y;  ny = x;  output.SetPixel(nx, ny, p);  }  return output  } |

四、測試結果

|  |
| --- |
| 範例一為[向右旋轉]一次之測試結果。左圖為input，右圖為output。由上圖可觀察到草地部分(綠色區塊)方向原本朝上, 處理過後, 草地部分朝右。 |
| 範例一為[向左旋轉]一次之測試結果。左圖為input，右圖為output。由上圖可觀察到天空原本向上,經過處理後, 天空朝向左邊 |
| 範例一為[向右旋轉]兩次之測試結果。左圖為input，右圖為output。由上圖可觀察到人偶正立, 處理過後, 人偶倒置 |

五、程式撰寫心得

這次作業花了一些時間學習設計使用者介面(WPF), 一個滿成熟的GUI framework.

使用xaml進行構圖, 控制項的綁定也很便利。

原本使用內建的System.Drawing.Bitmap就可以對單個pixel進行操作。但是當檔案有一定規模時,效率明顯不好。所以還是決定轉成Byte Array的形式再進行像素操作。然後再進一步封裝成Class RawImage, 包含常用的一些Method(讀\寫, GetPixel, SetPixel等)

之後有機會再進行更進階的優化(使用span<T>)減少memory的reallocate.

有趣的是Bitmap轉成Byte Array的順序是B,G,R,A害我花了一些時間Debug。