電機所控制組 110318051 張皓博

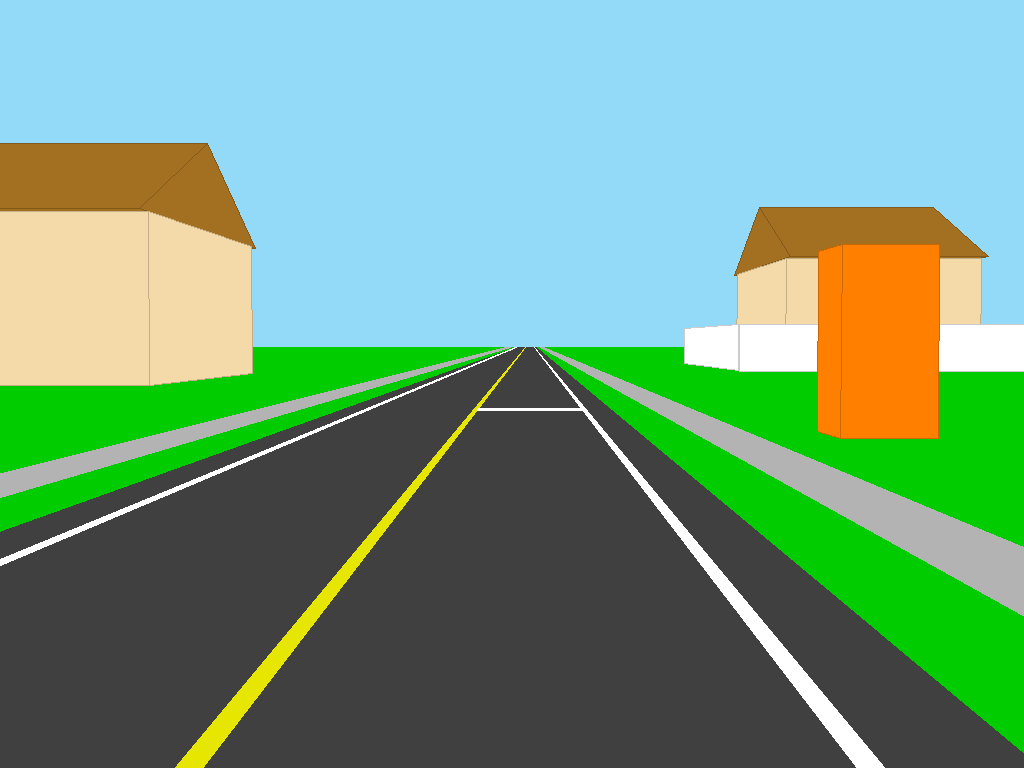
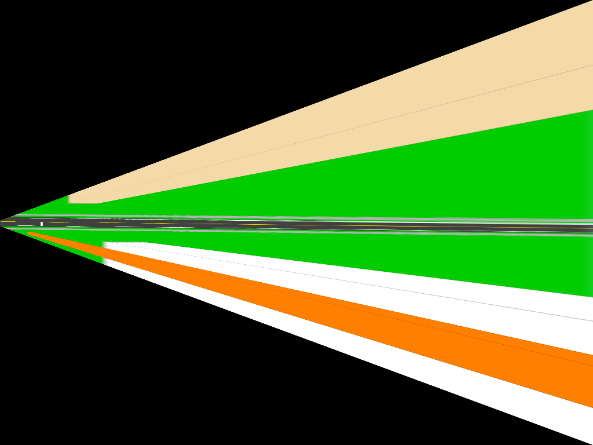
HW3\_opencv講解

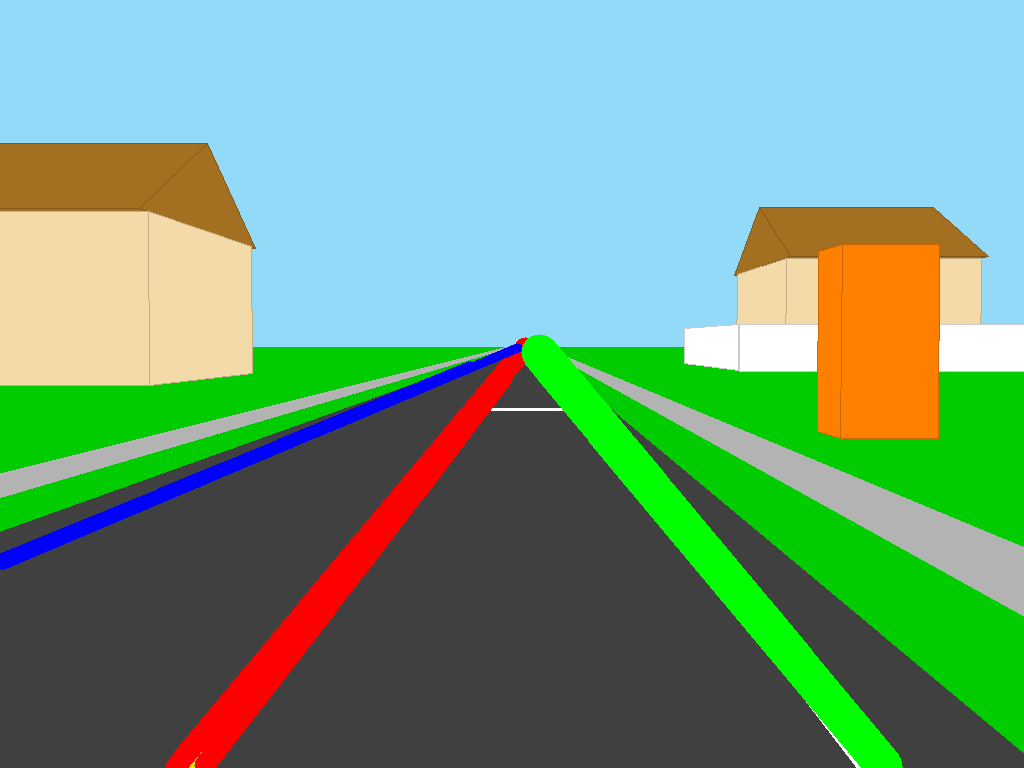
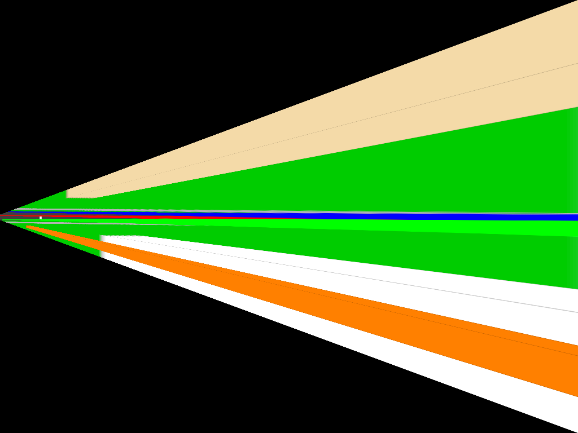
我使用Visual Studio來寫這個C++作業，執行方式為把 .exe檔跟road.bmp放在同個資料夾底下後，執行.exe即可產生產生輸出圖檔。

產出兩張圖：cropImage\_label.bmp(74x74)、cropImage\_normal.bmp(74x74)。

過程圖片(1024x768)：

總共有7個function：

1. **Mat bmpRead(string imageName):**

利用opencv的imread函數讀圖，用以讀取road.bmp，其中有防止讀取圖片為空的措施。

1. **void bmpWrite(char const\* pictureName, Mat OutputImage):**

用以輸出圖片，利用opencv的imwrite函數輸出，可以傳入圖片名稱及圖片的imageData。

1. **Mat rgb2gray(Mat inputImage):**

利用opencv的cvtColor函數，把RGB圖轉為灰階。

1. **Mat oneChannelstoThreeChannels(Mat inputImage):**

因為opencv的rgb2gray會把channel變為1，所以宣告一個Mat的vector，再將三層一樣的Mat merge在一起，使one Channel到three Channels。

1. **Mat LabelLines(Mat inputImage, Mat grayImage):**

用以畫出三條標線，先經過opencv的高斯模糊處理(GaussianBlur)、Canny的邊緣偵測(Canny)、再設定一個多角形的遮罩把標線以外的區塊先用黑色蓋掉後(fillPoly)，利用opencv的霍夫找線(HougjLinesP)找出標線的兩點，利用斜率作為區分三條線的標準，分別上色；由於線條如果太細，再轉換到IPM時會產生斷點，故意畫得很粗。

1. **Mat warping(Mat inputImage, int width, int height):**

先標好原圖需要轉換的圖片座標，以及轉換後的期望的座標，

A→A’、 B→B’、 C→C’、 D→D’，經由opencv的getPerspectiveTransform得到轉換矩陣，再代入opencv的warpPerspective做出IPM image。

1. **Mat crop(Mat inputImage):**

**利用**Rect框出要crop的方框座標，再直接輸出到outputImage。

先寫handmade後再來寫opencv真的輕鬆非常多，內含有warpPerspective、Gaussianblur、Canny、HouLinesP、line等等包好的function，不過最輕鬆的地方就是Mat在處理bmp時不需要自行調整header，不像在寫handmade的時候為了不想調整header使用raw檔寫作業，但在讀取大小不同的raw檔時每次都要重新設定圖片的width和height，非常麻煩。