電機所控制組 110318051 張皓博

HW4\_opencv講解

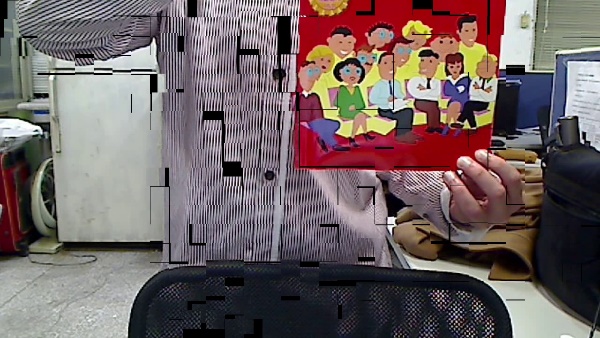
我使用Visual Studio2019 + 北科i學園提供的opencv4.4.0來寫這個C++作業，執行方式為把 .exe檔跟HW4\_video.avi放在同個資料夾底下後，執行.exe即可產生產生輸出圖檔即command資訊。

產出十二張圖：

reconstructFrame\_0th.jpg reconstructFrame\_30th.jpg

reconstructFrame\_60th.jpg reconstructFrame\_90th.jpg

reconstructFrame\_120th.jpg reconstructFrame\_150th.jpg

opticalFrame\_0th.jpg opticalFrame\_30th.jpg

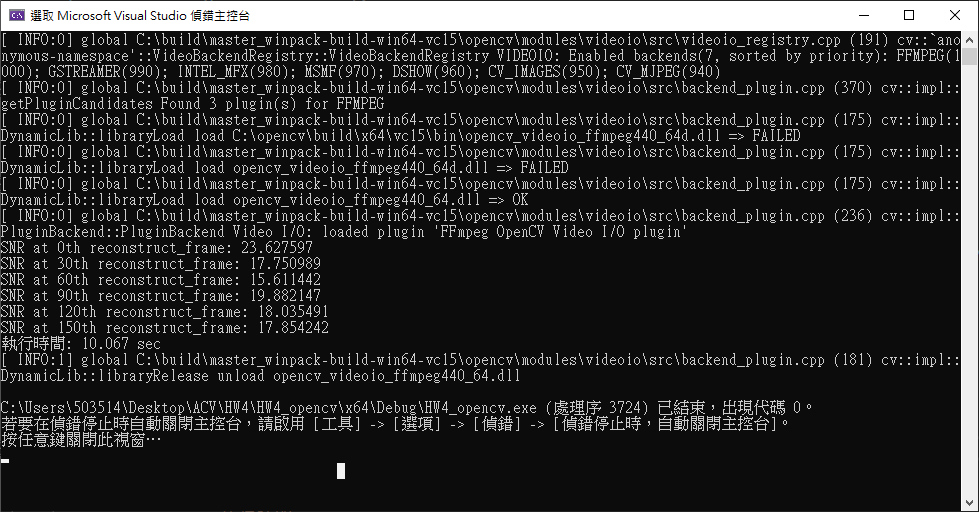
opticalFrame\_60th.jpg opticalFrame\_90th.jpg

opticalFrame\_120th.jpg opticalFrame\_150th.jpg

Command資訊(執行時間為同時做完motion vector reconstruct跟optical flow):



最一開始用opencv的VideoCapture讀取影片，再用.read提取frame，輸入兩個frame到我寫好的function作運算。

總共有4個function：

1. **Mat motionEstimation(Mat first\_frame, Mat second\_frame):**

根據題目描述，將frame切成多塊40x40的小template，再從上下左右延展20 (在80x80的框框中尋找下一個frame的template位置)；把frame轉為灰階後在運算上更為快速，我把所有的切塊存進一個vector裡，再傳入frameReconstruction()做比對並且重建frame。

1. **Mat frameReconstruction(vector<Mat> rgbBlocks40\_40, vector<Mat> grayBlocks40\_40, vector<Mat> grayBlocks80\_80, Point frameSize):**

將搜尋範圍80x80的image跟40x40的template傳入opnecv的matchTemplate，再藉由minMaxLoc篩選出相似度最高的座標位置，利用這個座標重建新的frame。

1. **Mat denseOpticalFlow(Mat previous\_frame, Mat next\_frame):**

利用opencv的calcOpticalFlowFarneback去計算兩張frame的光流，再將得到的資訊建構成hsv寫出來後，再轉回我們眼中看的RGB圖。

1. **double getSNR(Mat original\_frame, Mat reconstruct\_frame):**

將原圖與重建後的圖做SNR計算。



這次的作業相對來說比較好想一點，不過雖然我們能利用matchTemplate算出motion vector來做到影片壓縮的演算法，但我並不知道如何只利用第一個frame加上後面的motion vector就輸出回影片，單純將影像重建之後一張一張frame寫回影片並沒有壓縮的效果，甚至比原本的還大。