視訊通訊作業二 : Motion Estimate Evaluation

B04902021 資工三 陳弘梵

一、 編譯方式、環境

主程式使用 C++寫成，包含<cstdio, cstdlib, cstring, cmath>等函式庫，編譯時使用 gcc version 6.3.0，平台為 windows10。

二、 執行方式

我將執行檔以「> blockXX_stepXX.csv」的方式匯出計算之各項演算法 PSNR 值。

【額外程式】將上述的 CSV 檔案，以 python3.6 撰寫的程式進行繪圖，使用工具包含 numpy, matplotlib, pandas 等 module，再將資料整理予以比較。此部分只需更改檔案中「blockXX_stepXX」為對應的 CSV 檔案名稱即可。

三、 採用演算法

在設定各種大小的 macroblock 大小後，以 pixel 為單位，用不同的 step 大小計算：

(1) Full Search: 一定範圍內的完全搜索。
(2) Three Step Search: 九宮格的點搜索，範圍逐漸縮小。
(3) Orthogonal Search: 直行、橫排的三點搜索，直到 step=1。
(4) Cross Search Algorithm: 斜角上叉字型的五點搜索，step=1 且 best matching block 在左下右上時，轉為十字搜索最後一輪。
(5) Two logarithm Search: 不停的十字型搜索，直到 step=1 時，如同 three step search 搜索最後的九宮格。

四、 PSNR 與 Frame Chart

以下四個表格分別為四張 QCIF、CIF 圖的成果繪製，大致表現如下：

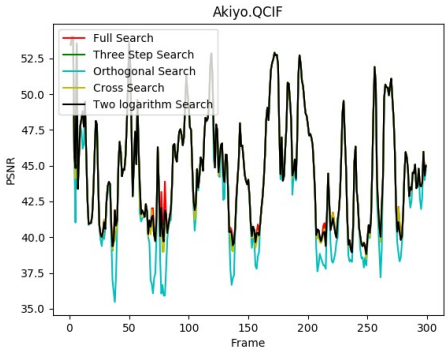
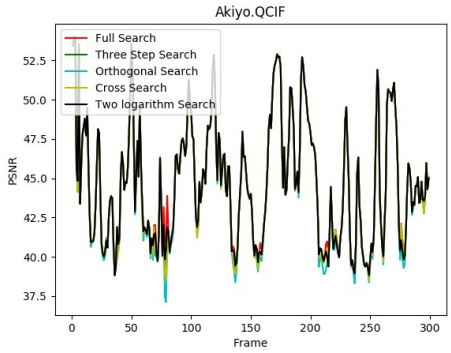
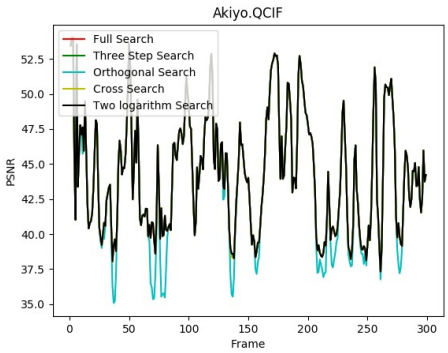
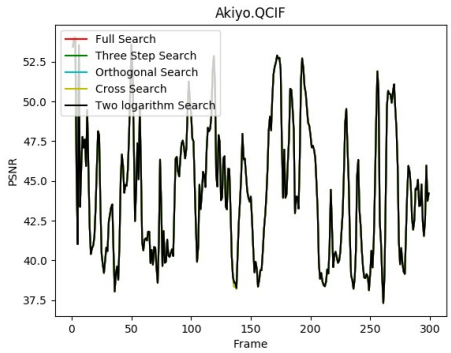
(1) 整體而言的 Motion Estimate 成功度，psnr:

Akiyo > Foreman > Table > Stefan.

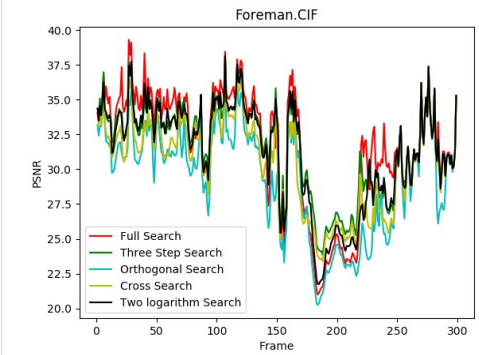
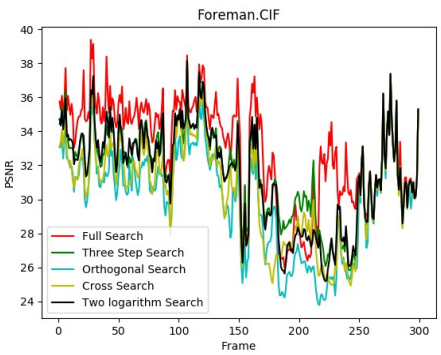
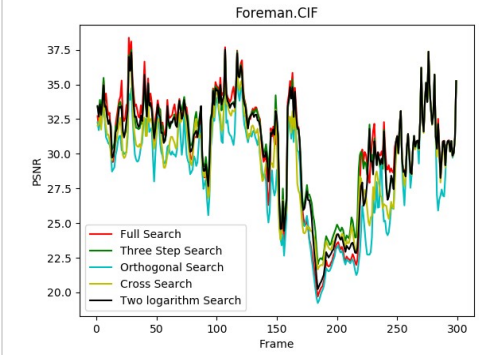
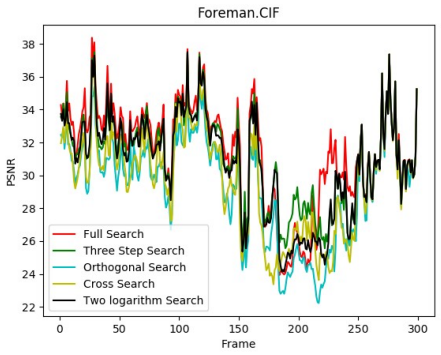
(2) 各種組合下差異最大的為 Block size=8, Step size=8 的設定，它會將 full search 較其他演算法的優勢拉到最大。
(3) 演算法大體上的 estimation 優劣為：
Full Search > Three Step Search ≒ Two Logarithm Search > Cross Search > Orthogonal Search

(4) 然而在運行時間上也有 trade-off，如 orthogonal 可以在五種當中，以最短的時間完成，Full 卻需要最長的時間、複雜度最高的運算。其中令人比較驚奇的是 TLS，它整體而言運算是完全少於 three step 的，約莫少了一半的複雜度，但是它在算到最後一個 step 時，會將四周九宮格內的 Pixel 全部爬過一遍，就能讓他達到與 three step 相當的 evaluation 水準，很神奇！

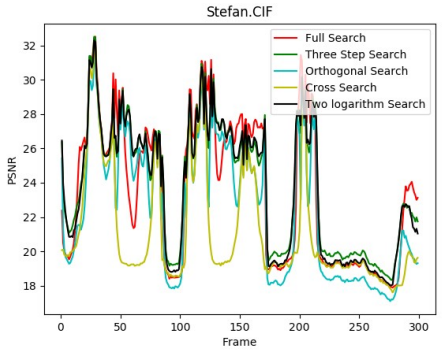
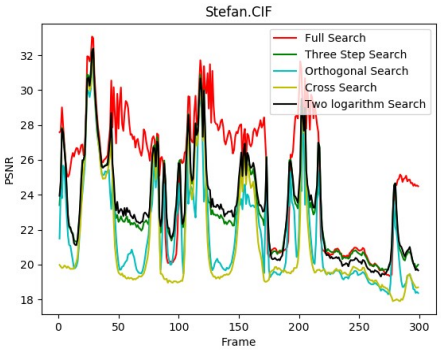
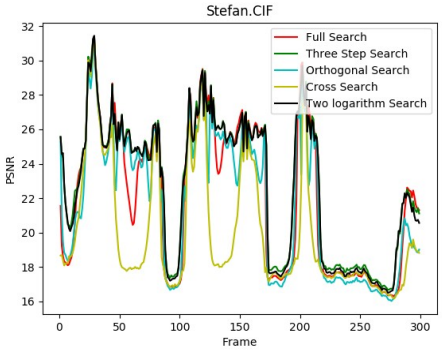
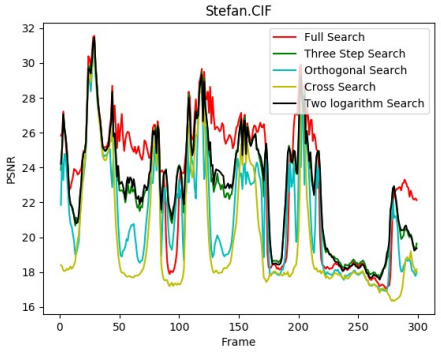
(5) 影片性質的差異會嚴重影響到各個演算法的表現，甚至 macroblock, step 大小等等更會直接造成 motion vector 的生成是否可用或是失敗，因此我們所能參考的各項通用格式規定，都是前人辛苦研究出來的較佳方式，在採用之餘，也可以思考他們發現、發明這些格式的動機。

Step size \ Block size	**4**	**8**
8		
16		

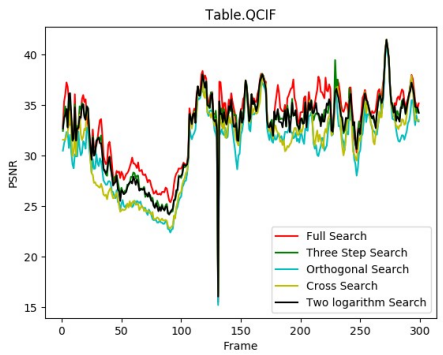
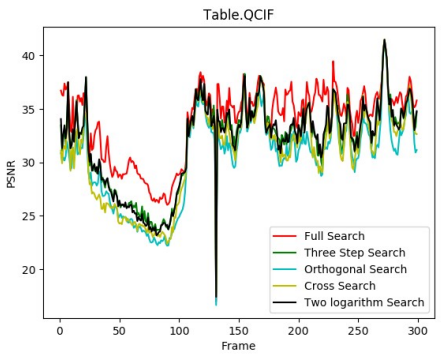
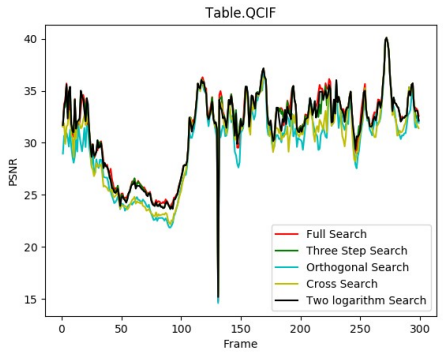
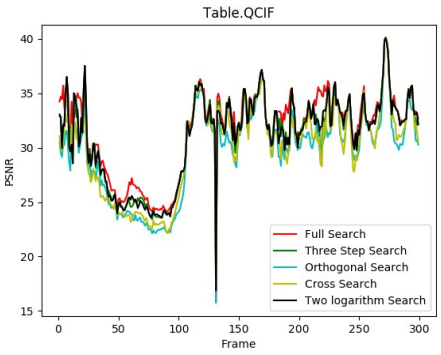
Akiyo↑

8		
16		

Foreman↑

8		
16		

Stefan↑

8	 Table.QCIF PSNR Frame Full Search Three Step Search Orthogonal Search Cross Search Two logarithm Search	 Table.QCIF PSNR Frame Full Search Three Step Search Orthogonal Search Cross Search Two logarithm Search
16	 Table.QCIF PSNR Frame Full Search Three Step Search Orthogonal Search Cross Search Two logarithm Search	 Table.QCIF PSNR Frame Full Search Three Step Search Orthogonal Search Cross Search Two logarithm Search

Table↑