

多媒體技術概論 Assignment3 Report

SAD Comparison(Block Size 8x8, Search Range 8, Case 1)

Frame #	Full Search	2D-Log
2	552444	836685
3	711380	977588
4	832647	906138
5	888334	1249290
6	968370	1448579
7	1038465	1552033
8	1127275	1738578
9	1346603	1894693
10	1784883	1980969
11	1983389	2087754
12	2124979	2195238
13	2227803	2269305
14	2325517	2376401
15	2408498	2428064
16	2477064	2498157
17	2531643	2533855
18	2576992	2575090
19	2612881	2615920
20	2646607	2638367

(Block Size 8x8, Search Range 8, Full Search)

Frame #	Case1	Case2
2	552444	552444
3	711380	711380
4	832647	832647
5	888334	888334
6	968370	968370
7	1038465	1038465
8	1127275	1127275
9	1346603	1346603
10	1784883	1784883
11	1983389	
12	2124979	674420
13	2227803	816031
14	2325517	887712
15	2408498	903125
16	2477064	977071
17	2531643	1097474
18	2576992	1294834
19	2612881	1635759
20	2646607	1789155

左表為兩種 motion estimation 處理 case1 後各 frame 的 SAD，原本認為 2D-log search 的準確度比 full search 低很多，所以 2D-log 的 SAD 應該要比 full search 的 SAD 高，結果雖然也如同預期，但卻發現 full search 的 SAD 上升的速度比 2D-log 還快，且在 frame 20 的時候超越了 2D-log。

右表為利用 full search 處理兩種 case，因為 case2 有兩個 reference frame(frame 1 & frame 11)，所以在 frame 12 的時候 SAD 會突然下降。Frame 2~10 是以 frame 1 為 reference frame，故其 SAD 是相對於 frame 1 所計算；而 frame 12~20 則是以 frame 11 為 reference frame，其 SAD 是相對於 frame 11 做計算。

Block Size Comparison (Range 8x8, Case 1, Full Search)

caltrain002

blockSize 8x8

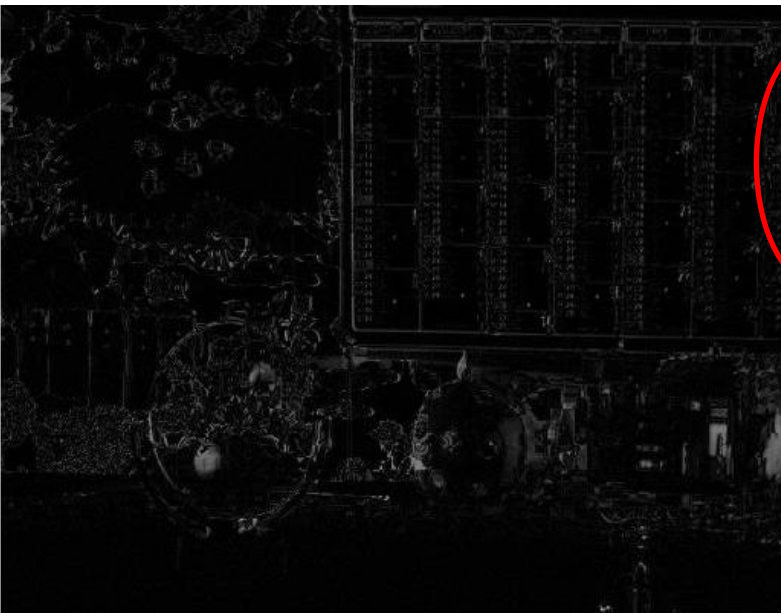
blockSize 16x16



caltrain010

blockSize 8x8

blockSize 16x16



Block size 會影響取樣的精準度，size 越小相當於取樣越多次，則壓縮後的精準度相對較高。由 caltrain002 兩張圖可以看見，16x16 的 residual image 比 8x8 的 residual image 有較多白色部分，即是比對後的誤差所在，而下兩張 caltrain010 因為縮小了較不易察覺，但可留意圈起來的部分，是差異較明顯的部分。由此可見，block size 越小，取樣次數越多，誤差則越小。

Search Range Comparison (Block Size 8x8, Case 1, Full Search)

caltrain002

searchRange 8

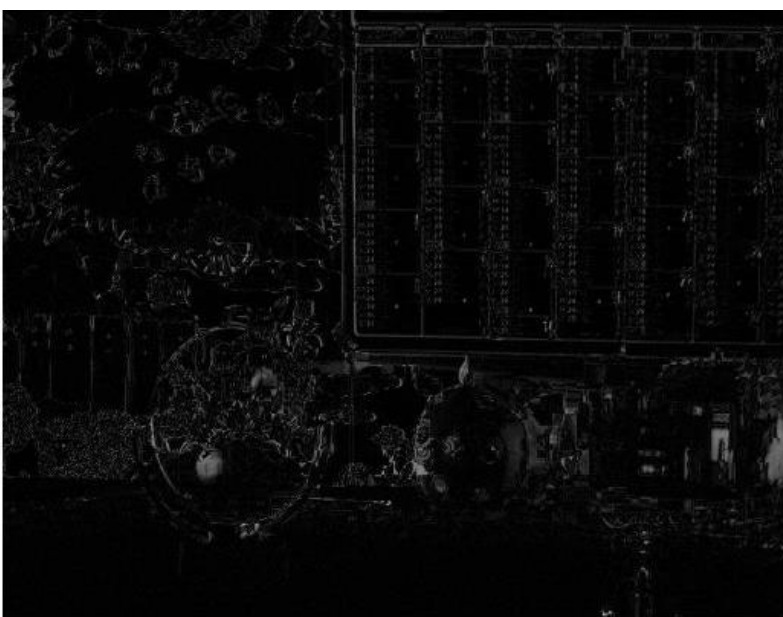


searchRange 16



caltrain010

searchRange 8



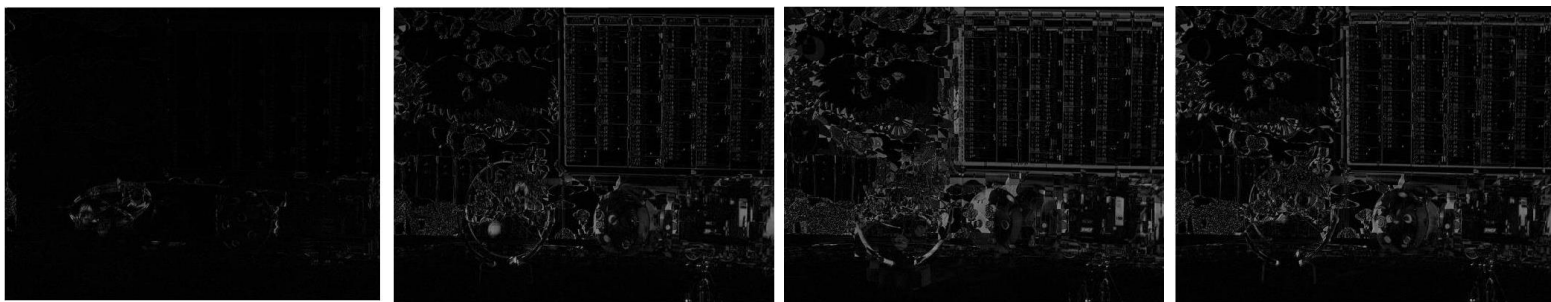
searchRange 16



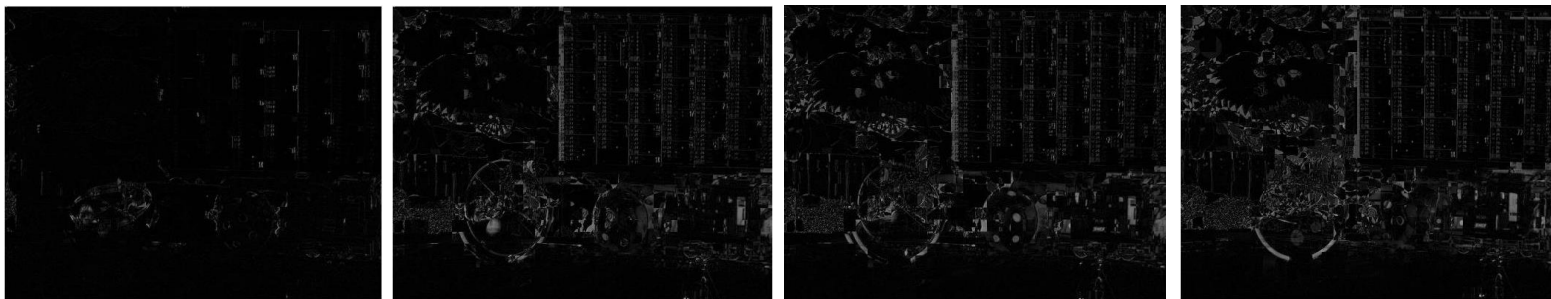
Search range 越大，則找到誤差小的 reference block 的機率更高，caltrain002 的差異過小，但 caltrain010 的差異就很明顯，range 只有 8 的話，比較無法找到符合的 block，故產生的誤差較多，即圖中白色部分；相較之下，range 為 16 的話，能夠選擇的 block 較多，因此找到符合的 block 的機率較 range 為 8 來的高，故 residual image 白色部分較少。

Motion Estimation Method (Block Size 16x16, Search Range 8, Case 1)

Full search, caltrain002, 010, 012, 020

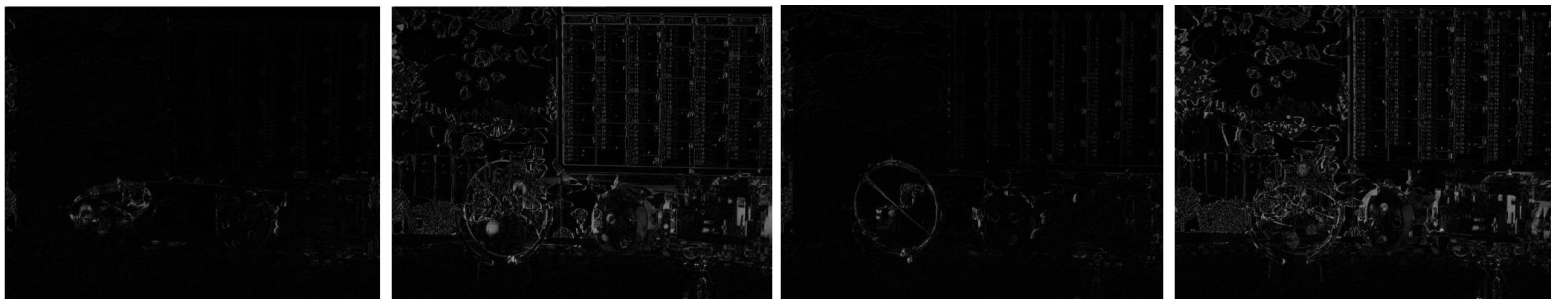


2D-log search, caltrain002, 010, 012, 020

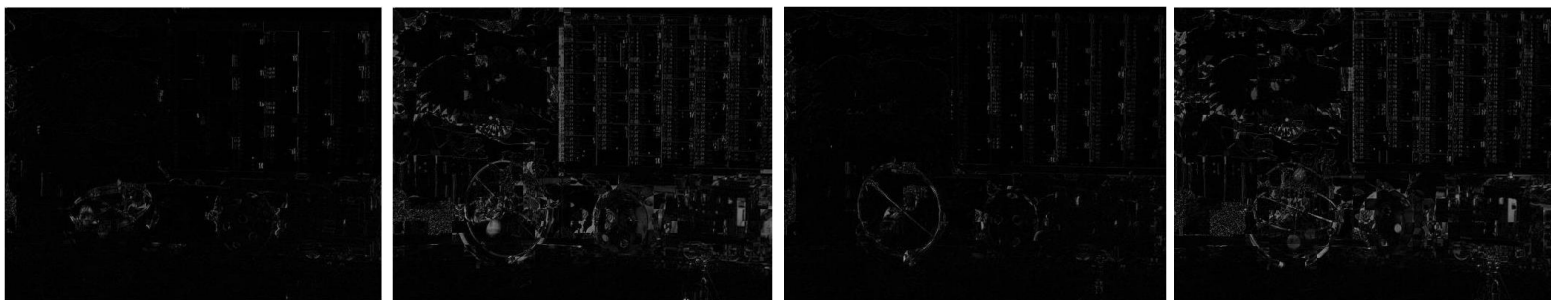


Motion Estimation Method (Block Size 16x16, Search Range 8, Case 2)

Full search, caltrain002, 010, 012, 020



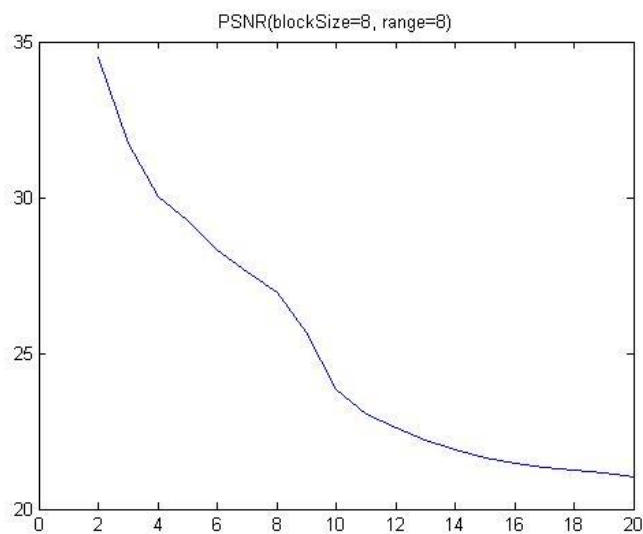
2D-log search, caltrain002, 010, 012, 020



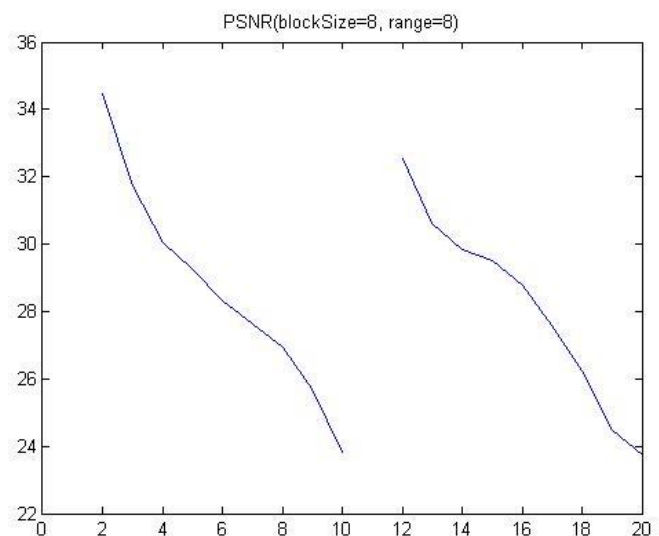
因為 full search 找了在 search range 中所有的 reference block，比對的次數較多，樣本也較多，可以看到其白色的誤差部分明顯少於 2D-log 的結果，以時間換取精準度；2D-log 方法能搜尋到的樣本非常少，符合最小誤差的 block 可能無法被順利找到，因此造成的誤差多很多，但若不需要極精細的比對，2D-log 仍然有其一定的效果，且在速度上遠勝過 full search。

PSNR (Block Size 8x8, Search Range 8)

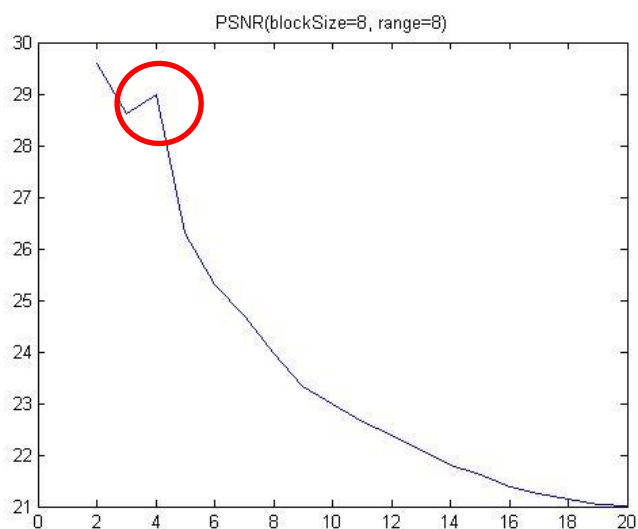
Full search, case1



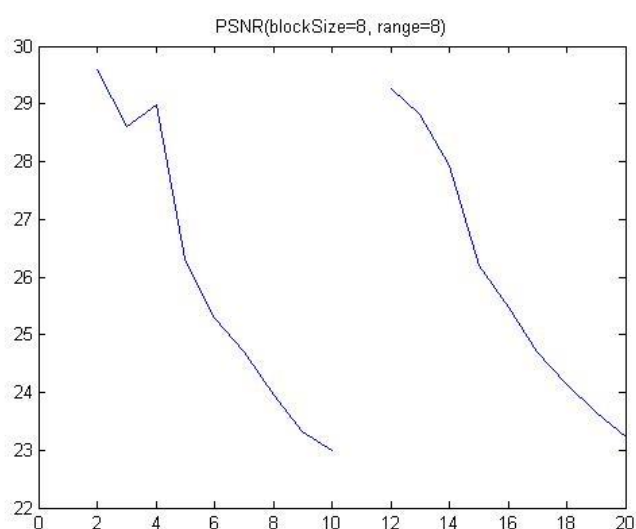
Full search, case2



2D-log, case1



2D-log, case2



PSNR 值越高則代表期受到雜訊的影響越少，誤差越低。隨著 frame 的順序，越後面的 frame 與 reference frame 的差異越大，誤差越多，因此 PSNR 隨著 frame 的順序越後面越低；

造成 case 2 圖中斷層的原因則是因為從 frame 12 開始都以 frame 11 作為 reference frame，而與 frame 11 最相近的即是 frame 12，因此 PSNR 又再次回到極大值，然而後面也隨著離 frame 11 越遠而 PSNR 越低。

Full search 的搜尋較為完善，且搜尋範圍的重疊性高，故找到的 reference block 的 SAD 連續性較高，PSNR 會較圓滑；而 2D-log search 省略許多樣本，離散性較高，應該是造成圖中紅色圓圈中情形的原因。

Time Complexity (Case 1)

Full Search

BlockSize, Range	8x8, 8	8x8, 16	16x16, 8	16x16, 16
Time(sec)	78.821180	287.460414	81.808992	22.682325

2D-log Search

BlockSize, Range	8x8, 8	8x8, 16	16x16, 8	16x16, 16
Time(sec)	17.066461	18.669800	4.711505	5.36954

此處時間計算不包括計算 MSE、PSNR、處理圖表，以及其他輸出，僅僅計算尋找所有 estimate frames 之時間。

Block Size : BxB, Search Range : R, Frame Size : MxN (M<N)

Full Search Time Complexity :

對於每個 block，都要在(2R+1)x(2R+1)的 search range 裡面對於每個 reference block 做比對。

$$T = O\left(\frac{MN}{B^2} \left(\frac{(2R+1)^2}{B^2}\right) B^2\right)$$

Diagram annotations for Full Search:

- Frame 中的 block 數 points to $\frac{MN}{B^2}$
- Search Range 中的 block 數 points to $\frac{(2R+1)^2}{B^2}$
- 對每個 pixel 做運算 points to B^2

2D-log Search Time Complexity :

對於每個 block，都要在(2R+1)x(2R+1)的 search range 裡面，每次找五個 reference block 做比對，而最後一次找八個 block 做比對。從 frame 最左上角的 block 開始，可以保證以新的中心點開始找時，上面以及左邊的點都已經被找過，所以搜尋方向必定往右方及下方擴展。而最遠路徑即是從 frame 左上角 block 以 zigzag 方式走到 frame 的右下角 block。

$$T = O\left(\frac{MN}{B^2} \left(\frac{5(2R+1)}{\log_2 R} \cdot 2 + 8\right) B^2\right)$$

Diagram annotations for 2D-log Search:

- Frame 中的 block 數 points to $\frac{MN}{B^2}$
- Search Range 中搜尋的 block 數 points to $\frac{5(2R+1)}{\log_2 R} \cdot 2 + 8$
- 對每個 pixel 做運算 points to B^2
- 最後一次要找周圍八個 block points to the constant 8 in the formula

Execution

執行方法：

輸入 `hw3(blockSize, searchRange, caseNum, searchMode)`

case : 1 則執行所有 frame 皆以第一張圖為 reference 。

case : 2 則執行第一張籍第十一張街為 reference 。

searchMode : 'full' 則以 full search 演算法執行 。

searchMode : '2d' 則以 2D-log 演算法執行 。

執行結果：

所有 frame 的 residual image 。

PSNR 對 frame 之對應圖 。

在 command window 中輸出每個 frame 的 SAD 。

圖檔名稱：

`blockSize-searchRange-caseNum-searchMode-frameNum.jpg`

Ex: 8-8-1-f-2 即為 size 8x8, range 8, case 1, full search, caltrain002 。