CS3570 Introduction to Multimedia Homework #3

## Q1. the full search and the 2D logarithmic search method.

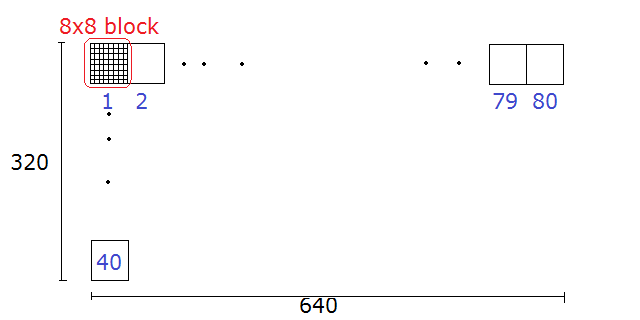
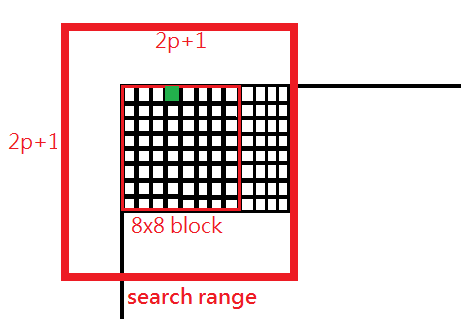
我寫了兩個function分別叫fullsearch()和logarithmic2D ()，功能都一樣是輸入Reference image ”Rimg”, Target image ”Timg”, range ”p”, macroblock size ”N”，就會輸出相應的motion vector跟最小的SAD。

以下詳述這兩個函式的實作方式

full search method

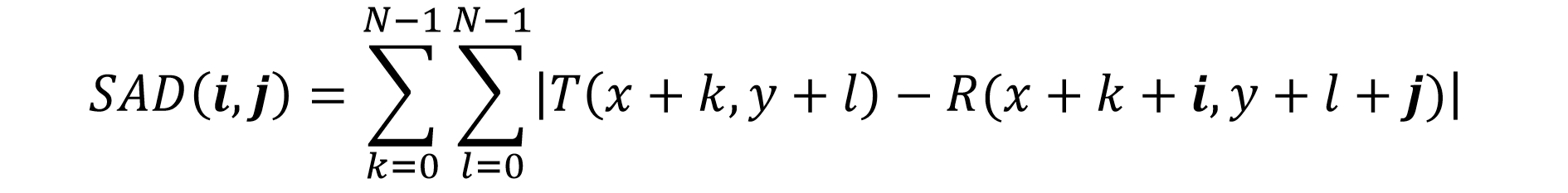
顧名思義就是搜尋遍整張圖，尋找最小SAD的block。

以p=8 N=8為例

作法：

Macroblock 的位置會在Timg上從1~80由左而右移動，第1列結束後再到第2列，一直到第40列，由左而右由上而下。每一個macroblock都會從Rimg左上頂點(x,y)、或中心(x+N/2,y+N/2)，左右延伸p的search range中尋找和Rimg上和Timg最小的SAD值。找到後將Rimg設為predictImg或是做motion compensation。根據講義上的公式



• 𝑁 is size of macroblock window typically (16 or 32 pixels), we have N=8

• (𝑥, 𝑦) is the position of the target macroblock 𝑇, and 𝑅 is the reference region to compute the SAD

• 𝑇 (𝑥 + 𝑘, 𝑦 + 𝑙) — pixels in the macroblock with upper left corner (𝑥, 𝑦) in the target

• 𝑅 (𝑥 + 𝑘 + 𝑖, 𝑦 + 𝑙 + 𝑗) — pixels in the macroblock with upper left corner (𝑥 + 𝑖, 𝑦 + 𝑗) in the reference

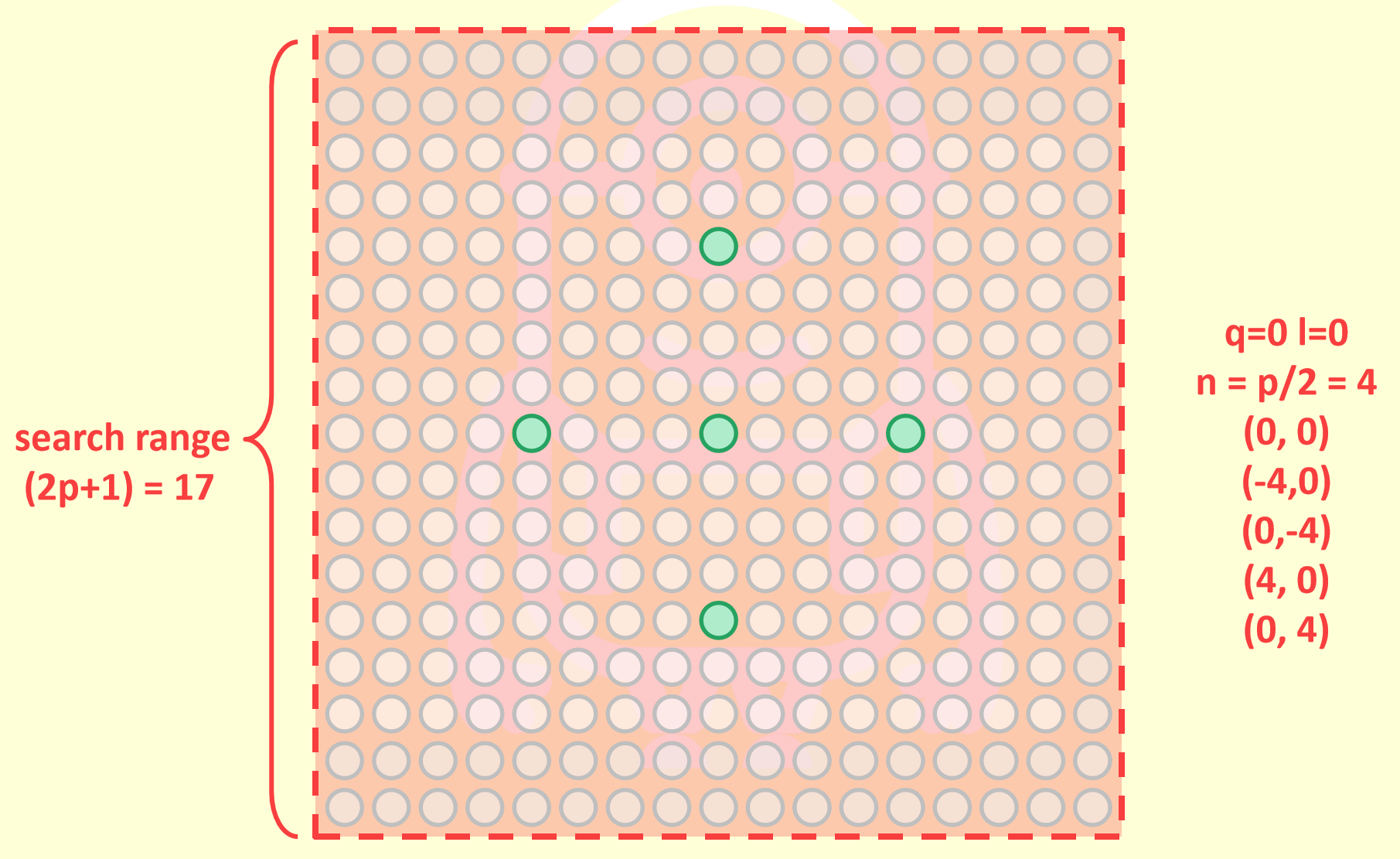
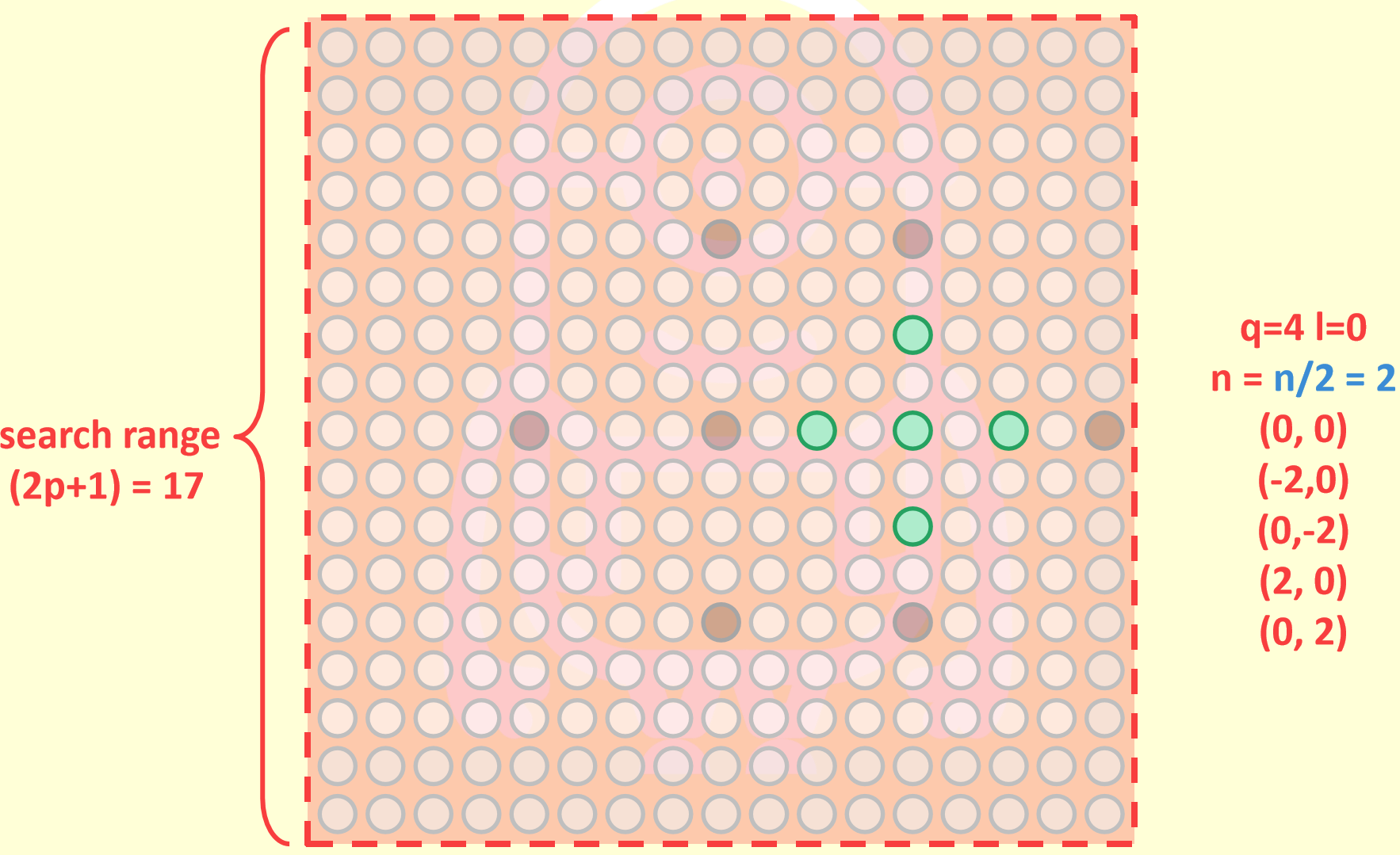
但這個是單通道的做法，我們RGB有三個通道，只要用sum(SAD,3)把三個維度的值都加起來即可。

最後把算完的target和predicted target image相減取絕對值然後3 channel相加即可得到residual image。

2D logarithmic search method

和full search比，只搜尋圖片中的小範圍，有可能找到的最小SAD不是全部裡面最小的，只是相對較小的而已。

以p=8 N=8為例

作法：

Macroblock 的位置和fullsearch一樣，會在Timg上從1~80由左而右移動，第1列結束後再到第2列，一直到第40列，由左而右由上而下。接下來每個block中所做的就不太一樣了，在上下左右四方search range一半位置的四個點，分別找出四方中，有最小SAD的方位，朝那個方位前進，把search range再次縮小一半，尋找四方中最小的值，直到縮小的範圍剩下一個pixel為止，那個點即是我們要的。重複上述步驟，直到整張圖的macroblock都找到相應的最小SAD block為止。

1. Timg= frame73.jpg

|  |  |
| --- | --- |
| Fullsearch | 2D logrithmic |
|  |  |

1. Timg= frame81.jpg

|  |  |
| --- | --- |
| Fullsearch | 2D logrithmic |
|  |  |

1. the total SAD values

fullsearch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. target image: frame73.jpg | | 1. target image: frame81.jpg | |
| P=8 | P=16 | P=8 | P=16 |
| N=8 | 3027.93 | 2972.76 | 11693.7 | 9041.9 |
| N=16 | 3516.97 | 3497.65 | 13903.8 | 11607.9 |

2D logrithmic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. target image: frame73.jpg | | 1. target image: frame81.jpg | |
| P=8 | P=16 | P=8 | P=16 |
| N=8 | 6166.15 | 6123.01 | 17545.4 | 16776.3 |
| N=16 | 6395.39 | 6395.39 | 19426.1 | 18978.1 |

結論：固定相同的range→ block size越小，SAD越小。若range越大，比對的block數量越多，越能找到較符合的frame，但frame被分成較少所以block解析度較低。

1. PSNR

fullsearch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. target image: frame73.jpg | | 1. target image: frame81.jpg | |
| P=8 | P=16 | P=8 | P=16 |
| N=8 | 84.4337 | 84.6435 | 72.6421 | 75.4455 |
| N=16 | 82.6637 | 82.7005 | 71.4191 | 73.3354 |

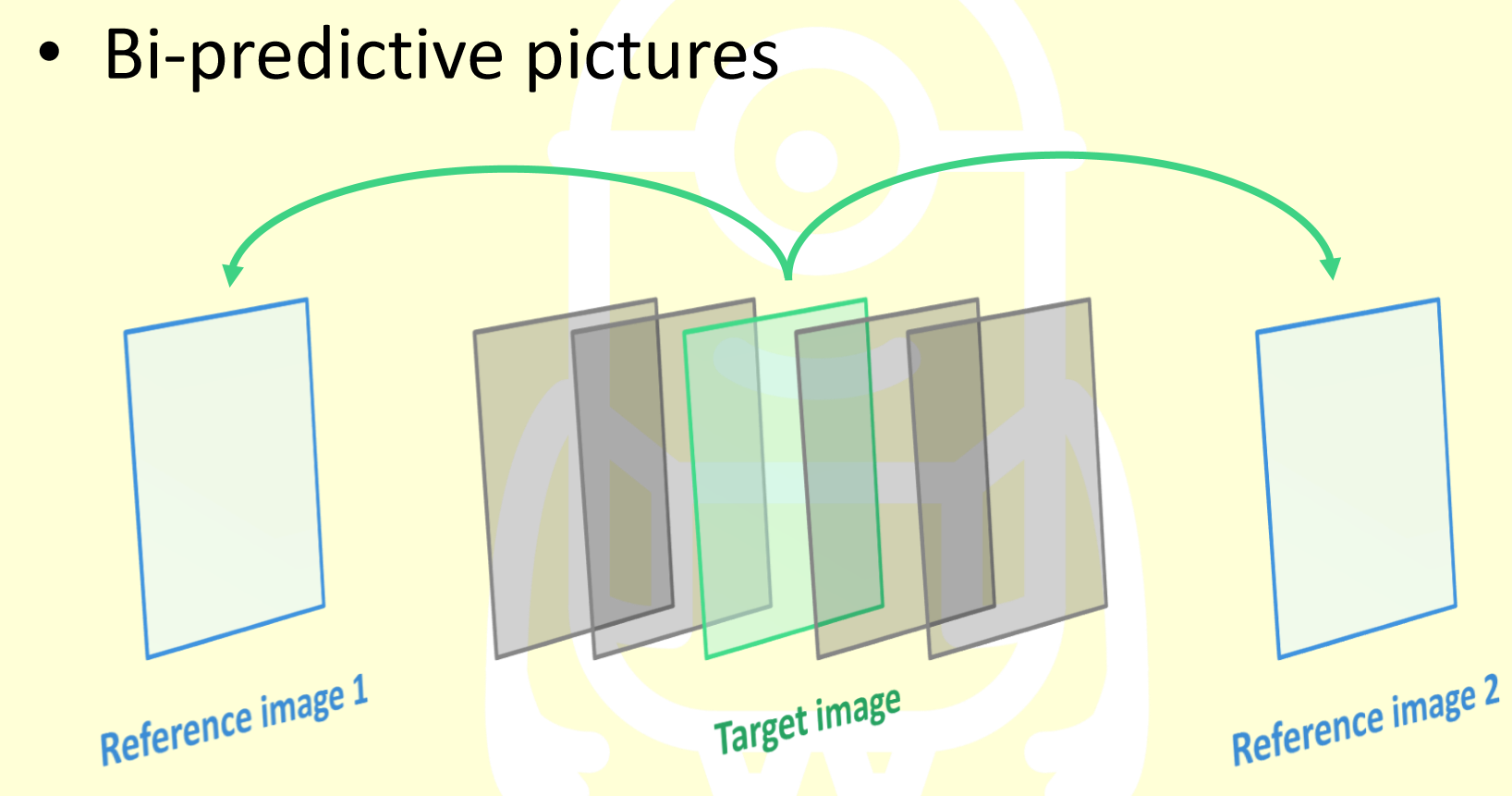
2D logrithmic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. target image: frame73.jpg | | 1. target image: frame81.jpg | |
| P=8 | P=16 | P=8 | P=16 |
| N=8 | 75.506 | 75.0874 | 68.0163 | 67.883 |
| N=16 | 75.0124 | 74.4391 | 68.0036 | 67.7854 |

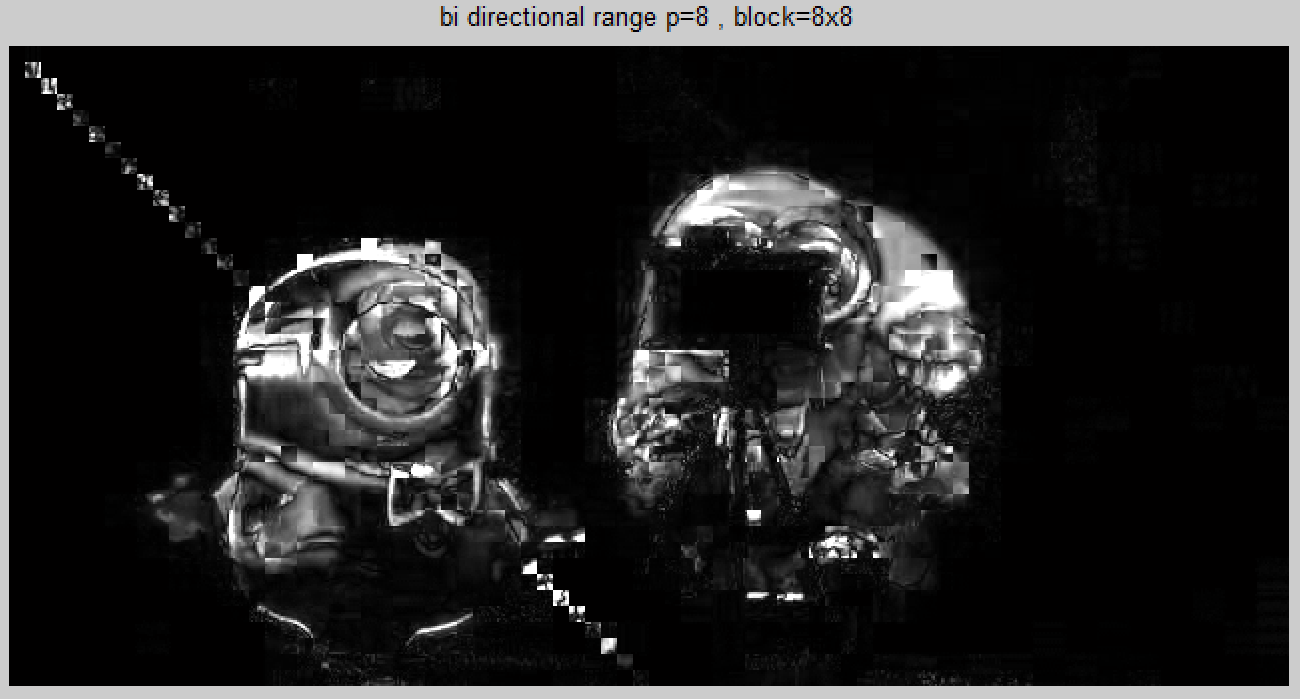
結論：固定相同的range→ block size越小，PSNR越大。若range越大，PSNR也越大，這是因為PSNR算出的結果，和誤差的平方成反比，誤差越多PSNR越小，而range越大，越有可能找到誤差更小的SAD。

## Q2. bi-directional prediction

依照助教給的資料以及網路上查詢的資訊顯示，我們要做的應是同時對兩張Rimg做2D log搜尋，取出兩者其中擁有最小SAD的block的平均。我利用Q1所寫的logrithmic() fuction 來改寫，同時進行兩個Rimg的比對。



1. residual images



1. Comparison

Bi directional(1)Time : 1.3125 (2)SAD : 17655.4 (3)PSNR : 69.8845

2D logarithmic(1)Time : 1 (2)SAD : 6166.15 (3)PSNR : 75.506

理論上，Bi directional的誤差應該要較小，由此可見，他花的時間稍多一些，SAD卻比較大，PSNR也較小，跟我原先預期的結果不太一樣，可能是我的實作方式尚有未修正的缺失。

## Q3. Analyze the time complexity

1. Search Time Complexity

假設兩張圖的大小皆為m\*n, range= p, block size= N

* Full search m\*n\*(2p+1)\*(2p+1)\*N^2 → O(4mn p^2 N^2)
* 2D logarithmic search 每次都篩掉所剩範圍的一半→ O(mn lg(2p+1)N^2)

1. Execution Time

fullsearch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. target image: frame73.jpg | | 1. target image: frame81.jpg | |
| P=8 | P=16 | P=8 | P=16 |
| N=8 | 9.59375 | 39.8281 | 10.0156 | 40.8281 |
| N=16 | 3.01563 | 14.125 | 3.25 | 11.7813 |

2D logrithmic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. target image: frame73.jpg | | 1. target image: frame81.jpg | |
| P=8 | P=16 | P=8 | P=16 |
| N=8 | 1 | 0.21875 | 0.671875 | 0.96875 |
| N=16 | 0.359375 | 0.328125 | 0.203125 | 0.265625 |

1. Compare and discuss the execution time with the theoretical time complexity.

理論上

O(4mn p^2 N^2)> O(mn lg(2p+1)N^2)

實際上

結果A和B小題有相同的結果，當N變大、search range p變大，時間會增加。

在相同的情況下2D logrithmic search 比 full search 快上許多。

## Reference

<http://www.mathworks.com/matlabcentral/fileexchange/44244-subpixel-motion-estimation-without-interpolation?focused=3800636&tab=function>

<https://en.wikipedia.org/wiki/Block-matching_algorithm>

<https://www.ece.cmu.edu/~ee899/project/deepak_mid.htm>

<http://eeweb.poly.edu/~yao/EE4414/HW5_solution.doc>

<https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node252.html>

<https://www.researchgate.net/post/How_do_I_get_PSNR_of_an_image_in_MATLAB>

<https://www.mathworks.com/help/matlab/ref/prod.html>