

multimedia-hw3

by 108062138 Po-Yu, Wu

Readme.md (<http://Readme.md>) contains only the *execution* part in report.pdf

report link (<https://hackmd.io/@sBeNJ4fqRNqa67PhyWWV4A/BJzT-Fwmn>)

1. (70%) Try the two search ranges ($p=8$ and $p=16$) for two macroblock sizes (8×8 and 16×16) by using the two search methods. The reference image is 40.jpg, and the target image is 42.jpg.

a. Show the predicted images by using the block matching with all the above combinations. (8 images)

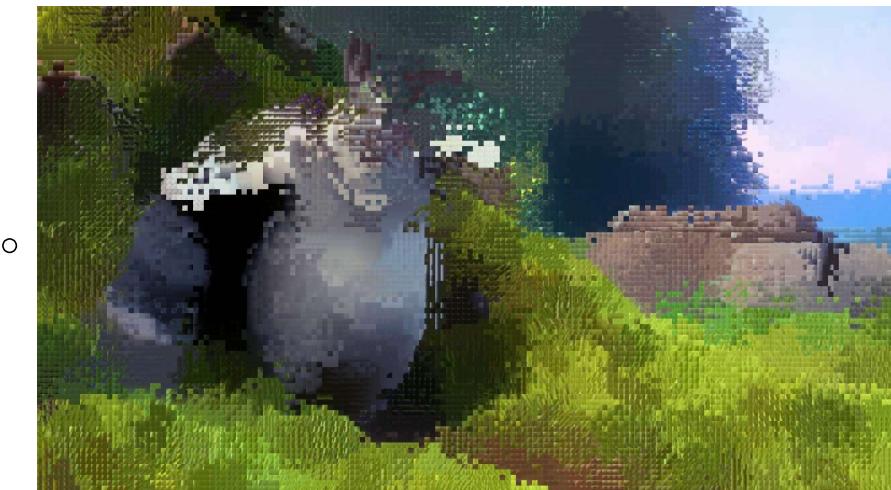
PREDICTED IMAGE: DISPLAY SEQUENCE IS {BLOCK SIZE=8, P=8}, {BLOCK SIZE=8, P=16}, {BLOCK SIZE=16, P=8}, {BLOCK SIZE=16, P=16}

- full search:





- 2d log:





b. Show the motion vectors images for all the above combinations. (8 images)

MOTION VECTOR: DISPLAY SEQUENCE IS {BLOCK SIZE=8, P=8}, {BLOCK SIZE=8, P=16}, {BLOCK SIZE=16, P=8}, {BLOCK SIZE=16, P=16}

- full search:





- 2d log:



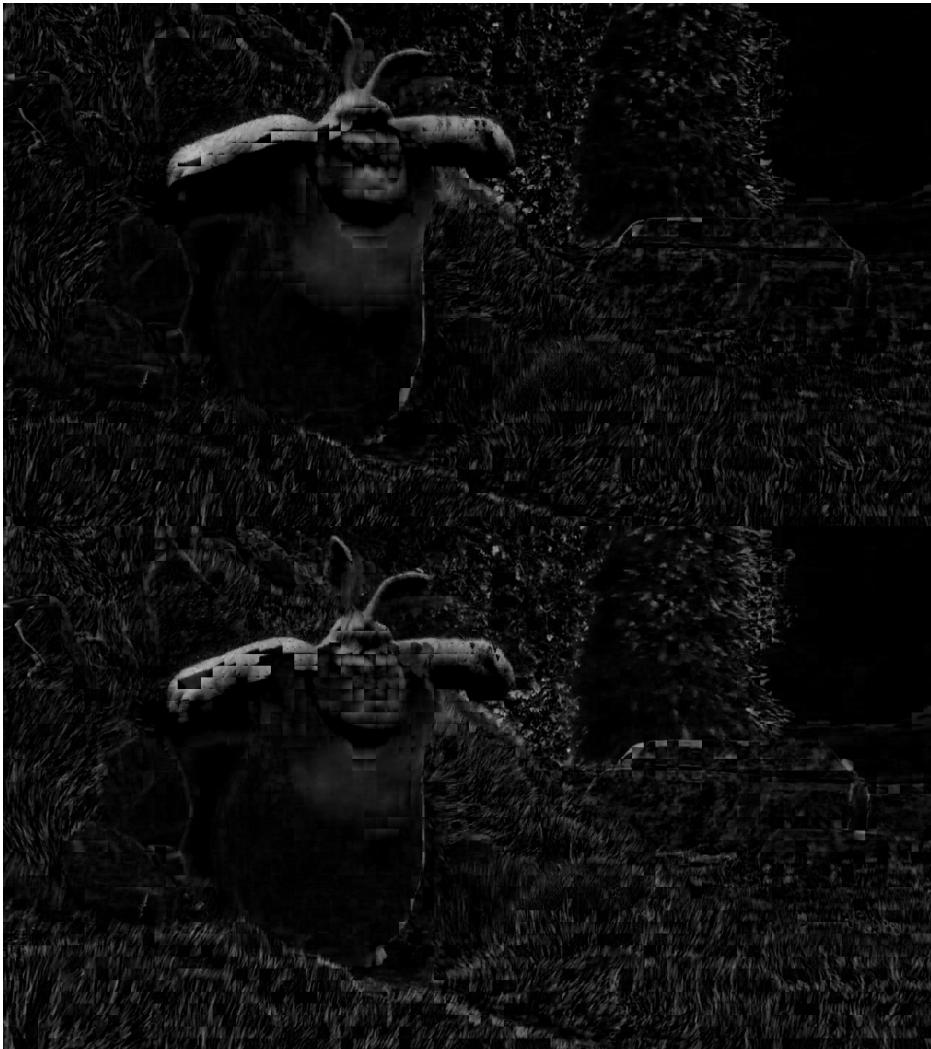


c. Show the residual images for all the above combinations. (8 images)

RESIDUAL IMAGE: DISPLAY SEQUENCE IS {BLOCK SIZE=8, P=8}, {BLOCK SIZE=8, P=16}, {BLOCK SIZE=16, P=8}, {BLOCK SIZE=16, P=16}

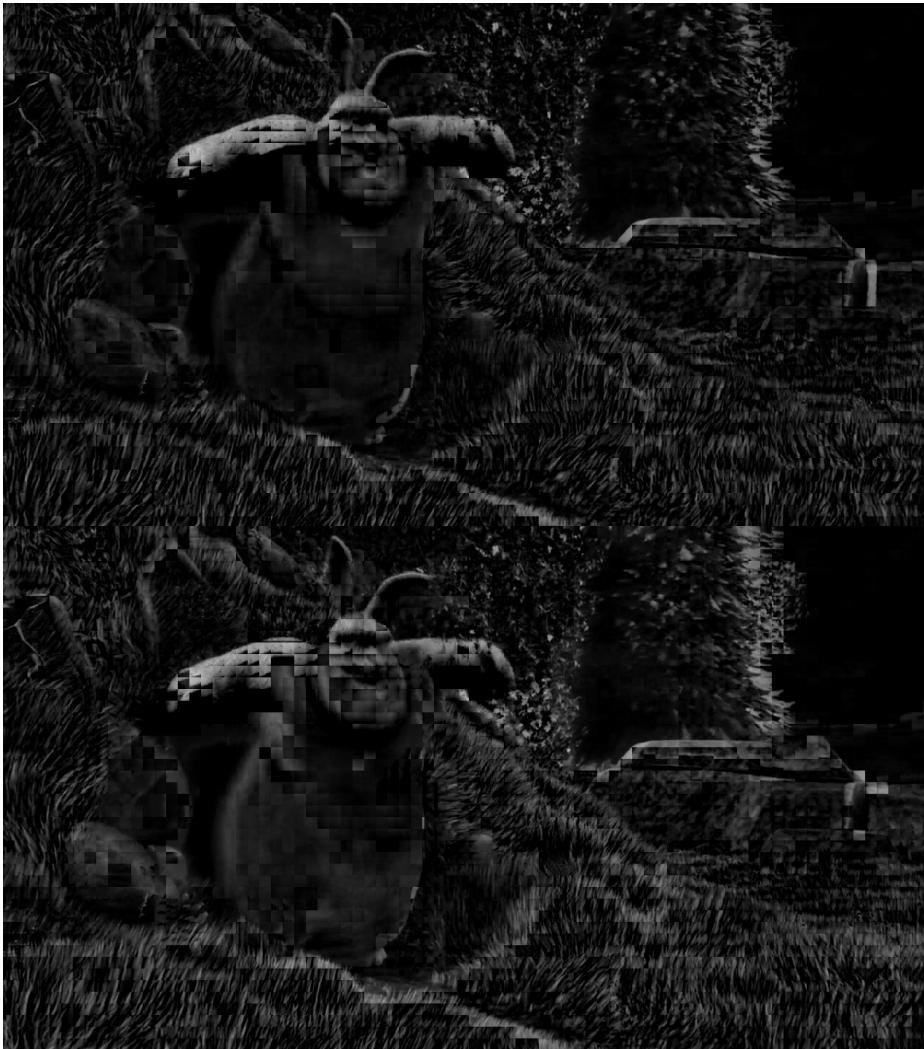
- full search:





- 2d log:



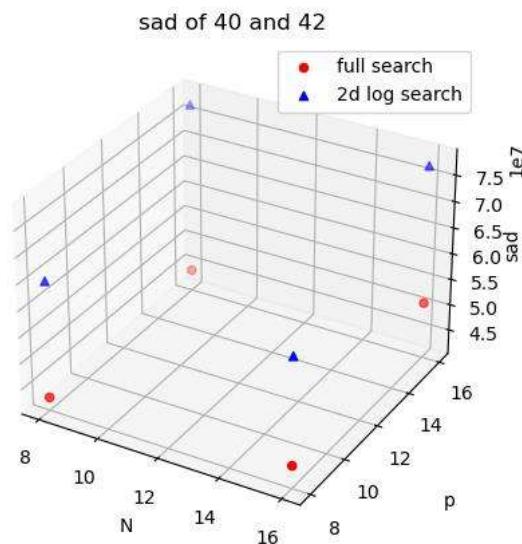
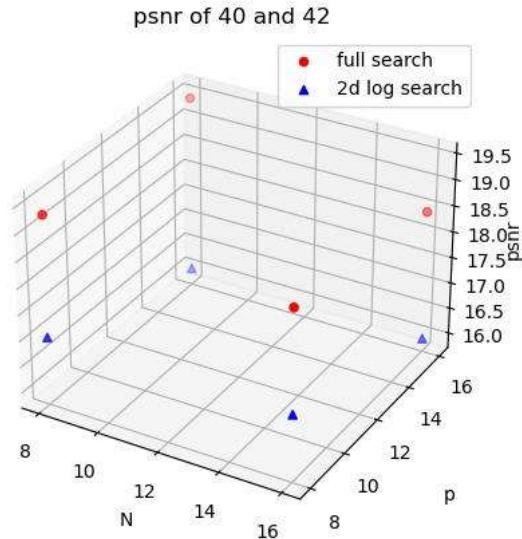


d. Compute the total SAD values and PSNR for all the results. Discuss the motion-based image prediction quality for all the above settings.

- SAD以及PSNR值將於下圖所示:

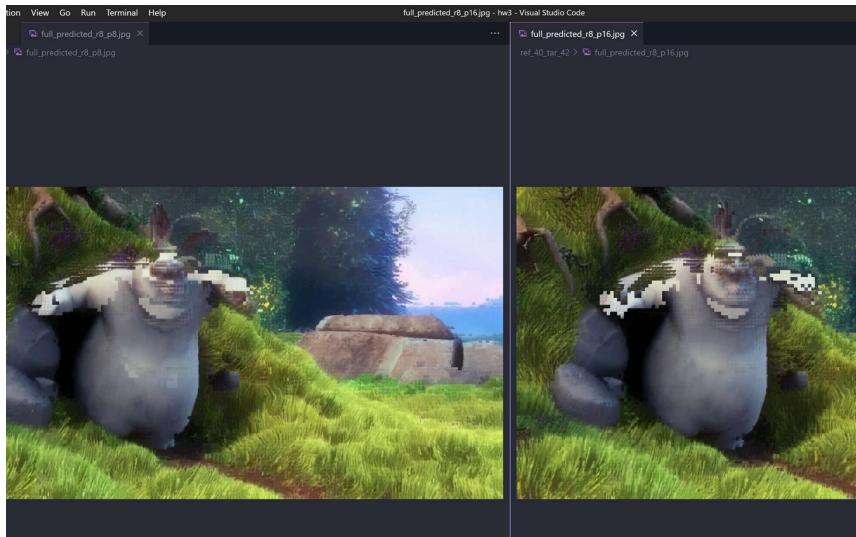
```
approach: full N: 8 p: 8 SAD: 42782373.0 PSNR: 19.325764646396262
approach: full N: 8 p: 16 SAD: 44348488.0 PSNR: 19.43705831450481
approach: full N: 16 p: 8 SAD: 44604168.0 PSNR: 18.967187538844943
approach: full N: 16 p: 16 SAD: 51056382.0 PSNR: 18.45081826241924
approach: 2d N: 8 p: 8 SAD: 65088101.0 PSNR: 17.015099391330942
approach: 2d N: 8 p: 16 SAD: 77253020.0 PSNR: 16.02899965940812
approach: 2d N: 16 p: 8 SAD: 65006158.0 PSNR: 16.999737214554024
approach: 2d N: 16 p: 16 SAD: 77371015.0 PSNR: 15.959778605185551
```

- 我將SAD(PSNR)值作為Z軸，p和macro block size(N)分別做為x,y軸，得出下圖。

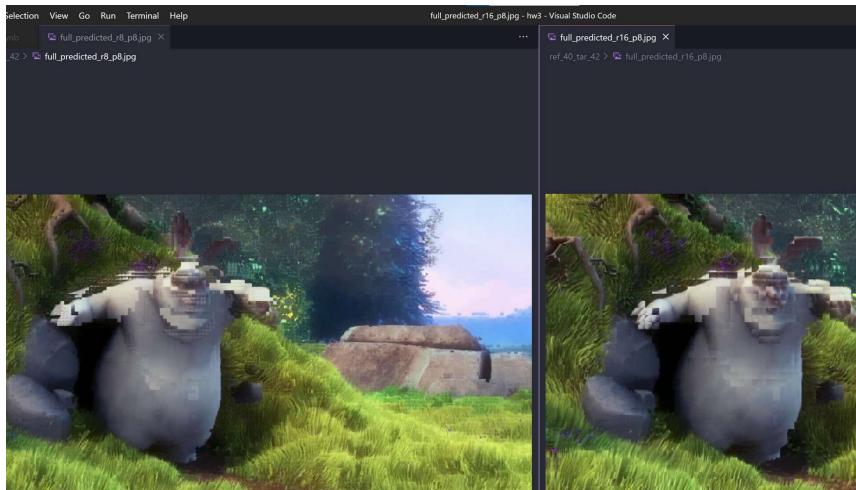


- 將p視為控制變因，其餘不動，可以觀察到p越大則PSNR值越大，SAD越小，但是p放大兩倍的效果sad和PSNR的數值均上升不到10%。在2d log search中，p相同者的PSNR以及SAD基本相同，看來在2d log search中，p的重要性相較macro block size而言大很多。此外，就圖片效果而言，p較大者在顏色的描繪上比較有連續的感覺，請見下圖兔子的肚肚，p較小

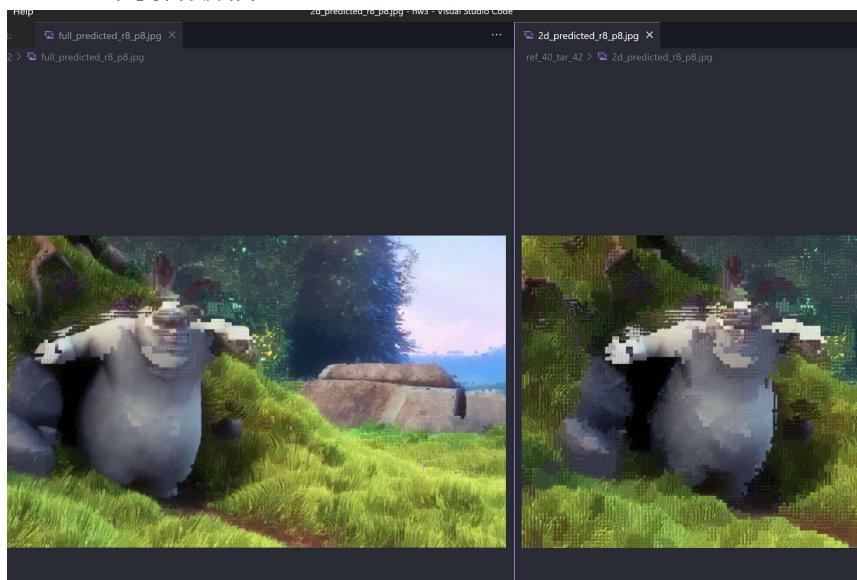
則會有一些奇怪得白色方塊，而放大p則可以發現方塊消失。



- 將macro block視為控制變因，其餘不動，可以看到N越大則PSNR值越小，SAD越大。可以發現在full search下，同一個macro block size的PSNR有相似度，但在SAD則未顯現此效果，看來在full search下，macro block size對PSNR值的影響力較p大，但其影響力卻不會擴散到SAD。就圖片效果而言，我發現macro block越大則方格感越重，而這是可以理解的，因為macro block都是不會overlap的，進而使macro block得error不能被其他東西遮掩，這會使macro block本身的error被保留。那在進一步思考，macro block的error來源是map回去reference block時找不著東西。那在考慮大的macro block和小的macro block誰比較容易找到相似的東西呢？很明顯是大的macro block比較難找，因為大macro block要考慮的東西比較多，故更難match(有點類似婚姻市場？QQ)



- 將approch做為控制變因，我發現full search所構成的平面在PSNR中均在2d log search之上，且SAD均在full serach之下。此舉很明顯表示在圖片效果上，full search比2d log search好得多。但是仔細看的話，可以發現其實2d log search輸的量很少(考量到計算輪是 $\log(O(\text{full search}))$)，在PSNR上，2d full search只輸了約3，而在SAD上，2d log search大概是full search的1.5倍。是故，這種程度的trade off是可以接受的。但是這只是數值討論，就圖片效果而言，左圖是full search而右圖是2d predicted search，full serch在畫面的連續效果很好，這個絲滑、連續的圖片感受是我願意花時間等待的，就算他的時間複雜度是2d log search的指數倍O $O(\log)$



2. (10%) Try the full search method with search range p=8 and macroblock sizes = 8x8. The reference image is 40.jpg, and the target image is 51.jpg. Show the PSNR of the result. Compare and discuss the PSNR with the result of same search range and macroblock in question 1.

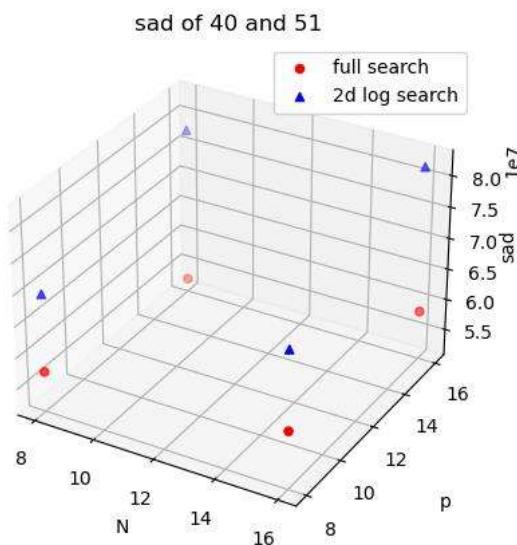
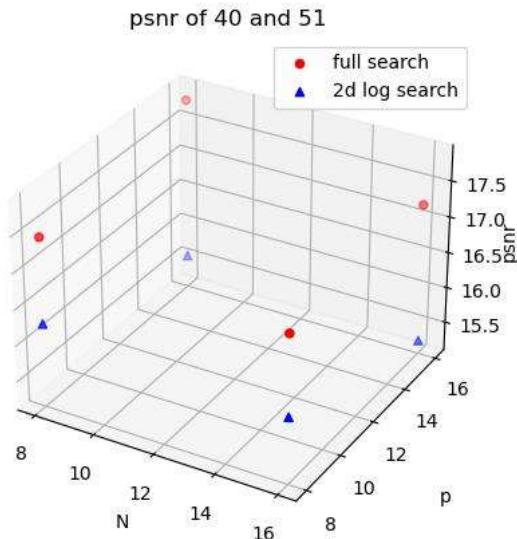
- SAD以及PSNR值將於下圖所示：

```

approach: full N: 8 p: 8 SAD: 57246561.0 PSNR: 17.452878227690338
approach: full N: 8 p: 16 SAD: 52896101.0 PSNR: 17.806378413786735
approach: full N: 16 p: 8 SAD: 60065389.0 PSNR: 17.151993179235653
approach: full N: 16 p: 16 SAD: 58604683.0 PSNR: 17.21593783831391
approach: 2d N: 8 p: 8 SAD: 69718257.0 PSNR: 16.26498435511055
approach: 2d N: 8 p: 16 SAD: 77856846.0 PSNR: 15.556119495070003
approach: 2d N: 16 p: 8 SAD: 72702512.0 PSNR: 16.030206359727867
approach: 2d N: 16 p: 16 SAD: 81905843.0 PSNR: 15.283551960175359

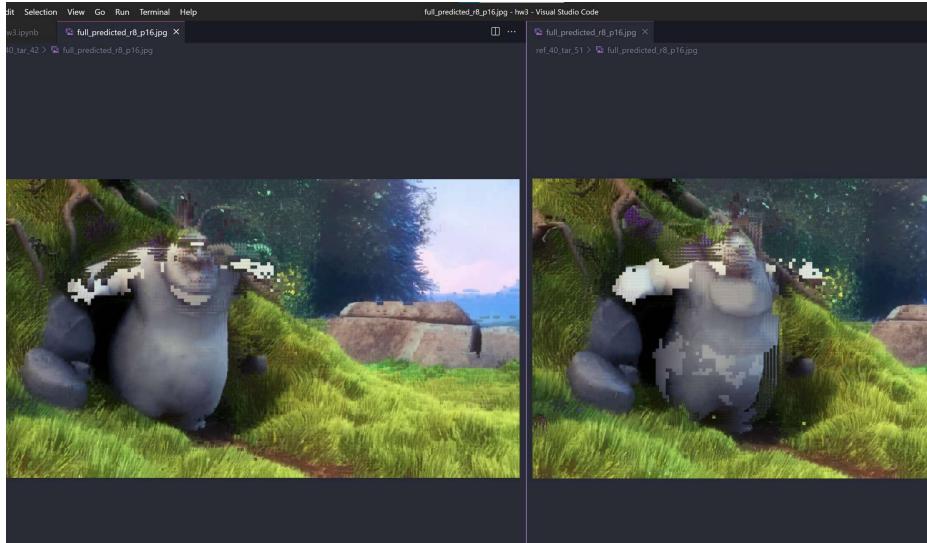
```

- 我將SAD(PSNR)值作為Z軸，p和macro block size(N)分別做為x,y軸，得出下圖。



- 我將1-d的圖和2的圖做對比，可以發現整體分布基本一致，但是PSNR值下降了3，但是在SAD數值卻差不多(改變約在10%以內)。我想這個原因可能是源自於圖片有一半都是草，進而使SAD的數值倍穩固了，但是在PSNR的部分，可能兔子的陰影或是亮面的改變太大，而新的圖片並沒有捕捉到該變成很亮的地方，進而使PSNR分數很差。
- reference image以及target的順序差異過大會造成整體圖片效果很跳，以下圖為例，{reference=40, target=51}兔子的頭頂都是方格，整體表現並不絲滑，而在{reference=40, target=42}則因為target和reference的順序只有相差1，是故頭頂沒有過量方格，進而使整體畫面

更加舒服。



3. (20%) Analyze the time complexity

a. Measure the execution time required for the two search algorithms with the two different search range sizes ($p=8$ and $p=16$).

- 這張圖有3個channel。一個channel可以被切割成 MN 個macro block。所以本圖片可以被切割成 $3MN$ 個macro block，其中每個macro block大小是 k^2 。
- 在full search的情況下，觀察的window大小是 $(2p + 1)^2$ ，每一個window會做 $(2p + 1 - k)^2$ 個SAD
 - 每一個SAD是 k^2 個加法。
 - 是故一個window則我會做 $k^2(2p + 1 - k)^2$ 。而在full search下，我每一個macro block都會找到一個相對應得window，所以總運算是 $3MNk^2(2p + 1 - k)^2$
 - 寫成bigO就是 $MNk^2(p - k)^2$
 - $p_1 = 8, p_2 = 16$ ，可以得到 $p_1 * 2 = p_2$ 。
 - 就 p_1 而言，計算花費是 $3MNk^2(2p_1 + 1 - k)^2$
 - 就 $p_2(2 * p_1 = p_2)$ 而言，計算花費是 $3MNk^2(2p_2 + 1 - k)^2 = 3MNk^2(4p_1 + 1 - k)^2$
- 在2d log search下，我的window大小仍然是 $(2p + 1)^2$ ，但是我不會對一整個window滑動，我按照2d log search得規則，每層算9個sad，然後選一個最好的然後進入下一層，直到我到最後面一層($p=1$)。是故對一個window size為 $(2p + 1)^2$ 得window而言，我們會做

$9 * (\lceil \ln(p) \rceil - 1) + 1$ 個SAD，又因為我們考量的是O，

$$9 * (\lceil \ln(p) \rceil - 1) + 1 \leq 9 * (\lceil \ln(p) \rceil)$$

- 故一個window頂多算 $9 * (\lceil \ln(p) \rceil)$ 次的SAD
- 每一個SAD是 k^2 個加法。
- 而在2d log的情況下，我每一個macro block都會找到一個相對應得window，故我會有

$$3MNk^2 * 9 * (\lceil \ln(p) \rceil) = 27MNk^2(\lceil \ln(p) \rceil)$$
- 寫成bigO就是 $MNk^2(\lceil \ln(p) \rceil)$
- $p_1 = 8, p_2 = 16$ ，可以得到 $p_1 * 2 = p_2$ 。
 - 就 p_1 而言，計算花費是 $27MNk^2(\lceil \ln(p_1) \rceil)$
 - 就 p_2 而言，計算花費如下，第一個小魚等魚是把ceil拿掉，第二個小魚等魚是把 $\ln 2$ 拿出來。

$$27MNk^2(\lceil \ln(p_2) \rceil) = 27MNk^2(\lceil \ln(p_1 * 2) \rceil) \leq 27MNk^2(\ln(p_1 * 2))$$

```

1   p   ->9
2   p/2->9
3   p/4->9
4   p/8->9
5   ...
6   1   ->1(9)

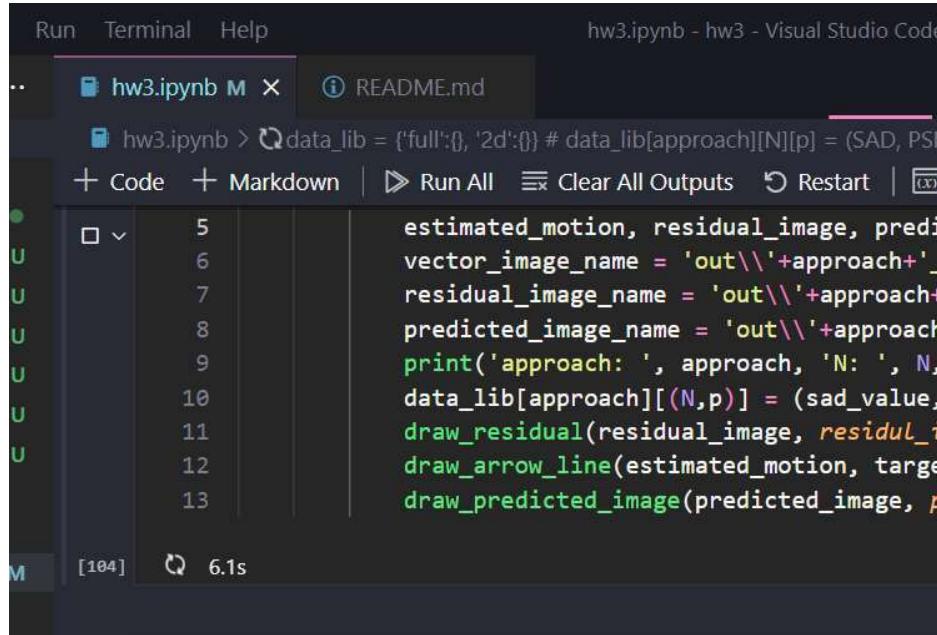
```

b. Compare and discuss the execution time with the theoretical time complexity for the two search algorithms.

- 就以full search而言，其big O複雜度是 $MNk^2(p - k)^2$ 。我發現full search的情況下，p提升兩倍，整體計算花費會多四倍，因為p是在平方裡面。
- 至於2d log search，其big O複雜度是 $MNk^2(\lceil \ln(p) \rceil)$ 。在p變為兩倍的情況下，計算只會增加 $MNk^2 * \ln(2)$ ，則可以視為constant increment。而這也是為什麼2d search可以很高速的處理完p很大的case。

execution

- 在 hw3.ipynb 按下 Run All，會在 .\out 輸出八張image



```
Run Terminal Help
hw3.ipynb - hw3 - Visual Studio Code
hw3.ipynb M X README.md
hw3.ipynb > data_lib = {'full':{}, '2d':{}} # data_lib[approach][N][p] = (SAD, PSM
+ Code + Markdown | ▶ Run All ⚡ Clear All Outputs ⚡ Restart | ☰
5
estimated_motion, residual_image, predi
vector_image_name = 'out\\'+approach+'_
residual_image_name = 'out\\'+approach+
predicted_image_name = 'out\\'+approach
print('approach: ', approach, 'N: ', N,
data_lib[approach][(N,p)] = (sad_value,
draw_residual(residual_image, residul_i
draw_arrow_line(estimated_motion, target_
draw_predicted_image(predicted_image, p
[104] 6.1s
```