楊宗翰
101062142
2015/6/1

Introduction to Multimedia Homework 4 Report

• Part (a)

In this part, we will implement 2 motion estimation algorithms which are used in video compression.

For full search algorithm, it's necessary to enumerate all macroblocks in the range and find minimum SAD among them, which is exhaustive. It's easy to implement, though. Using many for-loops will do the trick.

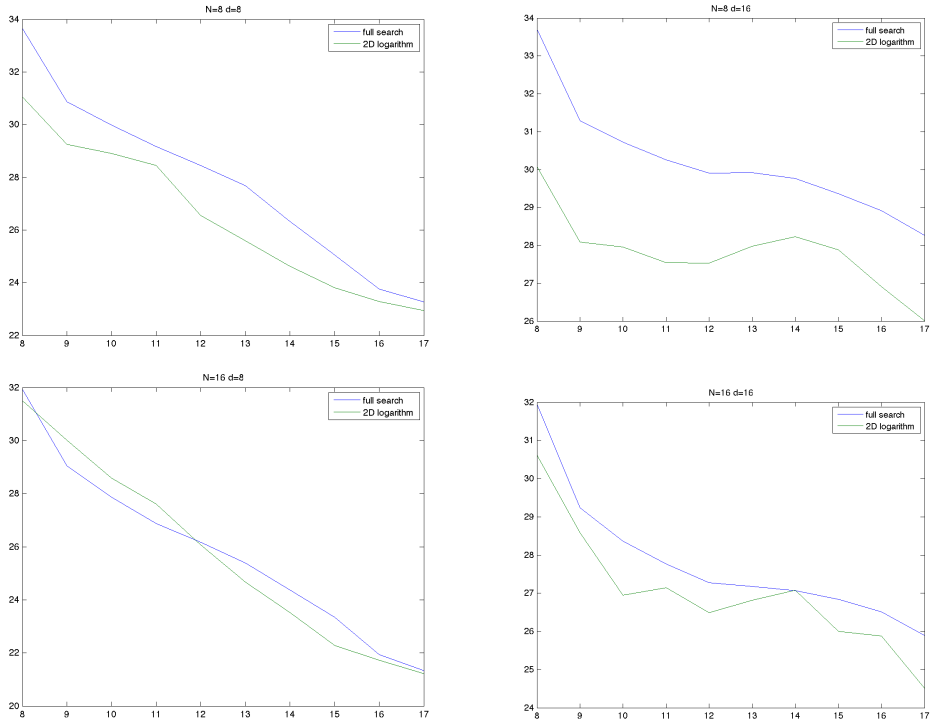
For 2D logarithmic algorithm, it's smarter and less computation is needed. By searching 5 macroblocks, we can determine which direction to go by finding which way has minimum SAD. We will decrease neighbor range until the range is 1. It's a little bit like best-first search for me.

```
Full N= 8 d= 8 frame=08 Total_SAD=2388.325 2Dlg N= 8 d= 8 frame=08 Total_SAD=2455.839
Full N= 8 d= 8 frame=17 Total_SAD=7462.212 2Dlg N= 8 d= 8 frame=17 Total_SAD=6862.608
Full N= 8 d=16 frame=08 Total_SAD=2373.376 2Dlg N= 8 d=16 frame=08 Total_SAD=2597.408
Full N= 8 d=16 frame=17 Total_SAD=4084.655 2Dlg N= 8 d=16 frame=17 Total_SAD=4341.749
Full N=16 d= 8 frame=08 Total_SAD=2708.067 2Dlg N=16 d= 8 frame=08 Total_SAD=2394.988
Full N=16 d= 8 frame=17 Total_SAD=9665.153 2Dlg N=16 d= 8 frame=17 Total_SAD=9059.231
Full N=16 d=16 frame=08 Total_SAD=2702.396 2Dlg N=16 d=16 frame=08 Total_SAD=2522.549
Full N=16 d=16 frame=17 Total_SAD=5107.133 2Dlg N=16 d=16 frame=17 Total_SAD=5280.863
```

From the output result, we can observe that with farther frame from frame7, the total SAD value is larger. Since frame8 is very near from frame7, the total SAD value does not change much with different parameter. For frame17, which is far from frame7, the total SAD is larger. And we notice that which larger search range, we can reduce total SAD value greatly. With larger macroblock size, the total SAD value rises slightly. Maybe, it's the difference inside the larger block makes it inevitable.

The residual images are in the /output folder. For frame8, the difference is not so obvious since they are pretty near from frame7. With higher search range, we get much less gray for frame17, which means less difference.

- Part (b)



From the output result, we can observe that with farther frame from frame7, we get less PSNR, meaning lower compression quality. In general, full search algorithm has higher PSNR value under the same condition.

- Part (c)

```
Target = frame17, Reference = frame7

Full Search  N= 8 d= 8 time= 6.186 Total_SAD=7462.212
2D logarithm N= 8 d= 8 time= 0.876 Total_SAD=6862.608
Full Search  N= 8 d=16 time=22.480 Total_SAD=4084.655
2D logarithm N= 8 d=16 time= 1.050 Total_SAD=4341.749
Full Search  N=16 d= 8 time= 1.580 Total_SAD=9665.153
2D logarithm N=16 d= 8 time= 0.240 Total_SAD=9059.231
Full Search  N=16 d=16 time= 5.873 Total_SAD=5107.133
2D logarithm N=16 d=16 time= 0.298 Total_SAD=5280.863
```

- Assumption:

  image size = m * m
  macroblock size = N
  search range = d
  mb = macroblock

- Full search algorithm

(1) Outer loop to enumerate mb: (m/N) * (m/N)
(2) For each mb, find other mb to compare in search range: (2*d+1)^2 ~= 4*d^2
(3) For each comparison, calculate SAD value: N * N

Multiply (1), (2) and (3) up, we get O(m^2 * d^2)
The result seems to irreverent to N. But the execution time suggests otherwise. Maybe it's some optimization in vector and matrix computation that causes the difference.

- 2D logarithmic algorithm

(1) Outer loop to enumerate mb: (m/N) * (m/N)
(2) Find minimum SAD in for each macroblock: O(lg(d))
(3) For each comparison, calculate SAD value: N * N

Multiply (1), (2) and (3) up, we get O(m^2 * lg(d))
We face the same problem as above again. Neglecting this factor, we can find the ratio for d = 16 and d = 8 is lg(16) / lg(8) ~= 1.050 / 0.876 (exec time) ~= 1.3