Adam Boyer

CS-6323-DF1

Assignment 3

Included in my deliverable are the following files:

1. assignment.py
2. Frame1.png
3. Frame2.png

To run the program please follow the following steps (on Windows machine)

1. Install python3 here: <https://www.python.org/downloads/>
   1. note : leave option to install pip checked during setup wizard
2. Place all provided files in the same directory
3. Open command prompt in directory with files
4. Run the following commands to install dependencies
   1. pip install numpy
   2. pip install scipy
   3. pip install opendv-python
5. Run this command to execute the program
   1. python assignment.py

Experiment Steps and Explanations:

* 1. The first step I took was to read the two images into the program and convert them to grayscale to reduce computation cost.
     1. *#Read in the two frames*
     2. ref\_img = cv2.imread("Frame1.png")
     3. target\_img = cv2.imread("Frame2.png")
     4. *#Convert the images to grayscale to reduce the computation cost*
     5. *#We are really only concerned with the change in intensity levels*
     6. ref\_gray\_scale = cv2.cvtColor(ref\_img, cv2.COLOR\_BGR2GRAY)
     7. target\_gray\_scale = cv2.cvtColor(target\_img, cv2.COLOR\_BGR2GRAY)
  2. For my first go through, I want to find the the most similar block in frame1 to the top left 16X16 of block two
     1. *Here is the block size 16x16*
     2. block\_size = 16
     3. *#Start at the top left corner in our target image*
     4. print("")
     5. print("Here is the top left block")
     6. print("")
     7. top\_left\_block = target\_gray\_scale[:block\_size, :block\_size]
     8. compute\_sad\_and\_dct(top\_left\_block, ref\_gray\_scale)
  3. Next, I compared the 16x16 top left block of frame2 to each 16x16 block of frame1. I searched from the top left of the reference frame and left to right, top to bottom. I set the minimum SAD value to the max allowed for floats in python, and if the new reference block’s SAD with the target block is lower, I update the best match block and record the motion vector.
     1. *#Here is the block size 16x16*
     2. block\_size = 16
     3. *#largest possible float value set to minimum SAD and motion vector set to zero*
     4. min\_SAD = sys.float\_info.max
     5. motion\_vector = (0,0)
     6. best\_match\_block = None
     7. best\_match\_x1 = None
     8. best\_match\_y1 = None
     9. best\_match\_x2 = None
     10. best\_match\_y2 = None
     11. *#Find the best match in the reference image*
     12. *for* i *in* range(block\_size, *ref\_gray\_scale*.shape[0] - block\_size, block\_size):
     13. *for* j *in* range(block\_size, *ref\_gray\_scale*.shape[1] - block\_size, block\_size):
     15. *#Get current ref block*
     16. x1 = j - block\_size // 2
     17. y1 = i - block\_size // 2
     18. x2 = j + block\_size // 2
     19. y2 = i + block\_size // 2
     20. ref\_block = *ref\_gray\_scale*[y1:y2, x1:x2]
     21. *#Calculate the Sum of Absolute Difference between the target and ref blocks*
     22. current\_SAD = np.sum(np.abs(ref\_block - *target\_block*))
     23. *#Update best match*
     24. *if* current\_SAD < min\_SAD:
     25. best\_match\_block = ref\_block
     26. best\_match\_x1 = x1
     27. best\_match\_y1 = y1
     28. best\_match\_x2 = x2
     29. best\_match\_y2 = y2
     30. min\_SAD = current\_SAD
     31. motion\_vector = (j - block\_size, i - block\_size)
  4. Next, using the motion vector found during motion estimation, I perform motion compensation on the reference block. I also add padding if after motion compensation I go out of bounds for the reference fame.
     1. *# Calculate the new position of the best matching block in the reference frame*
     2. x, y = motion\_vector
     3. new\_x1 = best\_match\_x1 + x
     4. new\_y1 = best\_match\_y1 + y
     5. new\_x2 = best\_match\_x2 + x
     6. new\_y2 = best\_match\_y2 + y
     7. *#print(x,y,new\_x1,new\_y1,new\_x2,new\_y2,x1,y1,x2,y2)*
     8. *#while the block is out of bounds for ref image add padding of zeros*
     9. *while* new\_y2 > *ref\_gray\_scale*.shape[0] or new\_x2 > *ref\_gray\_scale*.shape[1]:
     10. *# Add zero padding to the image*
     11. pad\_size = 50
     12. *ref\_gray\_scale* = cv2.copyMakeBorder(*ref\_gray\_scale*, pad\_size, pad\_size, pad\_size, pad\_size, cv2.BORDER\_CONSTANT, *value*=0)
     13. *# # Extract the new block from the reference frame*
     14. new\_ref\_block = *ref\_gray\_scale*[new\_y1:new\_y2, new\_x1:new\_x2]
  5. Now, with the new motion compensated ref block, I computed the difference block.
     1. *#Now find the difference block of the best match block and the target block*
     3. diff\_block = *target\_block* - new\_ref\_block
  6. Next, I computed the Discrete Cosine Transformation block of the difference block in both the horizontal and vertical direction.
     1. *#Compute the Discrete Cosine Transform of the diff\_block*
     2. dct\_block = dct(dct(diff\_block, *axis*=0, *norm*='ortho'), *axis*=1, *norm*='ortho')
  7. Finally, I rounded all the values of the dct\_block and converted the floats to integers
     1. *#Round values to the nearest integer*
     2. dct\_int = np.round(dct\_block)
     3. dct\_int = dct\_int.astype(int)
  8. All steps were repeated for the center block of the target frame2 and the bottom right block for frame2

Experiment Outcomes and Discussion:

First I will share my output for the program, then I will discuss my thoughts on these outputs.

Program output:

$ python assignment.py

Here is the top left block

Here DCT of the difference between the target block and the best match block after movement compensation

3272 -249 -141 -8 90 50 -78 -6 64 22 -97 47 -36 47 -23 28

379 279 167 8 -108 -59 106 15 -83 -31 121 -65 49 -58 33 -36

-537 -89 -59 10 65 36 -89 -26 60 27 -74 64 -41 34 -35 20

-141 -116 -58 -24 -1 -14 55 33 -33 -15 20 -52 27 -8 32 -4

32 231 116 22 -55 -2 -4 -23 5 -6 22 28 -9 -11 -17 -9

-562 -194 -98 -3 91 5 -55 1 16 34 -42 8 -12 17 -9 19

154 59 39 -8 -99 -12 93 23 -25 -52 47 -46 33 -13 31 -25

-436 72 10 3 82 24 -104 -40 29 55 -45 72 -48 9 -41 30

30 -122 -9 19 -47 -43 80 39 -31 -38 45 -78 54 -11 38 -35

-289 75 -26 -40 6 62 -38 -24 37 7 -53 66 -48 22 -23 33

9 32 65 38 29 -66 0 5 -46 25 60 -42 32 -35 8 -19

-222 -125 -66 0 -50 47 10 4 61 -43 -57 21 -11 42 -4 -5

-13 155 19 -67 55 -10 4 -3 -72 44 42 -8 -9 -41 13 34

-80 -123 44 122 -47 -28 -29 -6 72 -33 -22 3 22 31 -24 -55

-65 67 -81 -133 33 45 43 13 -59 18 6 -3 -24 -18 28 57

2 -24 62 87 -17 -34 -31 -10 34 -7 1 3 15 8 -19 -35

motion vector: (32, 32)

Here is just the DCT of the target block not the difference block

723 -4 -1 -1 1 1 2 0 0 0 0 0 1 -1 0 0

90 2 0 0 -1 0 -1 1 0 0 -1 1 -1 1 0 0

119 -1 -2 0 2 -1 0 0 0 0 1 0 1 0 1 0

170 2 2 0 -1 0 0 0 -1 0 0 0 -1 0 0 0

134 -1 1 0 0 0 0 0 0 0 -1 0 0 0 0 0

122 -1 1 2 0 -1 -1 0 1 0 0 0 0 1 0 0

108 -1 -2 0 2 -1 -1 0 0 0 -1 0 1 0 0 0

108 -2 0 1 0 0 1 0 0 0 1 0 0 0 0 0

97 2 0 -1 0 0 0 0 0 0 -1 0 0 0 0 0

91 -1 0 1 -1 0 0 1 -1 0 1 0 0 0 0 0

77 0 -1 -1 1 0 0 -1 0 0 0 0 0 0 0 0

65 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0

53 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0

42 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0

27 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

11 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0

Here is the center block

Here DCT of the difference between the target block and the best match block after movement compensation

702 32 20 -3 6 3 1 -1 -2 1 0 -1 4 -1 -1 2

-45 58 -5 -19 -13 -1 -7 3 -4 3 0 -2 2 -1 -2 0

1 -24 -32 -14 3 -5 1 2 1 1 0 -1 1 0 0 1

-21 -13 -11 3 6 -9 11 3 2 1 0 1 1 1 0 0

3 -1 -2 8 0 6 10 4 3 2 -1 0 0 0 -1 0

-1 4 8 3 9 6 5 -3 -1 -2 -1 0 -1 -1 0 0

-1 4 0 4 -1 4 0 -2 -3 -3 -3 -1 -1 0 0 0

2 0 3 1 3 1 0 -4 0 -1 0 0 0 0 0 0

-1 5 2 3 -2 3 -3 -1 1 -1 2 0 0 0 0 0

0 -1 0 1 1 0 -1 0 1 -1 1 0 -1 0 1 -1

-2 2 0 1 0 0 -2 0 1 1 0 1 0 0 0 -1

2 -1 0 0 0 0 -1 -1 1 -1 0 0 0 1 0 0

0 -1 -1 -1 -1 1 -1 0 1 0 1 1 0 0 0 0

0 -1 -2 -1 1 0 0 0 0 1 -1 0 0 0 0 1

-1 -1 -2 -1 0 0 0 1 0 0 0 1 0 0 0 0

1 0 0 -1 -1 1 0 -1 0 0 0 0 1 0 0 0

motion vector: (448, 336)

Here is just the DCT of the target block not the difference block

702 32 20 -3 6 3 1 -1 -2 1 0 -1 4 -1 -1 2

-45 58 -5 -19 -13 -1 -7 3 -4 3 0 -2 2 -1 -2 0

1 -24 -32 -14 3 -5 1 2 1 1 0 -1 1 0 0 1

-21 -13 -11 3 6 -9 11 3 2 1 0 1 1 1 0 0

3 -1 -2 8 0 6 10 4 3 2 -1 0 0 0 -1 0

-1 4 8 3 9 6 5 -3 -1 -2 -1 0 -1 -1 0 0

-1 4 0 4 -1 4 0 -2 -3 -3 -3 -1 -1 0 0 0

2 0 3 1 3 1 0 -4 0 -1 0 0 0 0 0 0

-1 5 2 3 -2 3 -3 -1 1 -1 2 0 0 0 0 0

0 -1 0 1 1 0 -1 0 1 -1 1 0 -1 0 1 -1

-2 2 0 1 0 0 -2 0 1 1 0 1 0 0 0 -1

2 -1 0 0 0 0 -1 -1 1 -1 0 0 0 1 0 0

0 -1 -1 -1 -1 1 -1 0 1 0 1 1 0 0 0 0

0 -1 -2 -1 1 0 0 0 0 1 -1 0 0 0 0 1

-1 -1 -2 -1 0 0 0 1 0 0 0 1 0 0 0 0

1 0 0 -1 -1 1 0 -1 0 0 0 0 1 0 0 0

Here is the bottom right block

Here DCT of the difference between the target block and the best match block after movement compensation

2010 246 150 -316 255 -267 78 73 -137 132 -236 178 -102 91 -129 67

319 426 -107 -100 50 -114 27 0 -53 17 -74 9 3 6 -46 13

-33 -285 218 -190 21 -26 41 -145 107 -131 116 -126 50 -34 34 -40

-997 -367 376 -205 -31 11 -35 -3 46 -91 -8 17 -11 8 -24 -4

146 81 30 51 -138 35 -49 55 7 -53 -43 42 14 -23 -5 -11

-189 -142 -10 42 -10 -45 78 -69 -9 37 -17 15 -39 6 45 -48

-88 -195 -104 167 -90 62 16 -32 -47 74 4 3 -56 22 53 -49

-322 -112 36 51 -7 1 -10 72 -33 14 5 22 -7 20 -22 21

100 21 -8 13 96 -55 18 74 -15 -13 69 -46 45 10 -46 54

7 -95 -119 63 116 6 24 -3 -41 80 78 -61 -10 36 -2 25

-112 -172 -50 57 82 46 33 -55 37 60 30 7 -23 0 49 -6

90 -13 42 7 53 37 15 -24 115 -29 -16 68 24 -54 42 8

-26 41 -13 -3 76 63 -33 14 50 21 -12 45 5 -15 6 35

10 -33 -101 18 92 92 -41 -16 -3 75 45 -28 -16 21 6 27

-98 -30 0 -28 65 52 -27 -11 30 23 10 8 -3 -1 5 21

38 112 34 5 -28 45 -46 47 41 -28 -51 69 16 -26 -6 19

motion vector: (352, 144)

Here is just the DCT of the target block not the difference block

1249 -223 191 -123 41 51 -124 198 -251 278 -283 268 -235 187 -131 67

-468 -105 92 -78 62 -42 28 -12 -2 13 -18 19 -19 17 -11 6

433 92 -91 78 -61 44 -28 11 2 -8 16 -18 17 -15 11 -5

-314 -59 60 -51 40 -25 16 -4 -5 11 -14 15 -15 12 -9 5

292 61 -56 45 -38 26 -16 5 3 -8 13 -14 13 -10 8 -4

-237 -47 44 -40 30 -20 11 -3 -3 8 -10 11 -11 9 -6 4

176 39 -34 26 -22 16 -9 2 0 -6 6 -8 7 -7 5 -2

-110 -23 21 -19 13 -9 4 -3 -2 3 -3 5 -5 4 -3 1

55 11 -9 8 -8 5 -2 2 1 -3 2 -2 3 -2 1 -2

-3 -1 0 0 -1 -1 0 -1 0 0 -1 0 0 0 0 0

-35 -8 7 -5 5 -3 2 -1 1 1 -2 2 -1 1 -1 0

58 12 -12 8 -8 5 -3 1 1 -2 3 -2 2 -2 1 -1

-67 -14 12 -12 9 -6 3 -2 0 2 -3 3 -3 3 -2 1

64 14 -12 11 -7 7 -3 1 1 -1 2 -3 3 -3 2 -1

-49 -11 9 -10 6 -5 2 -2 -1 1 -2 2 -2 2 -1 1

27 6 -5 5 -4 2 -1 0 0 -1 1 -1 2 0 0 0

My thoughts:

If you look at the DCT outputs for the diff block and the target frame, it looks as though the diff block has higher values. This is true, but the values of the diff blocks are still relatively small throughout the matrix. If we were to add a quantization step to the program, most of the values in the bottom right section of the diff block would go to zero. The result of the image estimation/compensation and Discrete Cosine Transform allows us to perform intra-frame compression. We would need to only store a reference frame, the new diff blocks which hold far less information, and our motion vectors.