Advanced Digital Image Process

HW#5

作業#5

姓名：張耀中

學號：109368008

指導教授：高立人教授

### Development environment：

OS：unbuntu18.04

Editing tools：VScode

complilation tools：CMake

opencv version：3.2.0

2.2D-DFT

1. Write your own DFT subroutine (with origin shift) and test on rect256.raw, rotate256.raw, circle256.raw, square256.raw. Show the output of both magnitude and phase spectra. Discuss the difference of each result image.

rectangle

|  |
| --- |
| origin |
|  |

|  |  |
| --- | --- |
| real | image |
|  |  |
| magnitude | phase |
|  |  |

After 2D-DFT we’ll have two image real part and image part we can use get the magnitude by doing .As we can see in real part and image part there is something missing but is was complementary so after magnituding we get a complete image.To visualize the image I do the absolute value and normalize and powerlow 𝞬=0.4.

In spatial domain a vertical rectangle transfer to frequency domain becomes a horizontal sinc function.

And we can get the image phase by doing arctan() the values will between Ⲡ to -Ⲡ.

I shift the value by adding Ⲡ to make value in range 0~2Ⲡ and zoom in to 0~255 to show it.

In the corner of the DFT Image part is a little distortion so the phase at the corner is weird.

|  |
| --- |
| square origin |
|  |

|  |  |
| --- | --- |
| real | image |
|  |  |
| magnitude | phase |
|  |  |

As we can see a lot of features are the same as rectangles because they are all quadrangular and each corner is right angle but square is more symmetry so in frequency domain it is more symmetry too. The phase of DFT is more smooth too.

|  |
| --- |
| rotate origin |
|  |

|  |  |
| --- | --- |
| real | image |
|  |  |
| magtitude | phase |
|  |  |

As we can see. square diamond have a contour that is 45 degree , so In the frequency domain the feature is 45 degree too.

|  |
| --- |
| circle origin |
|  |

|  |  |
| --- | --- |
| real | imag |
|  |  |
| magnitude | phase |
|  |  |

As we can see, because a circle is symmetry from any degree. So to frequency domain, it is symmetry from any degree too. but the phase is messy.

question:

There is some missing part in all DFT images except the circle. Why?

1. Compare the output from (a) with output by using OpenCV DFT function. Discuss difference in execution time result images. Explain each other.

opencv

|  |  |
| --- | --- |
| rect | circle |
|  |  |
| square | rota |
|  |  |

In the resulting image, except the contrast it is the same as my DFT.

Execution Time

|  |  |  |
| --- | --- | --- |
|  | my | opencv |
| rectangle | 2923ms | 3798us |
| square | 2920ms | 2249us |
| rotate | 2911ms | 2109us |
| circle | 2873ms | 2111us |

Doing 2D-DFT function can using a 4 layers for loop to scale the x,y,u,v but its Execution time is about 6 mins, so we can use 1D-DFT to transfer f(x,y) to F(x,v) then do 1D-DFT again to get F(u,v), and a 1D-DFT only need 3 layers for loop. The time complexity of 4 layers for loop is n^4 and 1D-DFT is n^3 but we need to do it twice so the time complexity of 1D+1D DFT is 2n^3 and the execution time is only 3 second. but it is still 1000 times opencv's DFT. (The Execution time result image is at the bellow.

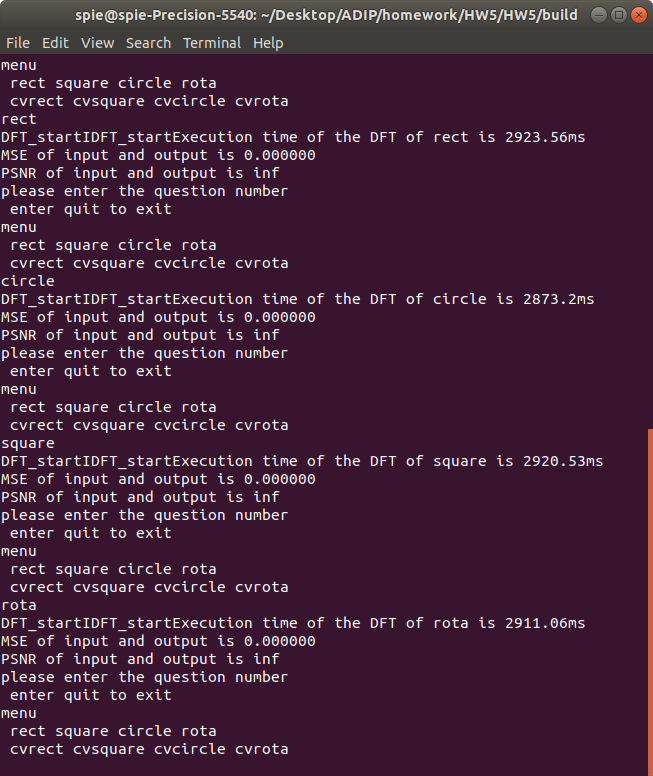
1. Write your own IDFT subroutine and test on DFT output from (a). Show the results of these images and corresponding with MSE and PSNR values.

|  |  |
| --- | --- |
| rota | circle |
|  |  |
| square | rect |
|  |  |

4-for-loop 2D-DFT Execution time and MSE PSNR

|  |  |
| --- | --- |
| square | result |
|  | MSE=0  PSNR=INF |
| rect |  |
|  | MSE=0  PSNR=INF |
| rota |  |
|  | MSE=0  PSNR=INF |
| circle |  |
|  | MSE=0  PSNR=INF |

3-for-loop 1D +1D DFT Execution time



To the result we can perfectly reduce the image by doing IDFT to the DFT image.

Result Image : [LINK](https://drive.google.com/drive/folders/1G74vD5xwUIov9wIThH1OBf_aCn01g1kH?usp=sharing)