EE5470 Computer Vision Lab 3
Yuzhuo Sun NetId: ys684

Binary Image Processing

Lab 4 Questions

Frequency Domain Filtering

1, In the above steps how did we manage to visualize the magnitude data?

In step 2, when command: vfix -float ted | vpix bf=-110.366 | vfft of=ted.xfft is executed, we subtract the DC component by 100.366. Then all the high frequency domain magnitude is easier to spot in stead of a single bright point in the center of the image. Then after command: ./vexfft -m -l ted.xfft of=ted.xlmag is executed, it is processed by a log function and into a visionX readable file.

2, How was the value for the bf= parameter selected?

The bf value is selected by the ted image's mean pixel value which is the DC component that we want subtract from the image.

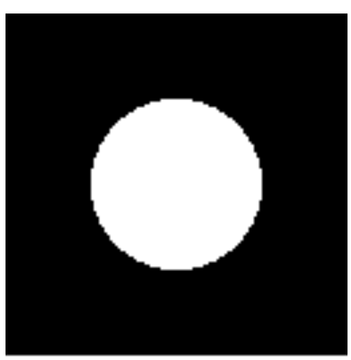
3, What does the first line accomplish?

The first line generates a circle with radius 32 on a 128x128 pixel image. And then vfix command changes the image type into float.

4, Why is vchan being used?

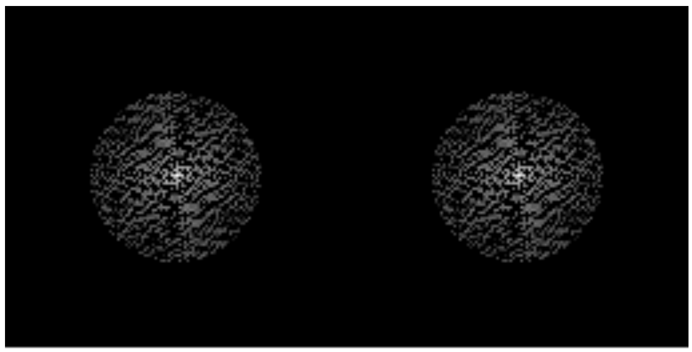
Command vchan is used to manipulate multichannel, this command (vchan if=f1 ig=f1 of=fil) in specific will chain image f1 with itself into a 2-channel image. And to use vexfft script, we need to have a 2-channel image.

5, Display the resulting image and comment on its filter characteristics.



f1 image

fil image fil phase

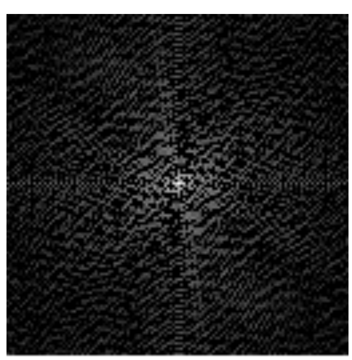


tedf.fft tedf.fft phase

(1), f1 image: vgenim x=128 y=128 c=32,32 hi=1 | vfix -float of=f1 is executed to generate f1, so f1 is generated on a 128x128 pixel image with a circle shape whose radius is 32.

(2), fil image: vchan if=f1 ig=f1 of=fil is executed to generate fil image, it does nothing to the original image but changes it into a 2-channel image for later process.

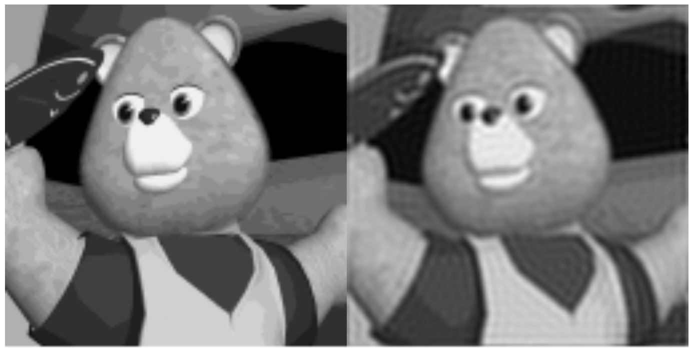
(3), tedf image: For this image, we need to compute the ted.fft first which is the FT transform of the original ted image by executing: vfix -float if=ted | vfft of=ted.fft, The magnitude of ted.fft is shown as follow, I did DC subtraction by: vfix -float ted | vpix bf=-110.366 | vfft of=ted.xfft, then display the log magnitude ./vexfft -m -l ted.xfft of=ted.xlmag



ted.fft image

Then we multiply the FT of ted image by low pass filter fil by executing: vop -mul if=ted.fft ig=fil of=tedf.fft, and we get rid of the high frequency component, then we get the answer.

6, Discuss the differences between the original and the filtered image.



Original image   vfft filtered

After filter on frequency domain, we got rid of the high frequency component, and the image is blurred.

7, Repeat the above filtering procedure with the image sshtl.

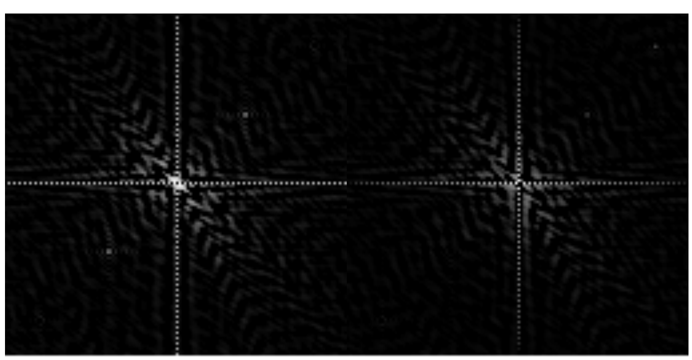
After executing:

vfix -float if= sshtl | vfft of= sshtl.fft

./vexfft sshtl.fft -m of= sshtl.mag

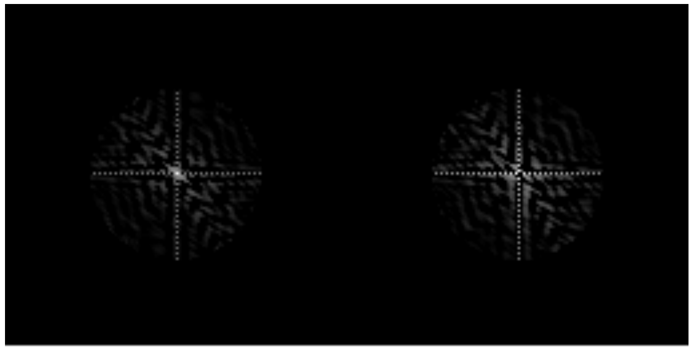
./vexfft sshtl.fft -p of= sshtl.phase

We got FT of sshtl image, but due to DC component as for ted image, we only have a dot on magnitude and phase. So I subtract the mean pixel value by executing: vfix -float ted | vpix bf=-128.987 | vfft of=sshtl.xfft then excute ./vexfft -m -l sshutl.xfft of=sshtl.xlmag | ./vexfft -p -l sshutl.xfft of=sshtl.xlphase to get the magnitude and phase of the original image.



sshtl magnitude   sshtl phase

Then execute vop -mul if=sshtl.fft ig=fil of=sshtlf.fft to multiply sshtl image and filter on frequency domain. Then we get the magnitude and phase by executing: ./vexfft -m -l sshtlf.xfft of= sshtlf.xlmag | ./vexfft -p -l sshtlf.xfft of= sshtlf.xlphase. The figures are shown as follow:



sshtlf magnitude   sshtlf phase

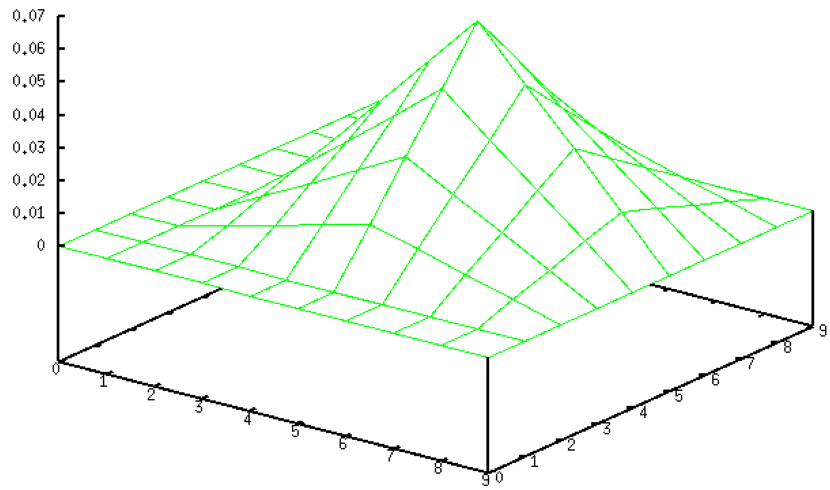
Then execute: vfft -i if=sshtlf.fft of=sshtl.f to get inverse of Fourier Transform, then execute: ./vexfft -m -l sshtl.f of=sshtlf.xlmag

original image   sshtl.f image

The image's high frequency component is filtered too. As shown in the image, the gradient change in the background become fuzzy and less smooth.

Spatial Domain Filtering

1, What does the spatial filter look like?



It looks like a pyramid in 3d plot.

2, How was it created?

It is generated by the following commands:

vgenim r=2,2 x=10 y=10 hi=1 of=sf1

vconv if=sf1 k=sf1 | vfix -float of=sf2

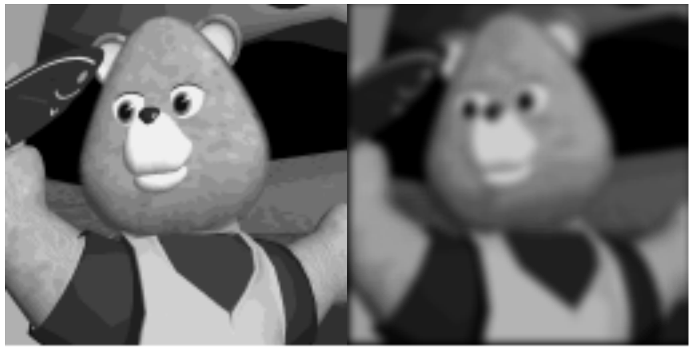
vpix if=sf2 tf=0.0039 of=sfil

The first vgenim will generate a rectangle with length 4 on a 10x10 pixel image. Then it is convolved with itself by vconv command, and when a rectangle convolve with itself, it will become a triangle. Then all pixel is multiplied by 0.0039. This filter is generated afterwards.

3, How was the above value of the tf= parameter selected?

tf parameter is selected by: 1/(100 * mean pixel value).

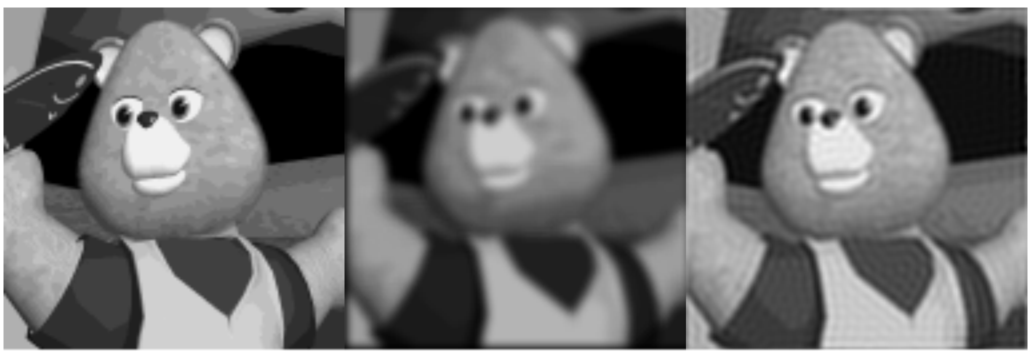
4, Discuss the differences between the original and the filtered image.



Original image sfil image

The original image and spatial domain filtered image are shown above. The filter I generated above serves as a low-pass filter. It will smooth the edges and blur the details of it. So, the filtered image looks a bit blurred compared with the original image.

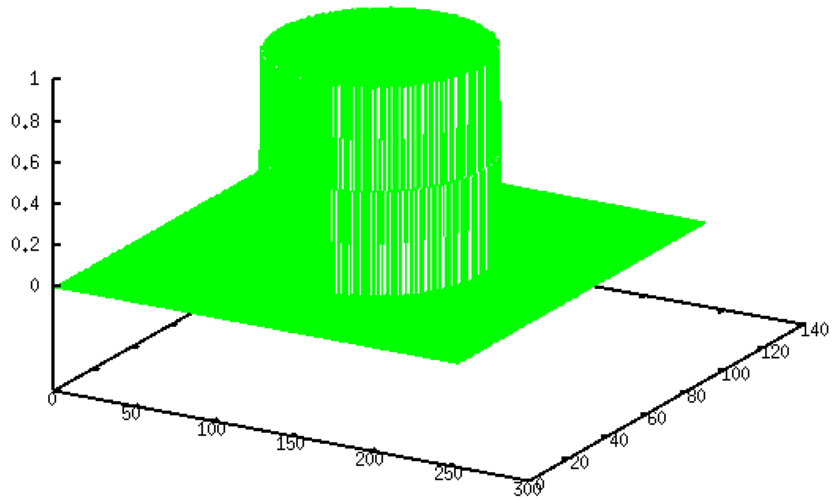
5, Also compare ted.sf with ted.f and discuss the differences.



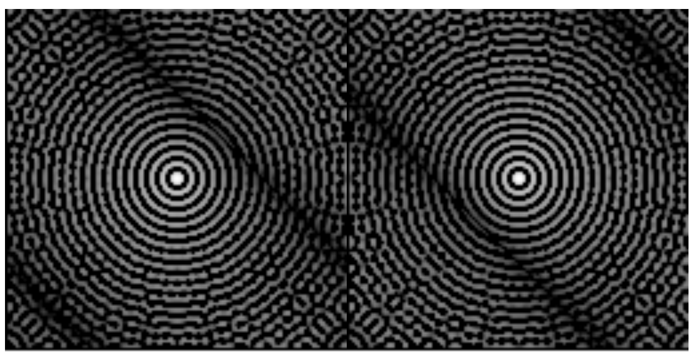
Original image sfil image vfft filtered

Frequency domain turned to be less blurred, and spatial domain is more blurred also smoother as well. It is because the image filter we used in ted.f is a cylinder in spatial domain while the ted.sf use a pyramid filter in spatial domain. Due to the cylinder's sharp transition in the boarder, while pyramid is more smooth. So ted.f is less blurred while ted.sf is more blurred and smoother as well

6, What does the image fil look like in the spatial domain? Can you make a good 2D graph of it?



In spatial domain, both magnitude and phase will be displayed together. In 2D, we need to execute: vfft fil of=fil.f, then excute: ./vexfft –m –l fil of=fil.mag. After the process, however, the image is too dark, so it is hard to see the 2D graph, then I used vpix if=fil.f fl tf=3000 of=fil3000.f to increase the mean pixel value to make 2D graph more visible, then execute the first and second command again to fil3000.f. And for phase, it is identical to above procedures.



2D magnitude 2D phase