

Lab 4

IMAGE FILTERING

Xinyi(Blanca) Bai

Cornell University Electrical and Computer
Engineering



Table of Contents

2. Frequency Domain Filtering	2
2.1 Display the original image using vview.....	2
2.2 in the man page for vfft, it says that the output file is a 2 channel file.What are the two channels?	3
2.3 In vview select Refresh and then review the images ted.mag.....	3
2.4 At this point I only find very little information in the magnitude image. Let's try a creative solution that will allow us to visualize the magnitude data :.....	4
1)observe the original image statistics. (in vview <i>Tools</i> → <i>Image Statistics</i>))	4
2)execute the following :.....	5
3)display the log magnituded of the FFT image	5
2.5 In the above steps how did we manage to visualize the magnitude data ?	6
2.6 How was the value for the bf= parameter selected.....	6
2.7 What does the first line accomplish ?	7
2.8 Why is vchan being used?	7
2.9 Display the resulting image and comment on its filter characteristics.....	7
2.10 to view the ted.f, use vexfft to display the log magnitude of it.	8
2.11 Discuss the differences between the <i>original</i> and the <i>filtered</i> image.	9
2.12 Repeat the above filtering procedure with the image sshtl.....	11
3. Spatial Domain Filtering.....	13
3.1 First generate the spatial filter :	13
3.2 What does the spatial filter look like?	14
3.3 How was it created?	15
3.4 How was the above value of the tf= parameter selected ?	15
3.5 Discuss the differences between the <i>original</i> and the <i>filtered</i> image.	16
3.6 Also compare <i>ted.sf</i> with <i>ted.f</i> and discuss the differences.....	16
3.7 What does the image <i>fil</i> look like in the spatial domain? Can you make a good 2D graph of it?.....	18
Appendix:.....	19

2. Frequency Domain Filtering

2.1 Display the original image using vview.



Figure1: vview ted

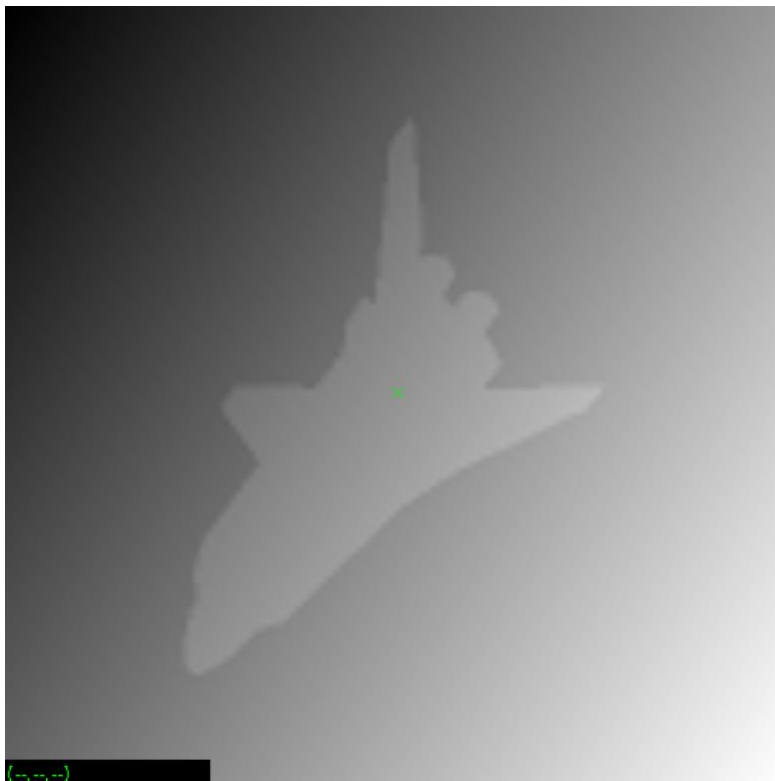


Figure2: vview sshtl

```
1. vfix - float if = ted | vfft of = ted.fft
```

2.2 in the man page for vfft, it says that the output file is a 2 channel file. What are the two channels?

It is said that “Vfft computes the FFT of the image. All computations are conducted in single precision floating point arithmetic and the result is a two channel (real, imaginary) image.”
So the two channels are Real and Imaginary channel.

since ted.fft is a 2 channel float image, we cannot display the file directly. In this case we can use the vexfft to extract viewable display images. From command menu select a method for viewing the data. Display the magnitude and the phase of the image ted.fft using the following procedure:

```
1. vexfft ted.fft - m of = ted.mag  
2. vexfft ted.fft - p of = ted.phase
```

2.3 In vview select Refresh and then review the images ted.mag

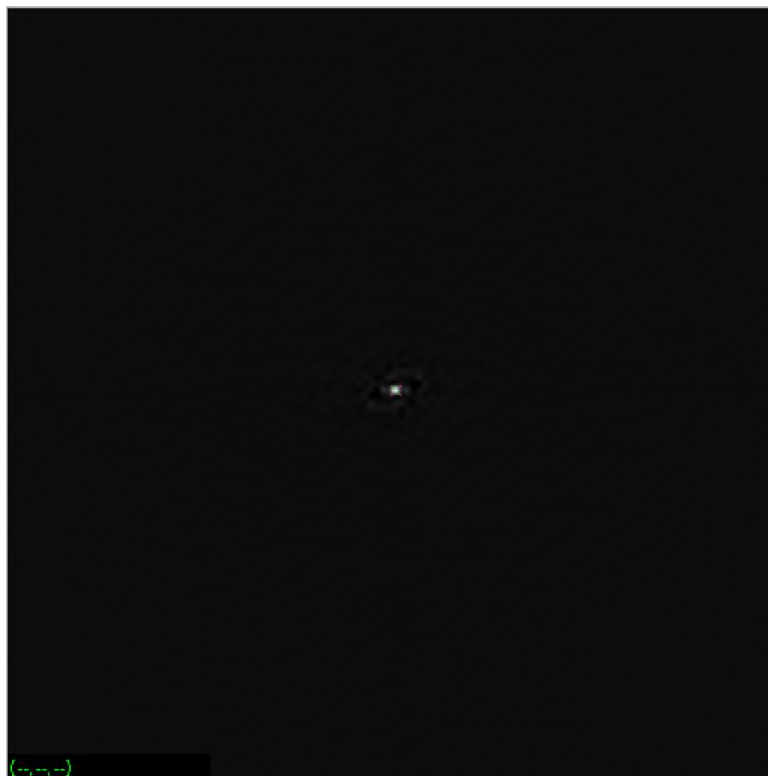


Figure3: vview ted.mag

2.4 At this point I only find very little information in the magnitude image. Let's try a creative solution that will allow us to visualize the magnitude data :

1)observe the original image statistics. (in vview *Tools*→*Image Statistics*))

```
Image Statistics for ted.phase

Global Statistics Summary

1 images in file      128 x 128 (0,0)

16384      (16383      ) number of pixels
255        (255        ) maximum pixel value
0          (29         ) minimum pixel value
255        (226        ) range
127        (127.008    ) mean pixel value
7.70129    (6.71733    ) variance
2.77512    (2.59178    ) standard deviation
9.75138    (19.1325    ) skewness
60.7345    (-272.215   ) kurtosis
```

Figure4: Statistics for ted.phase

```
Image Statistics for ted.mag

Global Statistics Summary

1 images in file      128 x 128 (0,0)

16384      (16378      ) number of pixels
255        (255        ) maximum pixel value
0          (4          ) minimum pixel value
255        (251        ) range
12.8142    (12.8189    ) mean pixel value
4.36599    (4.30743    ) variance
2.0895     (2.07543    ) standard deviation
96.2624    (98.3478    ) skewness
11041      (11346      ) kurtosis
```

Figure5: Statistics for ted.mag

2)execute the following :

```
1. vfix - float ted | vpix bf = -110.366 | vfft of = ted.xfft
```

See Appendix:

Another Figure: Statistics for ted.xfft

Another Figure: frequency histogram of ted.xfft

3)display the log magnitued of the FFT image

```
1. vexfft - m - l ted.xfft of = ted.xlmag
```

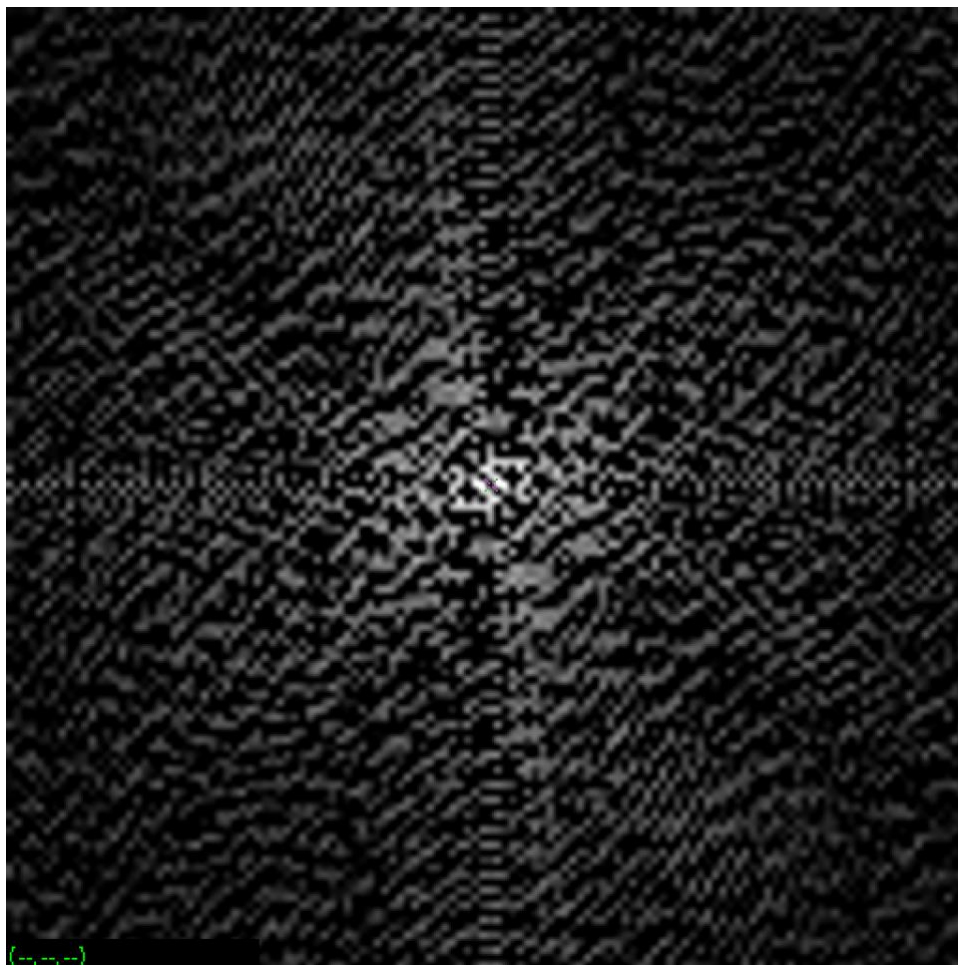


Figure6: vview ted.xlmag

See Appendix:

Another Figure: Statistics for ted.xlmag

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

We find if we use `vview` directly watch the `ted.meg` image, only very little information can be seen, and I also tried to see the Statistics of the image, and even `vplot` them.

Then according to the instruction this section, we find, first we subtract every pixel to 110.366, then select floating point (32-bit) output data type to output a new image `ted.xfft`. Next, we get the log of `ted.xfft` and output it as `ted.xlimg`. So, this time, we can see more information.

2.6 How was the value for the `bf=` parameter selected

Because the mean `ppixel` value of `ted` image is 110.366, To subtract every pixel with 110 on the one hand, it is a kind of compression method, it reduces the computing time in further operations, on the other hand, remains the relatively most information in the image, in other words, reduce distort.

Make a frequency domain filter and apply it to the image:

1. `vgenim x = 128 y = 128 c = 32, 32 hi = 1 | vfix - float of = f1`

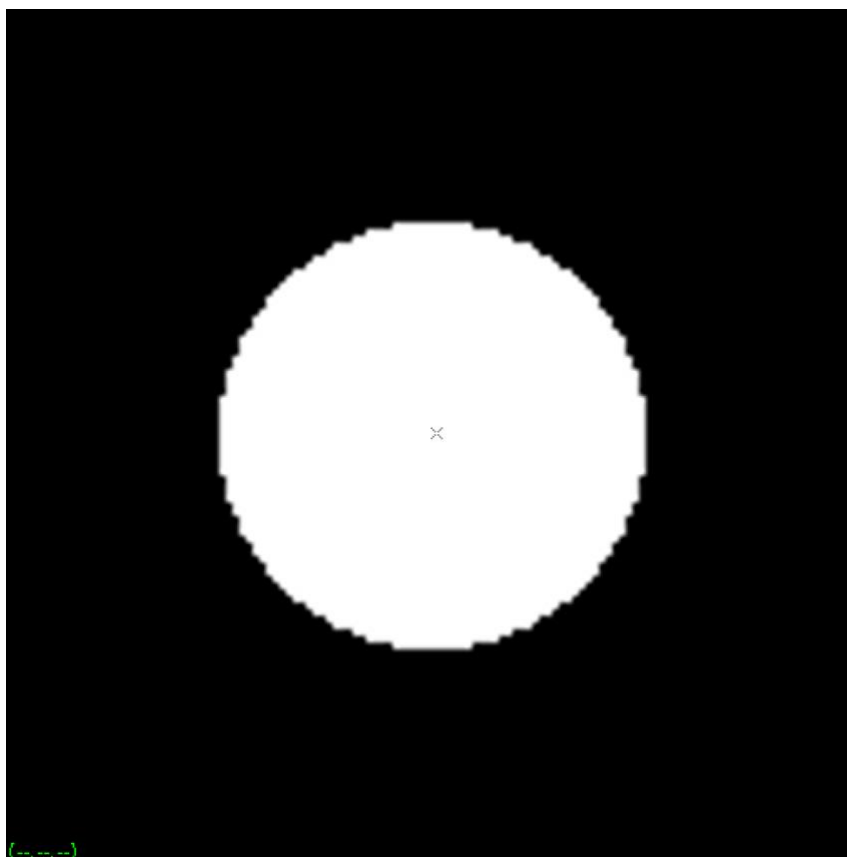


Figure7: vview f1

```
1. vchan if = f1 ig = f1 of = fil
```

See Appendix:

Another Figure: Statistics for fil

Another Figure: Statistics for tedf.fft

Another Figure: vplot ted.fft

```
1. vop - mul if = ted.fft ig = fil of = tedf.fft
use -mul to multiply ted.fft with fil.
```

2.7 What does the first line accomplish ?

Generate a single channel 128X128 byte image, it has a circle at (32,32), the high pixel value in this generated pattern is 1. Finally output the image as a float point type image names f1.

2.8 Why is vchan being used?

Vchan is used to manipulate multi-channel images, this is what frequency domain filter requires, it merge the image f1 with the image f1 and the output image names fil.

2.9 Display the resulting image and comment on its filter characteristics.

Because floating point type image cannot be open using vview, so we need to use vexfft to display the log magnitud of the FFT image fil, and ted.fft.

```
1. vexfft - m -l f1 of = f1.xlimg
```

See Appendix:

Another Figure: vview f1.xlimg

```
1. vexfft - m - l fil of = fil.xlimg
```

See Appendix:

Another Figure: vview fil.xlimg

```
2. vexfft - m - l tedf.fft of = tedf.xlimg
```

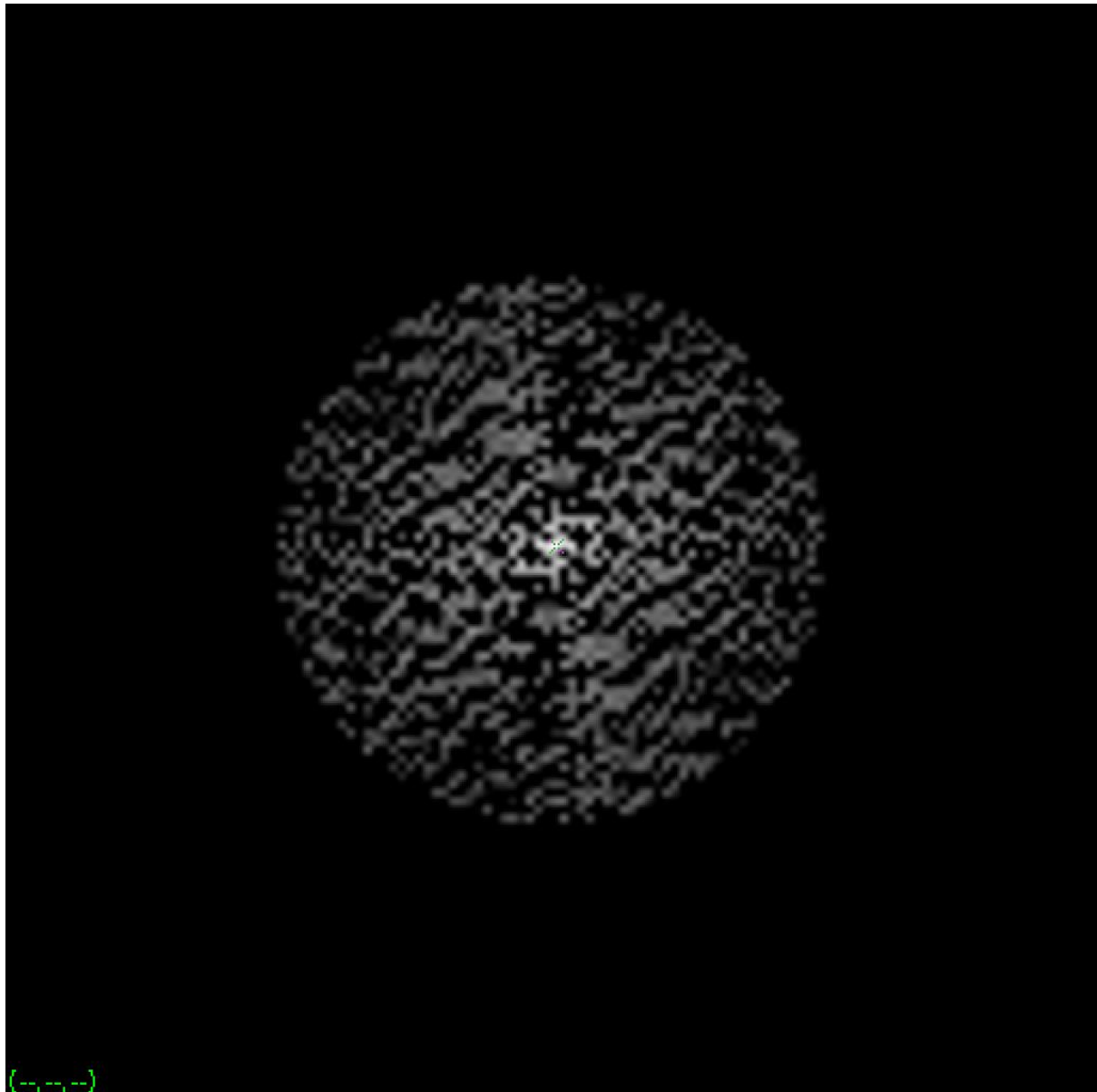



Figure8: vview tedf.xlimg

Now compute the *inverse FFT* and display the filtered image:

```
1. vfft - i if = tedf.fft of = ted.f
```

2.10 to view the ted.f, use vexfft to display the log magnitude of it.

```
1. vexfft - m - l ted.f of = tedinv.xlimg
```



Figure9: vview tedinv.xlimg

See Appendix:

Another Figure: Statistics for ted.f

Another Figure: vplot ted.f

2.11 Discuss the differences between the *original* and the *filtered* image.

For the ted image, the original image have more low frequency part, after frequency domain filter, it remove low frequency part and keep the high frequency part. For the circle we generated, the original image nearly has no high frequency noise, the filtered output has many high frequency noise.

To verify my gusee, I use vplot to show the histogram of input ted image and filtered output ted image as below:

```
1. vplot tedinv.xlimg
```

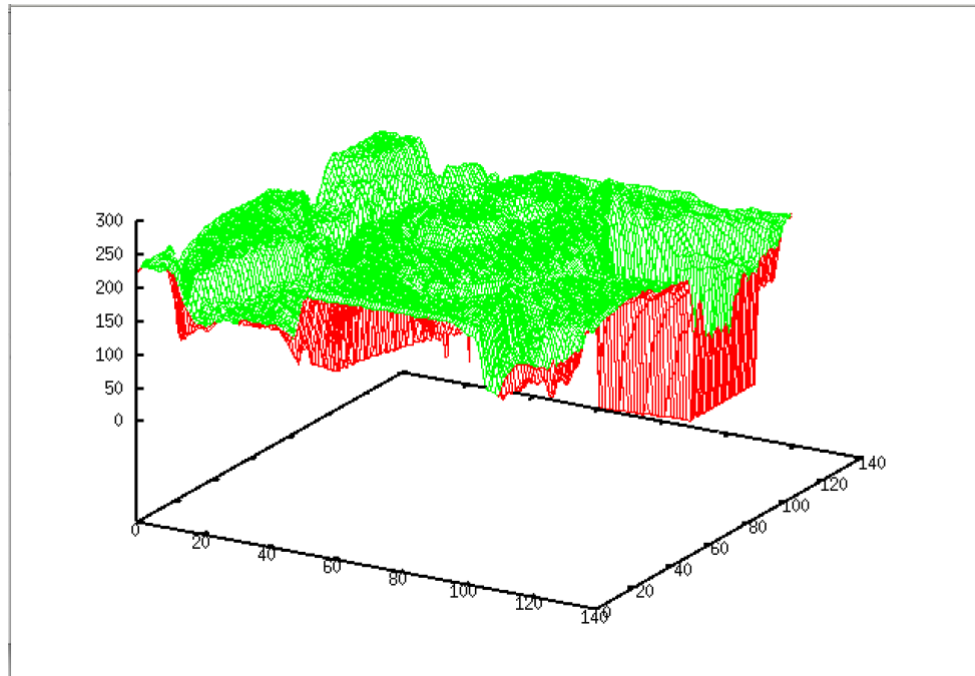


Figure10: frequency histogram of tedinv.xlimg

```
1. vplot ted
```

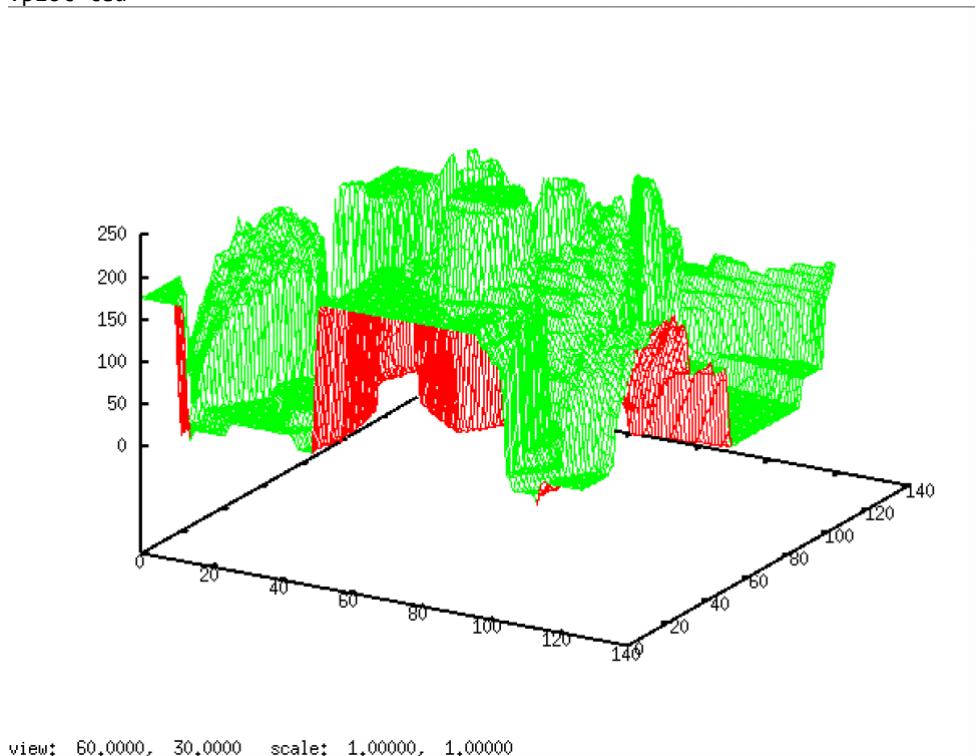


Figure11: frequency histogram of ted.xlimg

From the histogram, we can tell the frequency domain filter remove the frequency below 100.

2.12 Repeat the above filtering procedure with the image sshtl.

1. vfix - float if = sshtl | vfft of = sshtl.fft
2. vop - mul if = sshtl.fft ig = fil of = sshtlf.fft
3. vfft - i if = sshtlf.fft of = sshtl.f

then use vexfft to display the image

2. vexfft - m - l sshtl.fft of = sshtl.xlmag
3. vexfft - m - l sshtlf.fft of = sshtlf.xlmag
4. vexfft - m - l sshtl.f of = sshtlinv.xlmag



Figure12: vview the filtered output image sshtlinv.xlmag

```

Image Statistics for sst1.f

Global Statistics Summary

1 images in file      128 x 128 (0,0)

65536      (32764      ) number of pixels
16510.4    (16510.4    ) maximum pixel value
-3058.63   (-3058.63   ) minimum pixel value
19569      (19569      ) range
0.28711    (0.574289   ) mean pixel value
4877.67    (9756.37    ) variance
69.8403    (98.7743    ) standard deviation
201.583    (142.527    ) skewness
47806.1    (23899.3    ) kurtosis

```

Figure13: Statistics for sst1.f

1. vplot sst1.f

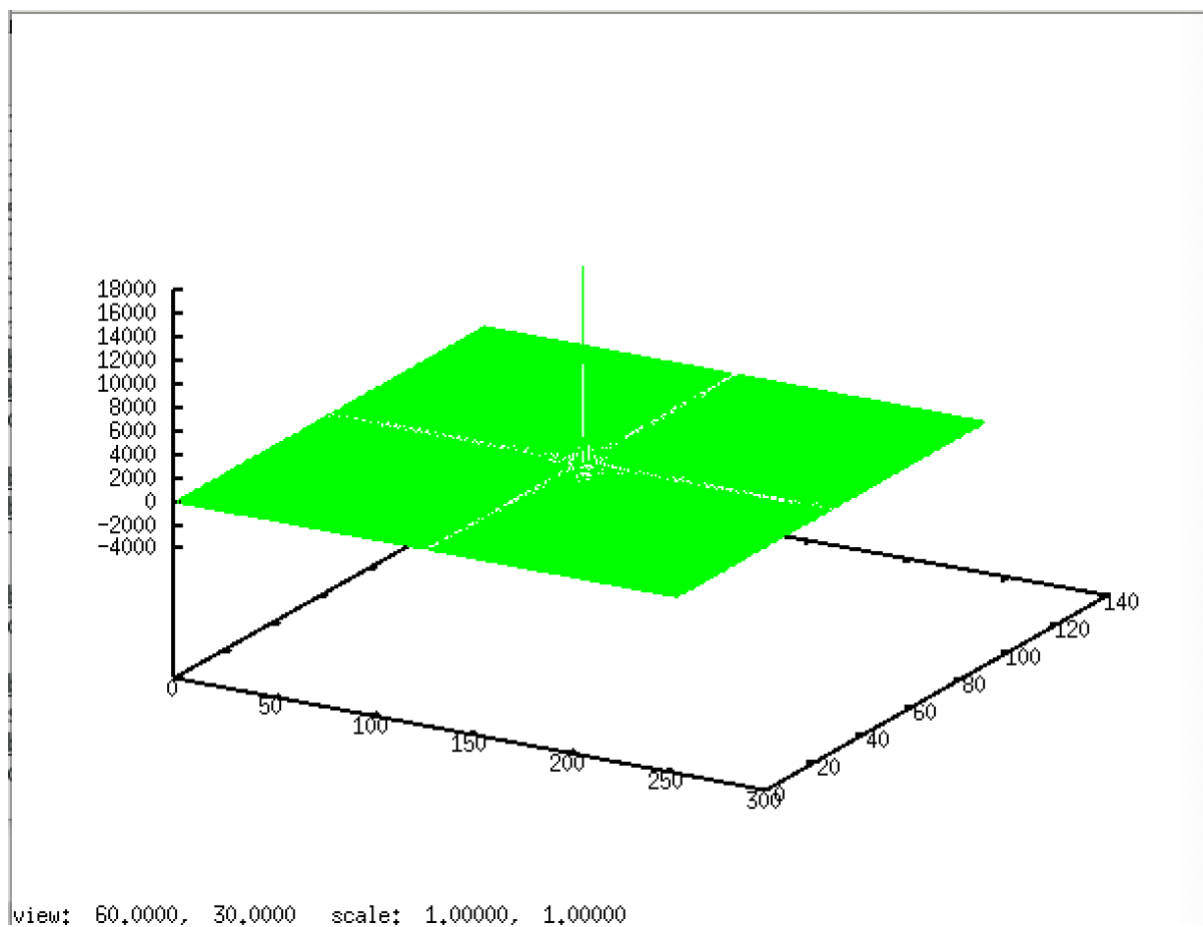


Figure14: frequency histogram of sst1.f

3. Spatial Domain Filtering

3.1 First generate the spatial filter :

```
1. vgenim r = 2, 2 x = 10 y = 10 hi = 1 of = sf1
1. vconv if = sf1 k = sf1 | vfix - float of = sf2
```

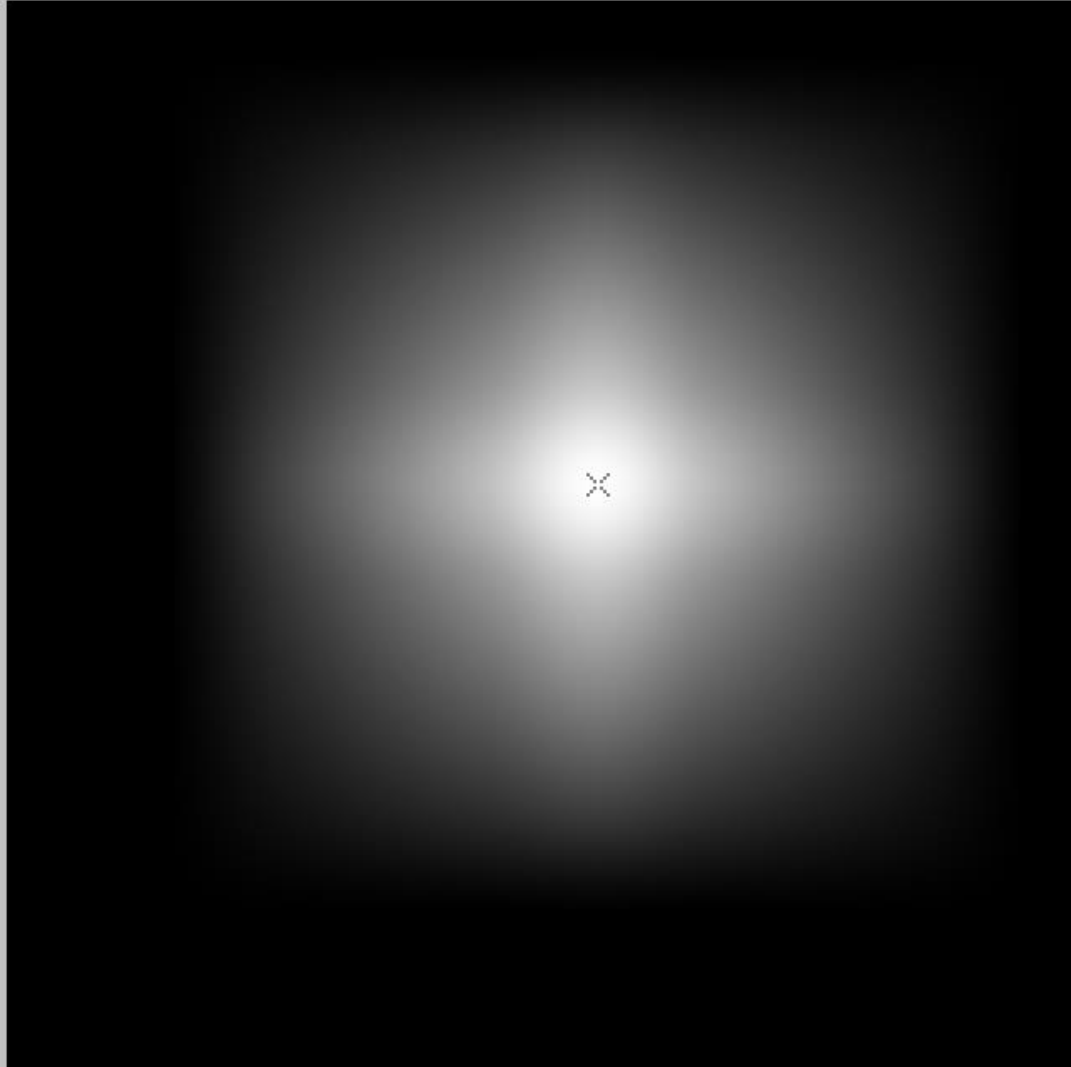


Figure15: view the after convolution output image sf2

```
1. Plot sf2
```

See Appendix:

Another Figure: frequency histogram of sf2

Another Figure: frequency histogram of sf1

```
1. vplot sf1
```

3.2 What does the spatial filter look like?

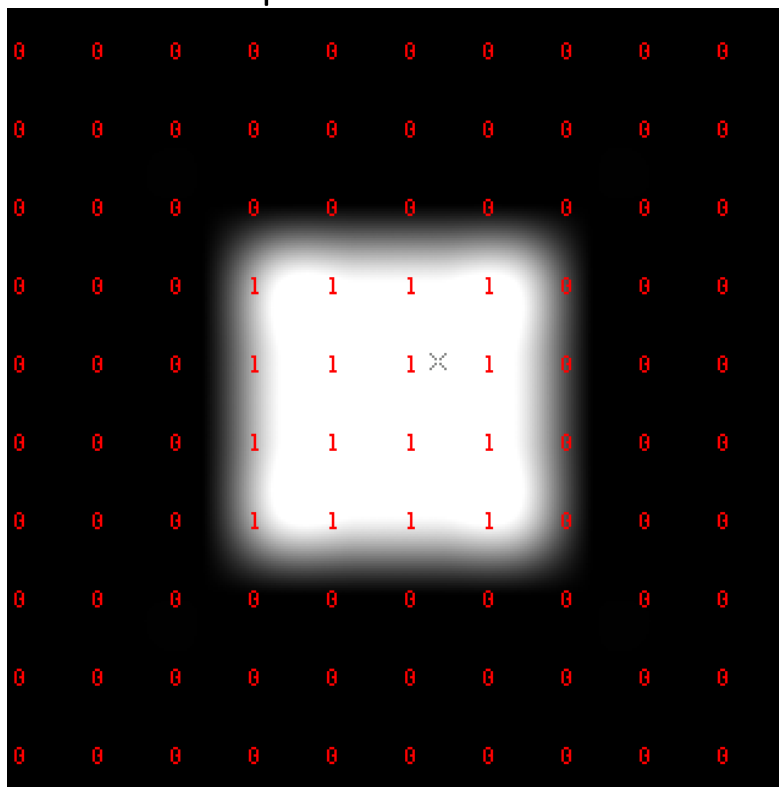


Figure16: vview the spatial filter sf1

1. Plot sf1

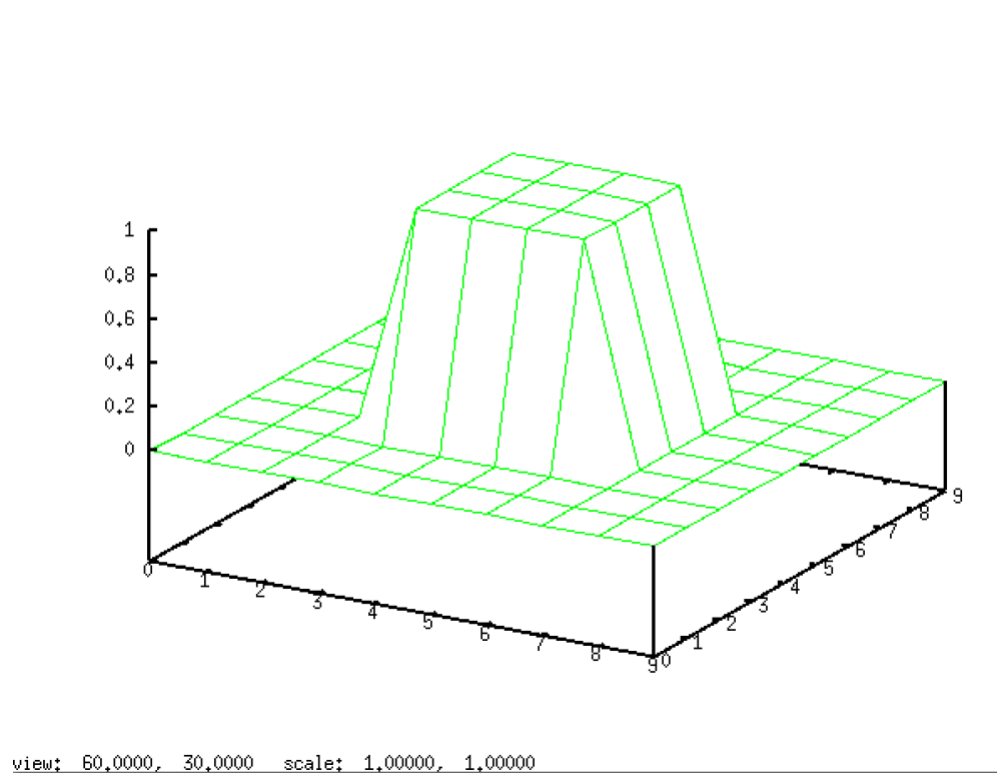


Figure17: the frequency histipgram of spatial filter sf1

3.3 How was it created?

First generate a 10X10 byte test image, which contains a 2X2 rectangle, the high pixel value is 1.

Then convolve the test image with itself (the kernel matrix is the test image itself), out put the result in float point type.

Next print the basic statistics of the output we get in last step.

Then traverse all pixels that are multiplied by 0.0039(we set), then output the result as sfil

Finally plot the histogram of sfil.

3.4 How was the above value of the tf= parameter selected ?

(hint: use `vpis` or *Tools→Image Statistics* `sfil` and consider the mean pixel value.)

1. `vpix if = sf2 tf = 0.0039 of = sfil`

after I use *Tools→Image Statistics* to show the basic statistics of sfil as below

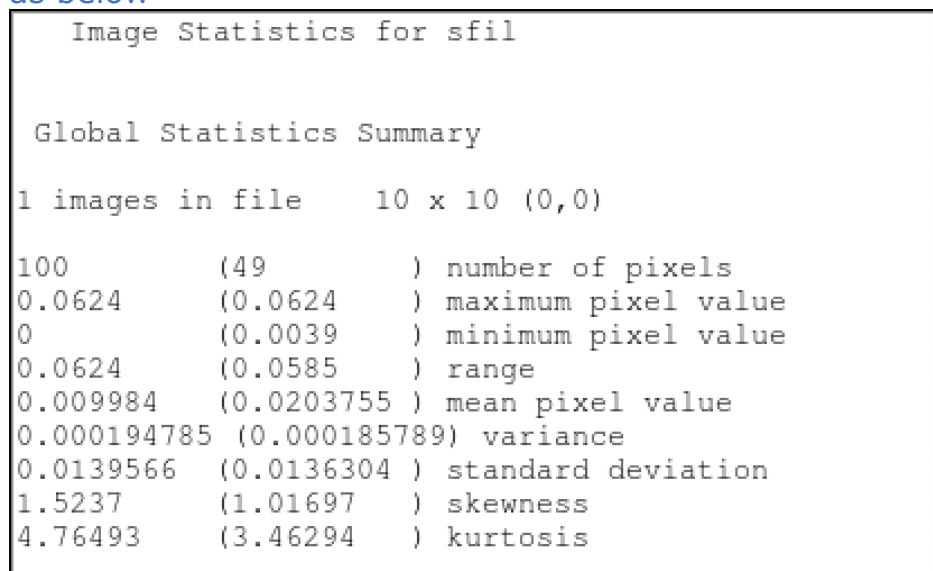


Figure18: Statistics for sfil

The mean value is 0.009984.

Now perform the convolution :

1. `vconv if = ted k = sfil of = ted.sf`

3.5 Discuss the differences between the *original* and the *filtered* image.

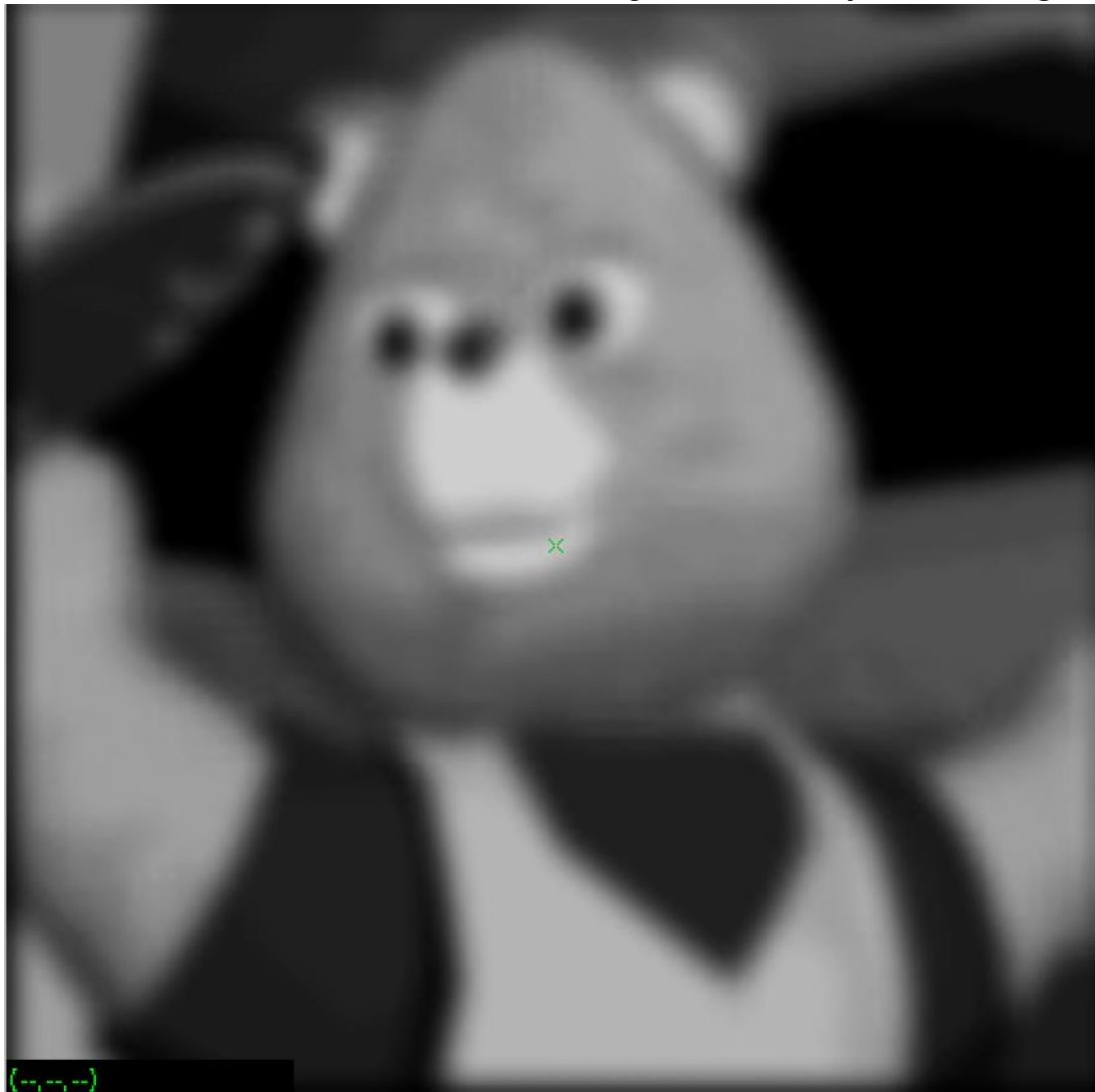


Figure19: vview ted.sf

The original image of ted has more high frequency domain, in human eyes, they are rigid edges, after spatial Filtering, edges are removed, and edges become very smooth, the whole image is more consistent in color, but looks blur.

3.6 Also compare *ted.sf* with *ted.f* and discuss the differences.

```
1. vexfft - m - l ted.sf of = tedsf.xlimg
```



Figure20: vview tedsf.xlimg

After filter, the ted.sf looks like the result of cclable, all background become the same black color, and the color range become less, there are only 6 colors in the filtered ted image, more colors in ted.f
And the mean pixel in ted.sf is less than ted.f.

To verify my gusee, I see the statistics of these two images.

```

Image Statistics for ted.sf

Global Statistics Summary

1 images in file      128 x 128 (0,0)

16384      (14893      ) number of pixels
206         (206         ) maximum pixel value
0           (1           ) minimum pixel value
206         (205         ) range
85.3939     (93.9431     ) mean pixel value
3425.26     (2965.04     ) variance
58.5257     (54.4522     ) standard deviation
0.144108    (0.108091    ) skewness
1.76392     (1.76773     ) kurtosis

```

Figure21: Statistics for ted.sf

```

Image Statistics for ted.f

Global Statistics Summary

1 images in file      128 x 128 (0,0)

65536      (32768      ) number of pixels
267.325     (267.325     ) maximum pixel value
-18.6363    (-18.6363     ) minimum pixel value
285.961     (285.961     ) range
27.5915     (55.1829     ) mean pixel value
3433.65     (5344.72     ) variance
58.5973     (73.1076     ) standard deviation
2.02803     (0.95635     ) skewness
5.80257     (2.4508      ) kurtosis

```

Figure22: Statistics for ted.f

the mean pixel value in ted.sf is 28, the mean value in ted.f is 85. My guess is right.

3.7 What does the image *fi* look like in the spatial domain? Can you make a good 2D graph of it?

(Hint, useful commands are: `vfft`, `vchan`, `vfix`, `vclip` and `vplot`.)

1. vfft - i if = fil of = fil.f
2. vexfft - m - l fil.f of = filinv.xlmag
3. vplot filinv.xlmag

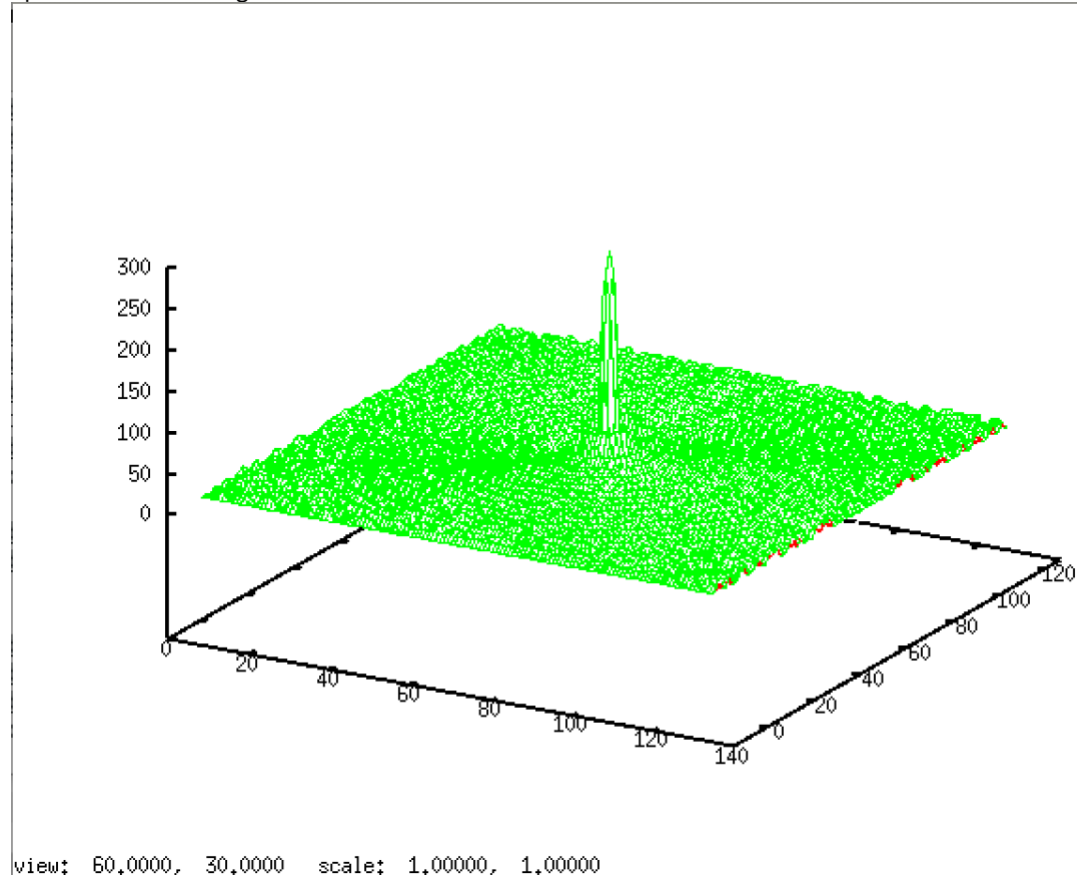


Figure23: spatial domain 2D image filinv.xlmag

The fil looks like a sinc function.

Appendix:

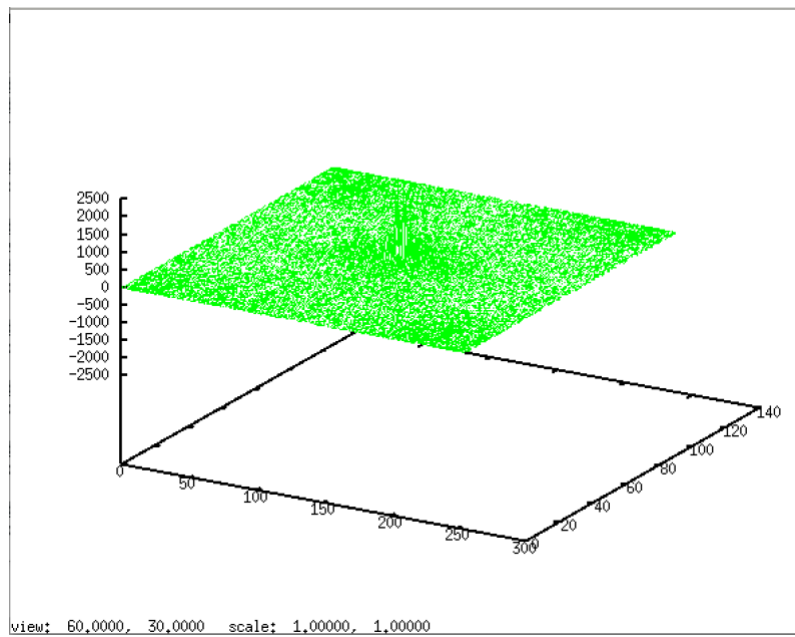
Image Statistics for ted.xfft

Global Statistics Summary

1 images in file 128 x 128 (0,0)

65536	(32764)	number of pixels
2275.97	(2275.97)	maximum pixel value
-2157.06	(-2157.06)	minimum pixel value
4433.03	(4433.03)	range
0.253191	(0.506443)	mean pixel value
1176.78	(2353.72)	variance
34.3042	(48.5151)	standard deviation
14.4039	(10.1697)	skewness
1751.54	(875.546)	kurtosis

FigureN: Statistics for ted.xfft



FigureN: frequency histogram of ted.xfft

Image Statistics for ted.xlmag

Global Statistics Summary

1 images in file 128 x 128 (0,0)

16384	(8381)	number of pixels
255	(255)	maximum pixel value
0	(3)	minimum pixel value
255	(252)	range
28.986	(56.6646)	mean pixel value
1454.21	(1274.43)	variance
38.1341	(35.6991)	standard deviation
1.28031	(0.769815)	skewness
4.16485	(3.8303)	kurtosis

FigureN: Statistics for ted.xlmag

```

Image Statistics for fil

Global Statistics Summary

1 images in file      128 x 128 (0,0)

65536      (6456      ) number of pixels
1          (1          ) maximum pixel value
0          (1          ) minimum pixel value
1          (0          ) range
0.0985107  (1          ) mean pixel value
0.0888064  (0          ) variance
0.298004   (0          ) standard deviation
2.69452    (0          ) skewness
8.26045    (0          ) kurtosis

```

FigureN: Statistics for fil

```

Image Statistics for tedf.fft

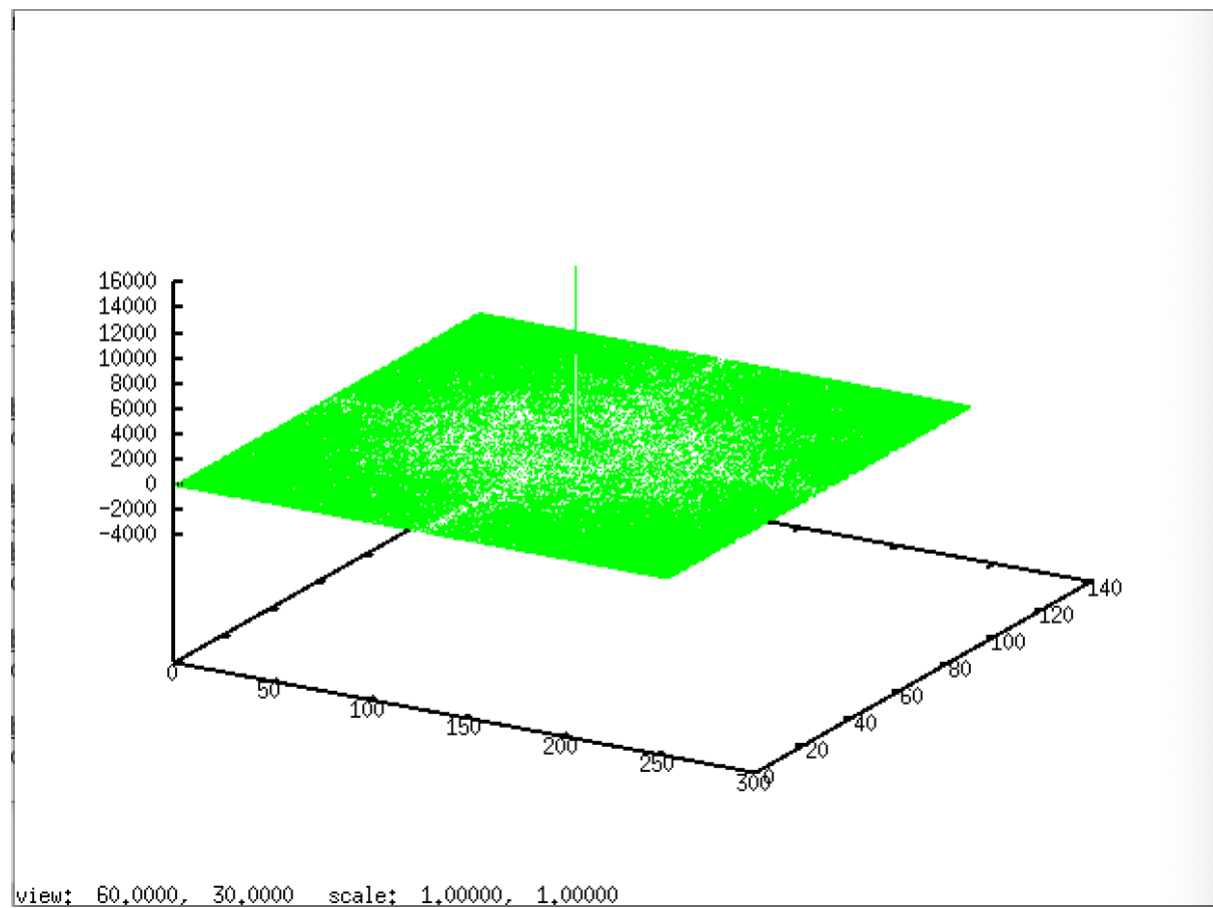
Global Statistics Summary

1 images in file      128 x 128 (0,0)

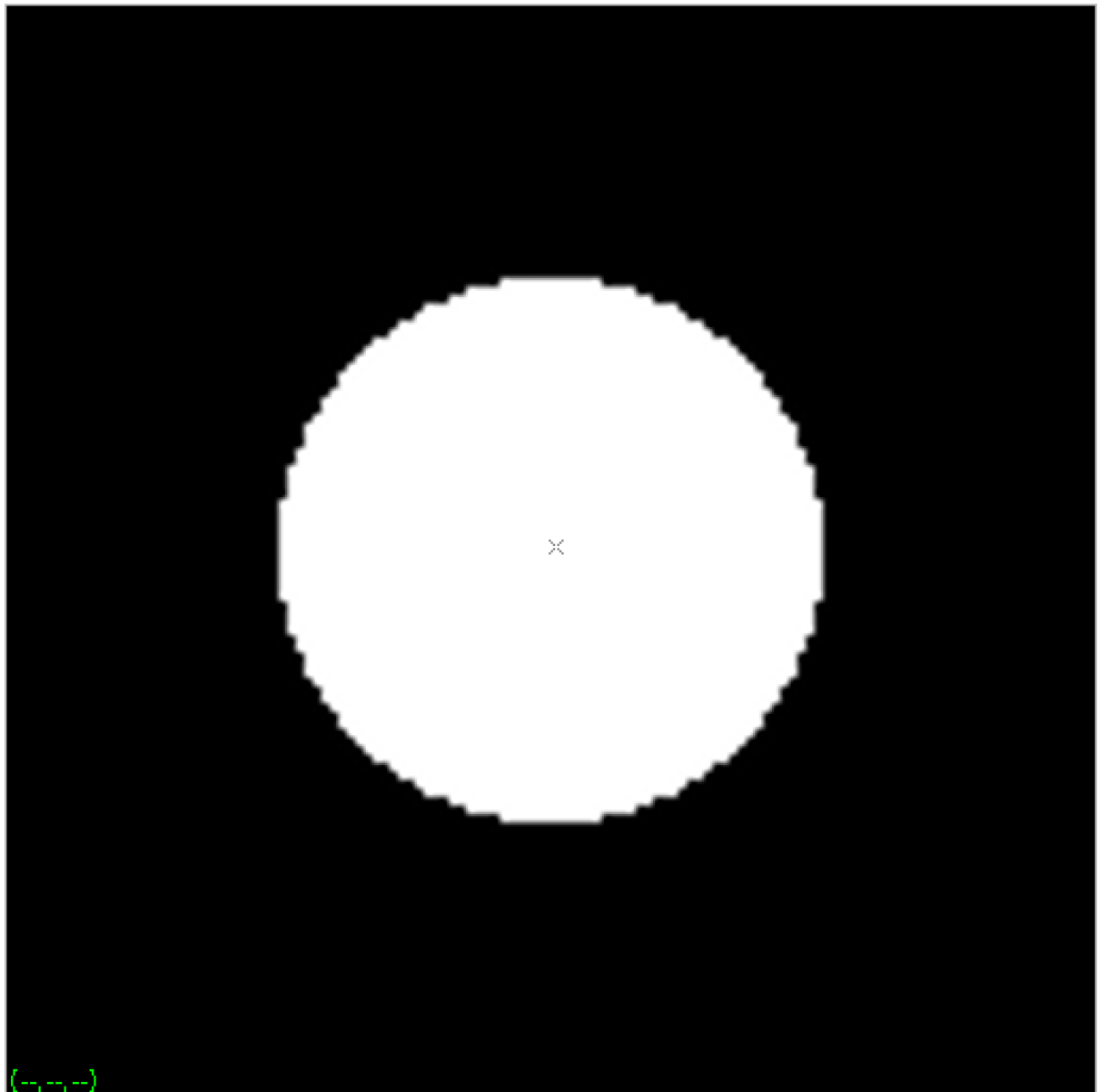
65536      (6455      ) number of pixels
14126.8    (14126.8    ) maximum pixel value
-2157.06   (-2157.06    ) minimum pixel value
16283.9    (16283.9    ) range
0.432645   (4.39254    ) mean pixel value
4194.76    (42570.9    ) variance
64.7669    (206.327    ) standard deviation
160.464    (50.3334    ) skewness
34670.5    (3413.81    ) kurtosis

```

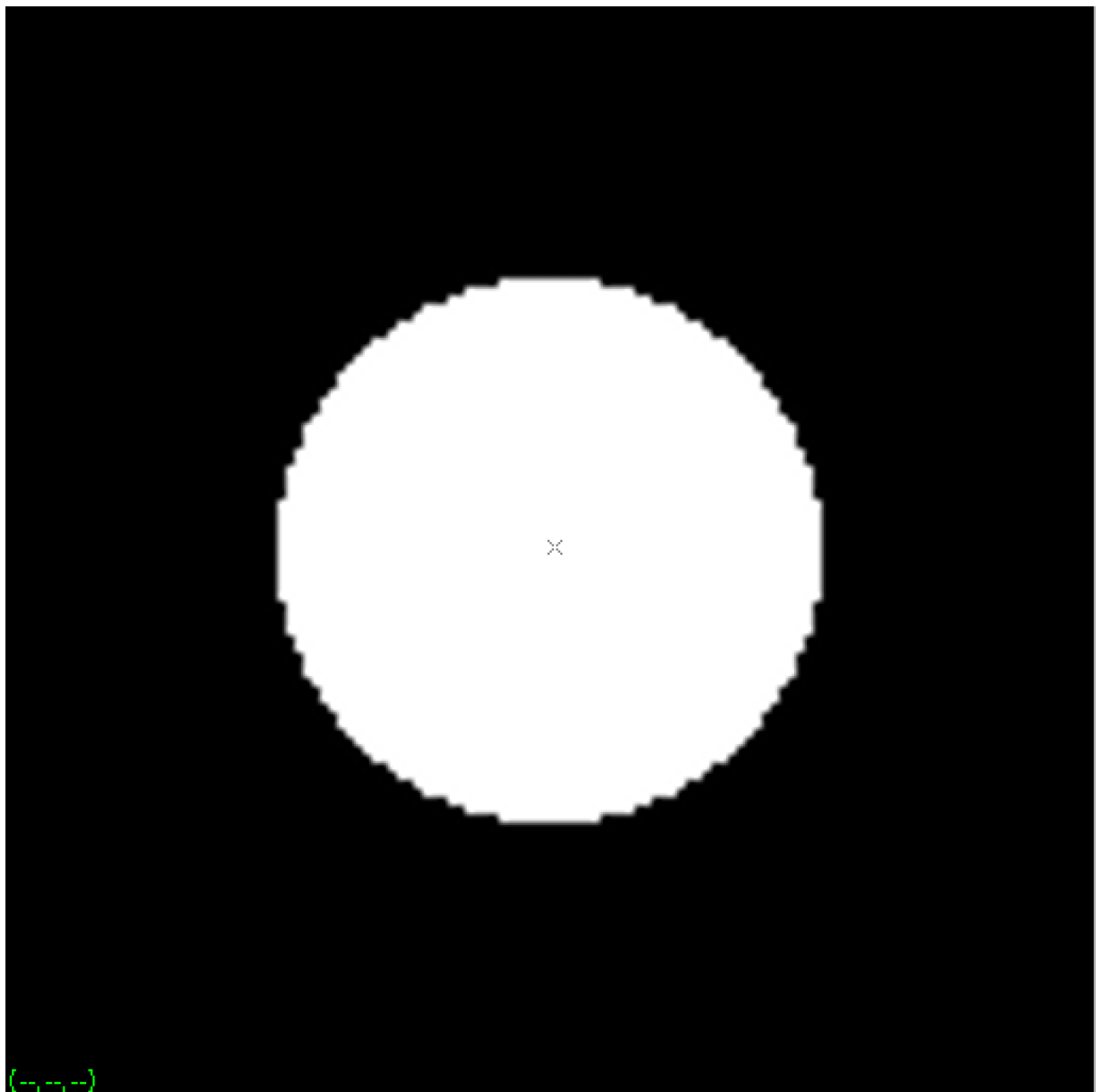
FigureN: Statistics for tedf.fft



FigureN: vplot ted.fft



FigureN: vview f1.xlmag



FigureN: vview fil.xlmag

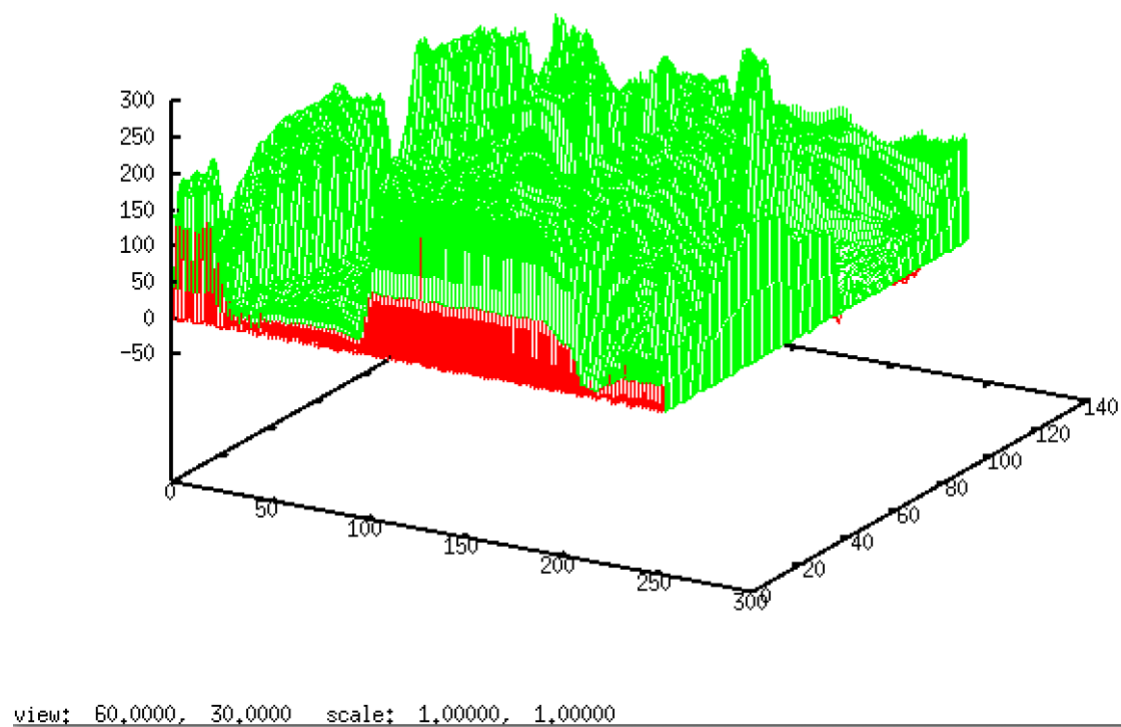
Image Statistics for ted.f

Global Statistics Summary

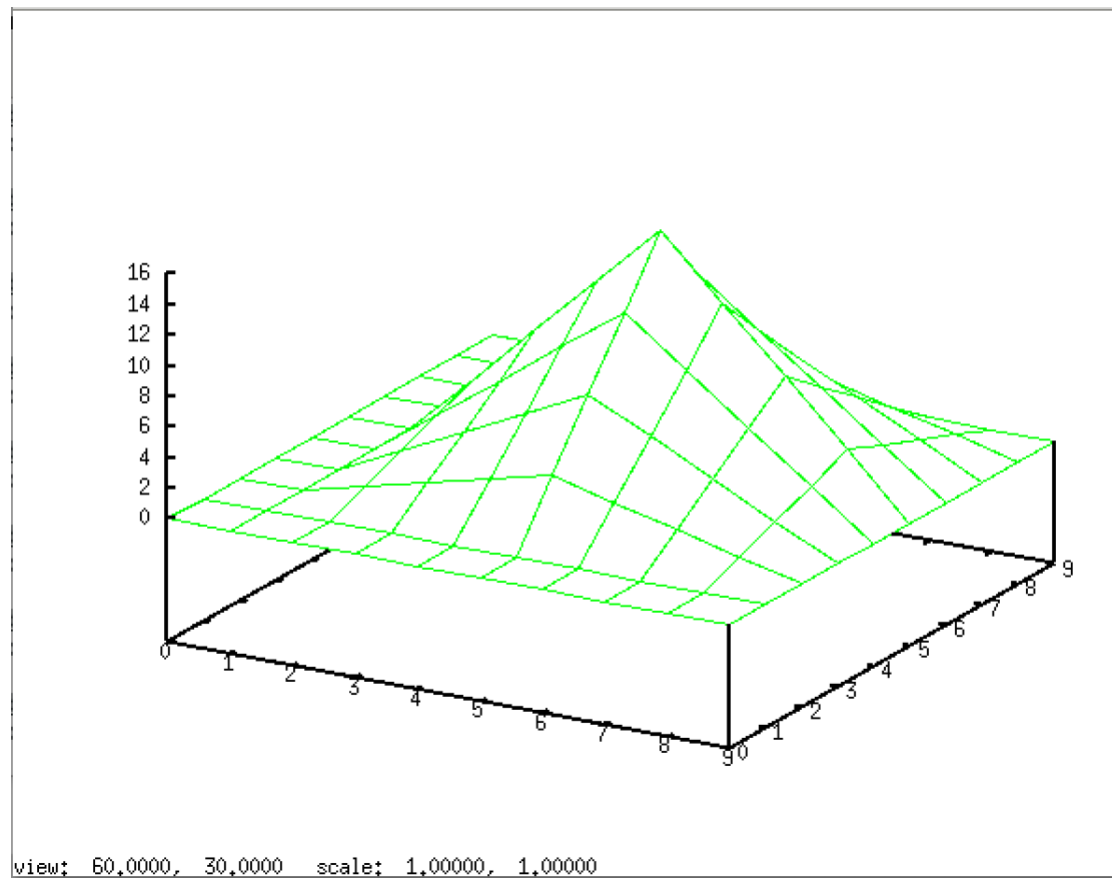
1 images in file 128 x 128 (0,0)

65536	(32768)	number of pixels
267.325	(267.325)	maximum pixel value
-18.6363	(-18.6363)	minimum pixel value
285.961	(285.961)	range
27.5915	(55.1829)	mean pixel value
3433.65	(5344.72)	variance
58.5973	(73.1076)	standard deviation
2.02803	(0.95635)	skewness
5.80257	(2.4508)	kurtosis

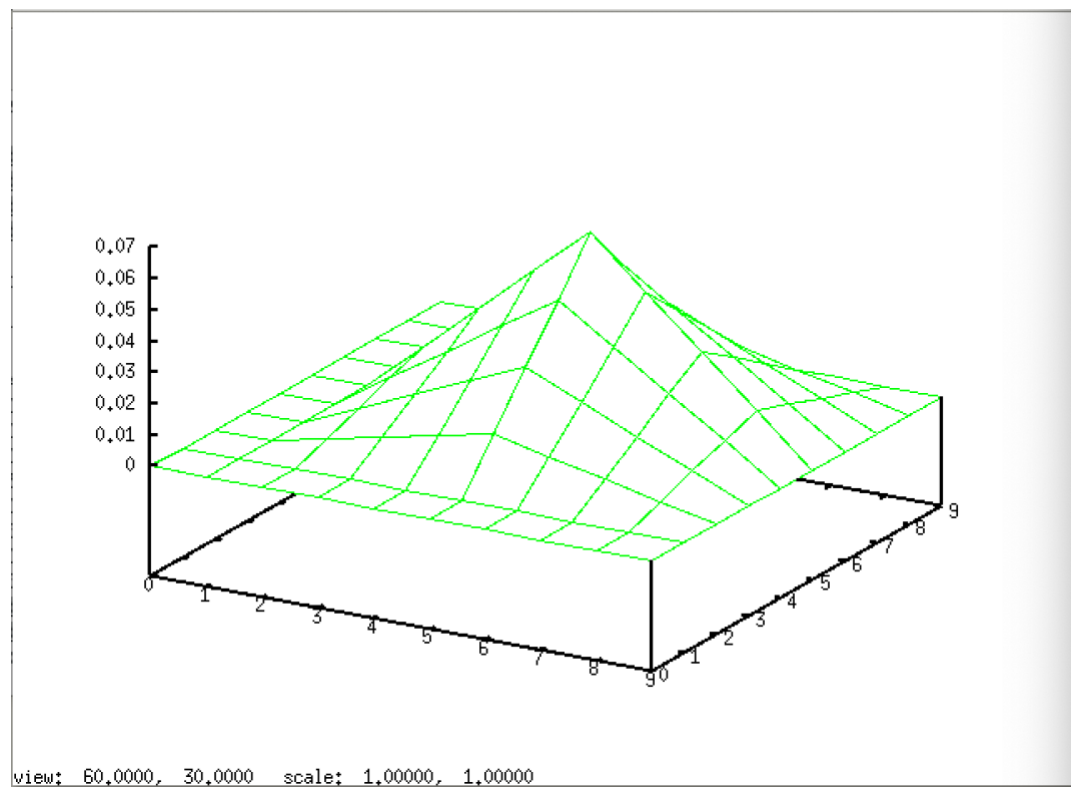
FigureN: statistics for ted.f



FigureN: vplot ted.f



FigureN: frequency histogram of sf2



FigureN: frequency histogram of sf1