Denoising images using Fast Fourier Methods

A novel way to filter images in the fourier domain

Naksatra Kumar Bailung

2023-01-11 Wed

Newcastle University

Outline

Newcastle University

Preliminary analysis on fourier form of images

- Clipping across the ends of the channels
- Grayscale consversion uses this formula :

ITU-R 601-2

$$L = R * 0.299 + G * 0.587 + B * 0.114$$

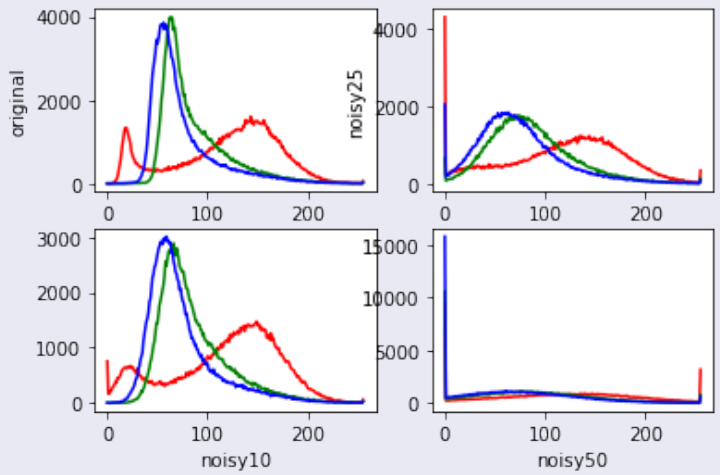
Newcastle University

Noise patterns

$$n(x, y) = g(x, y) - f(x, y)$$

$n(x, y)$ is defined as the noise defined either by a single value $\in \mathcal{R}$ or a 3-tuple $\in \mathcal{R}^3$ as is $g(x, y)$ which is the noisy_σ value and $f(x, y)$ which is the original$_{\text{image}}$.

Colored Image as it gets noisier

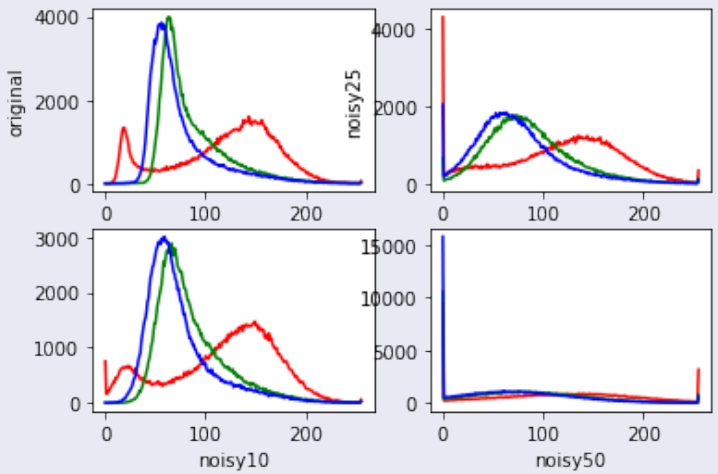


Noise patterns

$$n(x, y) = g(x, y) - f(x, y)$$

$n(x, y)$ is defined as the noise defined either by a single value $\in \mathcal{R}$ or a 3-tuple $\in \mathcal{R}^3$ as is $g(x, y)$ which is the noisy_σ value and $f(x, y)$ which is the original$_{\text{image}}$.

Colored Image as it gets noisier

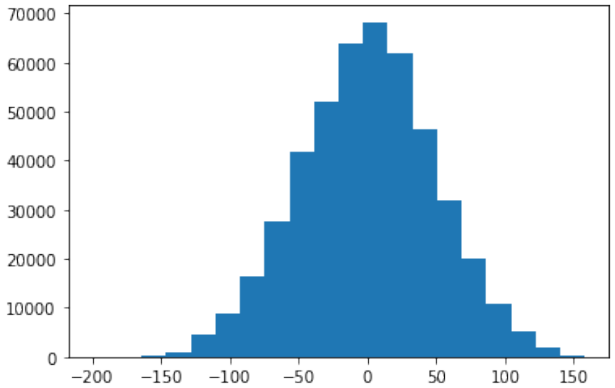


Image

Noise patterns

Plotting the histogram of the difference of the pixel values from the original and noisy values

- $\hat{\mu} = 0$
- $\sigma = 48.80873281190002$



Noise equation

$$n(x, y) \sim \mathcal{N}(0, \sigma)$$

Henceforth, we take $G(u, v)$ to be the 2d discrete fourier transformation of the noisy image, $F(u, v)$ be the 2d discrete fourier transform of the original image calculating we obtain:

Theorem (2d Fourier transform of Spaitally independent gaussian noise)

$$\sum_{x=0}^{w}\sum_{y=0}^{h} g(x, y)\ exp\left(-2\pi i\left(\frac{ux}{w}+\frac{vy}{h}\right)\right) = \sum_{x=0}^{w}\sum_{y=0}^{h}\left(f(x, y)+\mathcal{N}(0, \sigma^2)\right) exp\left(-2\pi i\left(\frac{ux}{w}+\frac{vy}{h}\right)\right)$$

$$G(u, v) = F(u, v) + \sum_{x=0}^{w}\sum_{y=0}^{h}\mathcal{N}(0, \sigma^2) exp\left(-2\pi i\left(\frac{ux}{w}+\frac{vy}{h}\right)\right)$$

Now we have to remember even though $\mathcal{N}(0, \sigma^2)$ remains unaffected by spatial dimensions, it will generate different values for each of x,y. Hence we cannot take it out of the equation for all x and

Newcastle University

Metrics and skipping preprocessing

- A fourier transform and no processsing in between and back preserves the pixel colors
- Preprocessing the histogram leads to worse values over normal values due to the histogram inconsistency.

nFFT over each channel FFT for colored channels

While applying the same filter on the nFFT domain yields same results however we can control nFFT over the third dimension to generate even better images.

Figure: R = 75, G = 75, B = 75

nFFT over each channel FFT for colored channels

While applying the same filter on the nFFT domain yields same results however we can control nFFT over the third dimension to generate even better images.

Figure: R = 75, G = 75, B = 75

Figure: (nFFT) R = 50, G = 75, B = 50

Newcastle University

nFFT over each channel FFT for colored channels

While applying the same filter on the nFFT domain yields same results however we can control nFFT over the third dimension to generate even better images.

Figure: R = 75, G = 75, B = 75

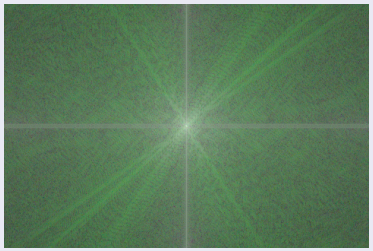
Figure: (nFFT) R = 50, G = 75, B = 50

Figure: (FFT) R = 50, G = 75, B = 50

Newcastle University

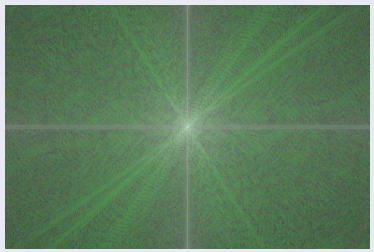
nFFT of the first plane image

Amplitude (original)

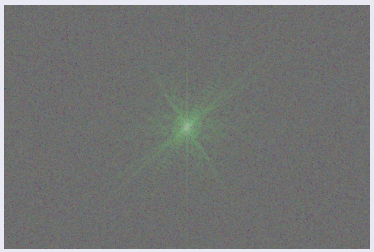


nFFT of the first plane image

Amplitude (original)

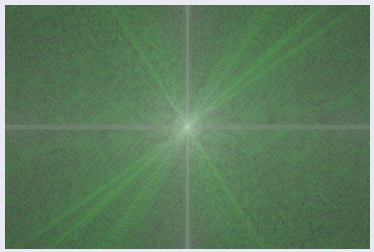


Amplitude (noise50)

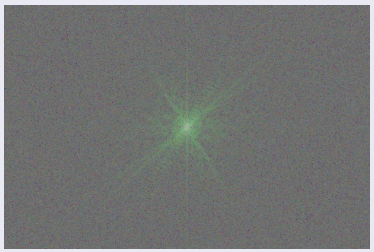


nFFT of the first plane image

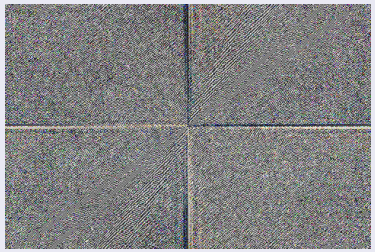
Amplitude (original)



Amplitude (noise50)

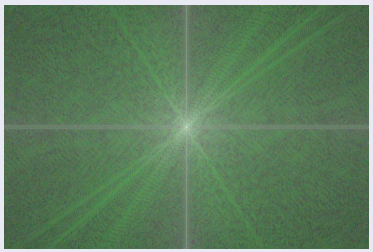


Phase (original)

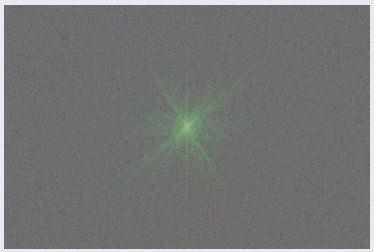


nFFT of the first plane image

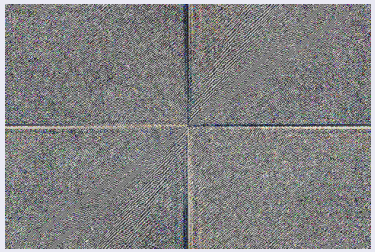
Amplitude (original)



Amplitude (noise50)



Phase (original)



Phase (noise50)

Newcastle University

Features to preseve

Usually an image has horizontal and vertical features that are most prominent and hence show up as the vertical and horizontal lines in a fft image. But what if we take this to the extreme.

Original images

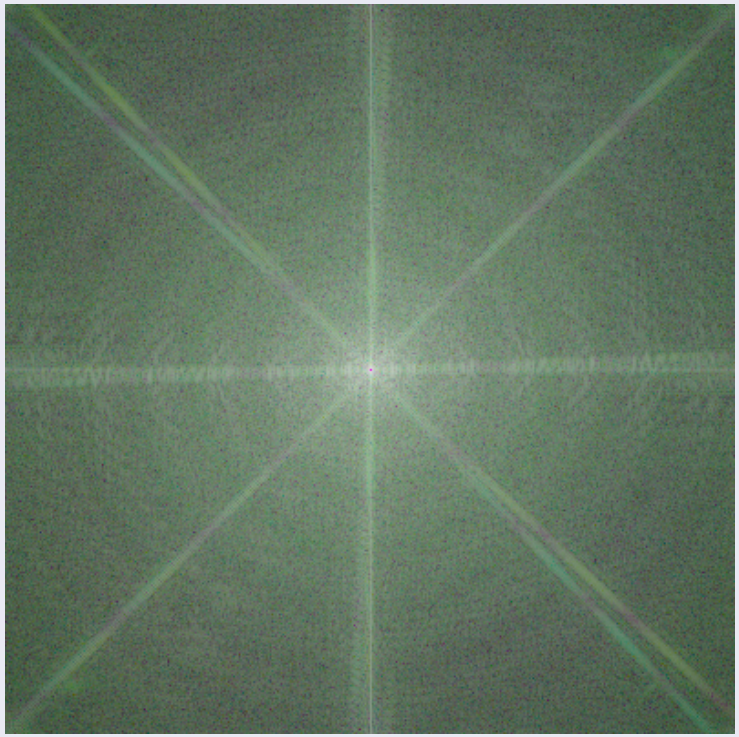
Newcastle University

Features to preseve

Usually an image has horizontal and vertical features that are most prominent and hence show up as the vertical and horizontal lines in a fft image. But what if we take this to the extreme.

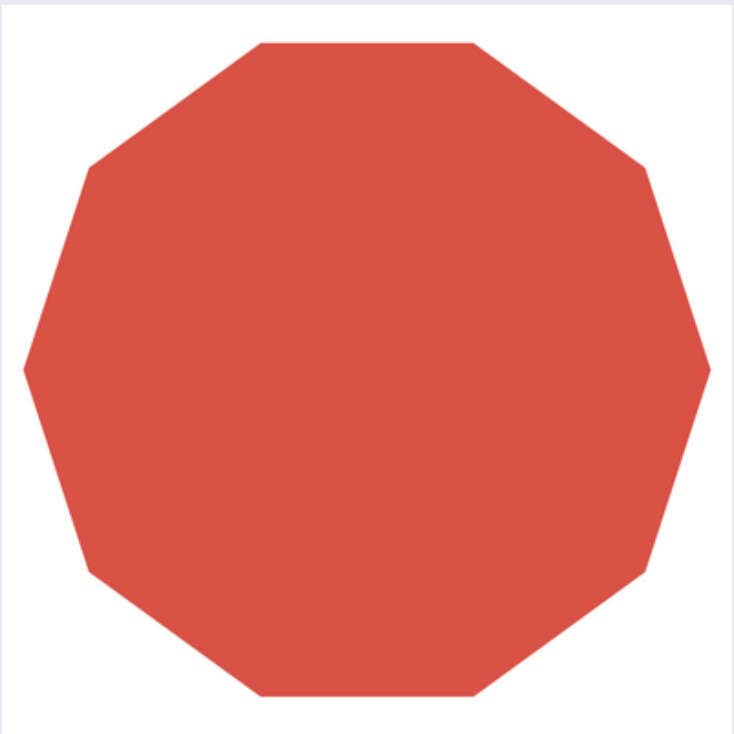
Original images

Fourier amplitude



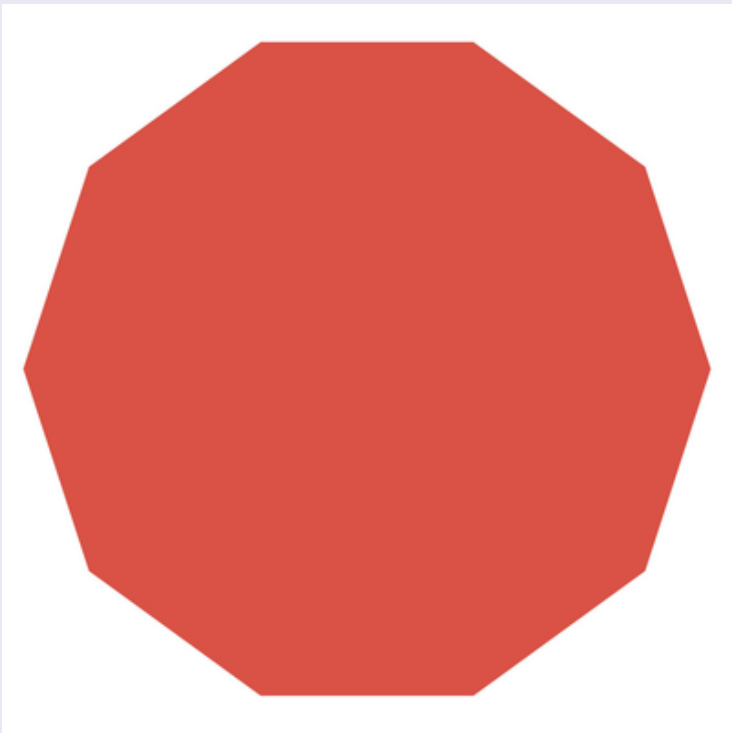
Decagon

Original images

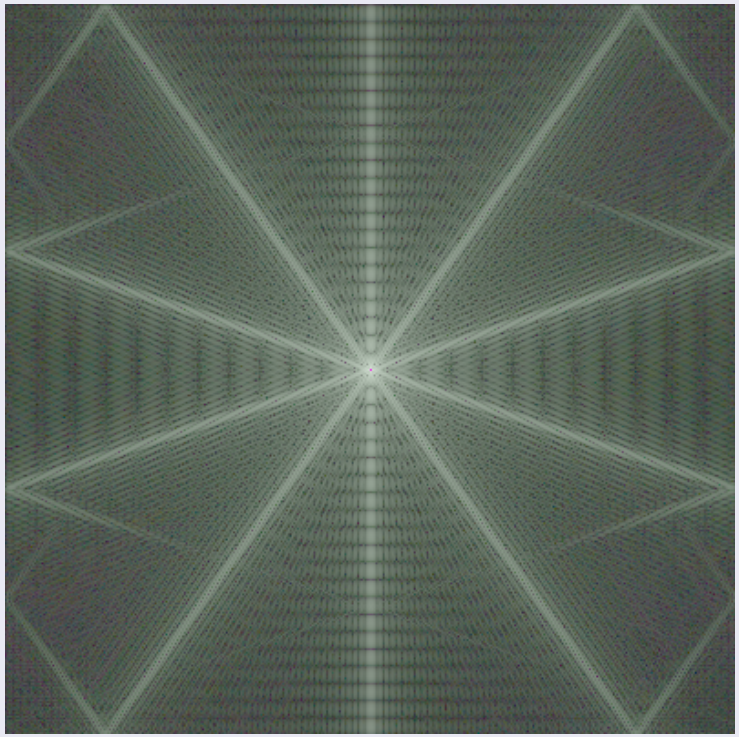


Decagon

Original images



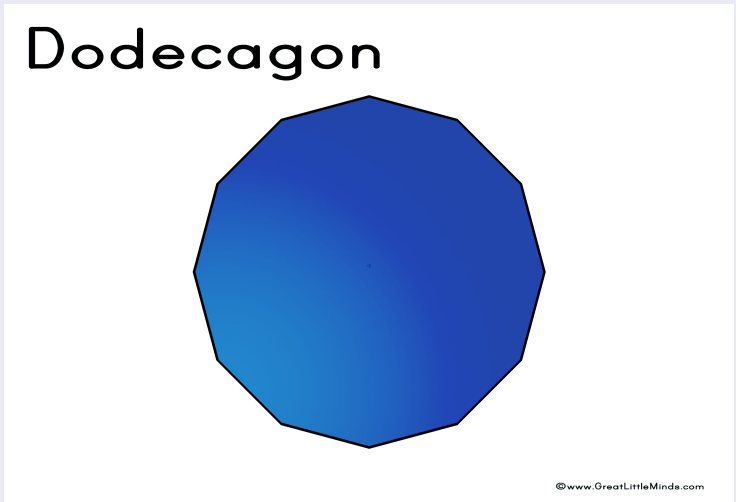
Fourier amplitude



Newcastle University

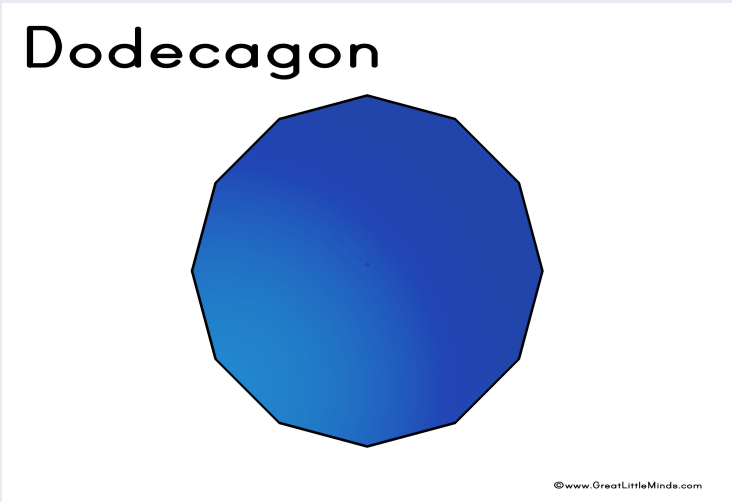
Dodecagon

Original images

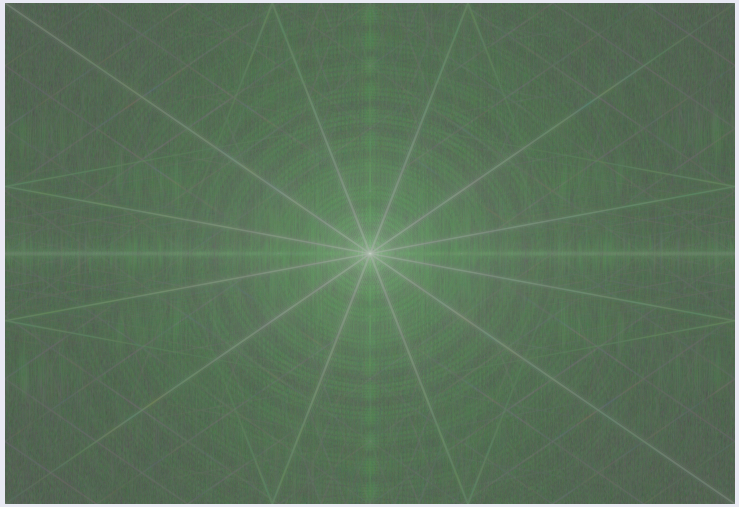


Dodecagon

Original images



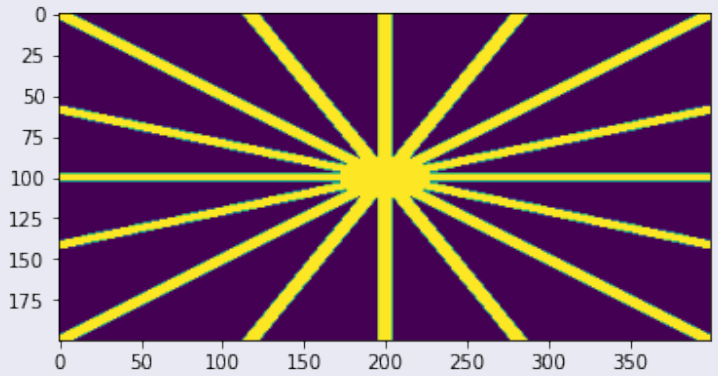
Fourier amplitude



Filters in 2d/3d domain

Since we need to account for a pattern that considers the horizontal vertical and diagonal, we generate a pattern by rotating a pattern which covers all the indicated angles. We do so by using affine transformations and SO_3 rotations.
We extend our filter to the 3d domain using the same pattern over in the 3-dimensional domain

2d filters

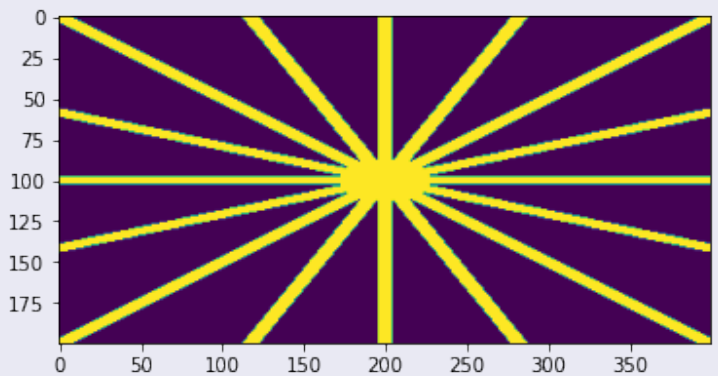


Newcastle University

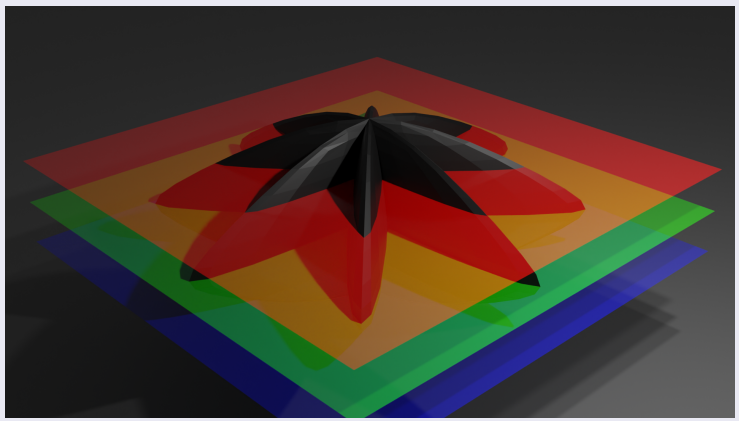
Filters in 2d/3d domain

Since we need to account for a pattern that considers the horizontal vertical and diagonal, we generate a pattern by rotating a pattern which covers all the indicated angles. We do so by using affine transformations and SO_3 rotations.
We extend our filter to the 3d domain using the same pattern over in the 3-dimensional domain

2d filters



3d filter



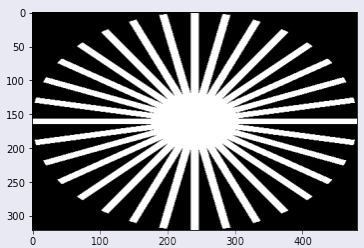
Another blurred filter we use for comparison

Gaussian filter over our designed filter

2d filters

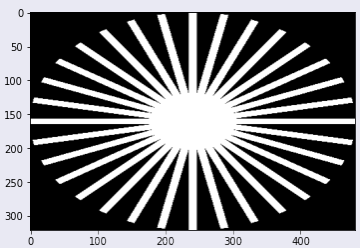
Filters chosen

2d filters

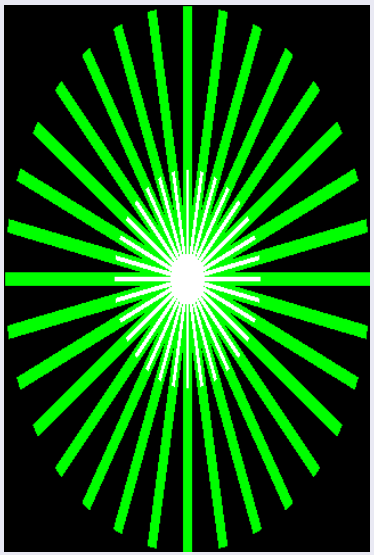


Filters chosen

2d filters

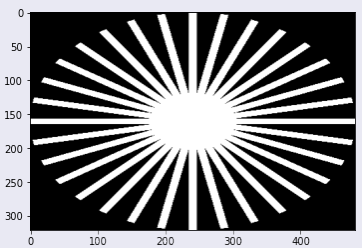


3d filter

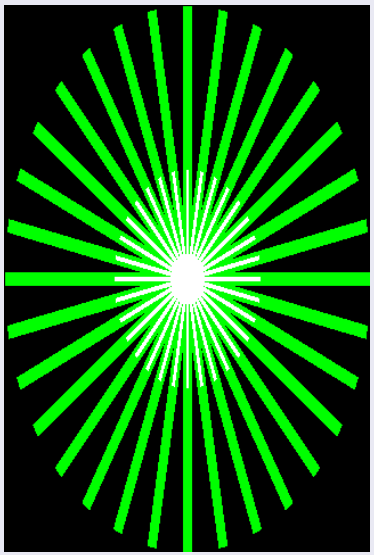


Filters chosen

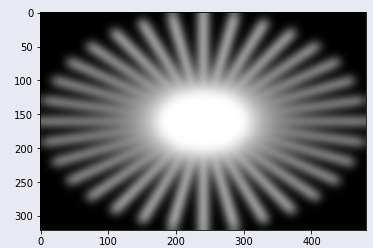
2d filters



3d filter



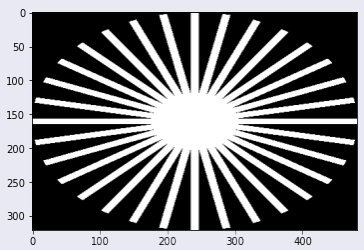
2d blurred filter



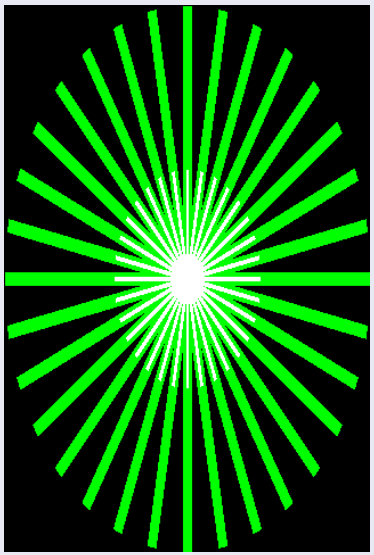
Newcastle University

Filters chosen

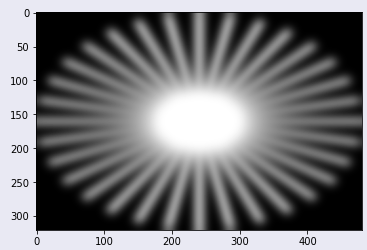
2d filters



3d filter



2d blurred filter



3d blurred filter

Newcastle University

Final result (Grayscaled SSIE)

Results produced are averaged over the whole dataset we have obtained

Dataset	noisy10	noisy25	noisy50
GS SSIE	0.7962	0.6303	0.4299
GS SSIE (+ Only amp)	0.7962	0.6303	0.4299
GS SSIE (+ Median)	0.8098	0.7302	0.5817
GS SSIE (+ Only amp + Median)	0.8098	0.7302	0.5817
GS SSIE (+ Blur)	0.8160	0.7143	0.5434
GS SSIE (+ Blur + Only amp)	0.8943	0.7534	0.5487
GS SSIE (+ Blur + Median)	0.8101	0.7480	0.6182
GS SSIE (+ Blur + Only amp + Median)	0.8448	0.7711	0.6276
Mean filter SSIE	0.8442	0.7535	0.5893
Median filter SSIE	0.8460	0.7171	0.5241
BayesShrink SSIE	0.9089	0.7759	0.6409
VisuShrink SSIE	0.6810	0.5566	0.4709
Origial SSIE	0.83176	0.5514	0.3265

Newcastle University

Final result (Grayscaled MSE)

Results produced are averaged over the whole dataset we have obtained

Dataset	noisy10	noisy25	noisy50
GS MSE	148.69	228.20	498.87
GS MSE (+ Only amp)	148.69	228.20	498.87
GS MSE (+ Median)	141.61	170.07	274.35
GS MSE (+ Only amp + Median)	141.61	170.07	274.35
GS MSE (+ Blur)	140.70	170.36	304.63
GS MSE (+ Blur + Only amp)	67.98	116.23	281.24
GS MSE (+ Blur + Median)	150.33	172.15	259.00
GS MSE (+ Blur + Only amp + Median)	120.62	147.83	240.24
Mean filter MSE	125.15	155.65	264.89
Median filter MSE	104.84	152.18	300.71
BayesShrink MSE	33.02	114.39	263.35
VisuShrink MSE	218.38	388.67	610.52
Origial MSE	44.04	263.35	953.55

Newcastle University

Final result (Colored SSIE)

Dataset	noisy10	noisy25	noisy50
COL SSIE	0.7887	0.6361	0.4424
COL SSIE (+ Only amp)	0.7887	0.6361	0.4424
COL SSIE (+ Median)	0.8167	0.6800	0.4929
COL SSIE (+ Only amp + Median)	0.8167	0.6800	0.4929
COL SSIE (+ Blur)	0.8077	0.7215	0.5652
COL SSIE (+ Blur + Only amp)	0.8934	0.7672	0.5724
COL SSIE (+ Blur + Median)	0.7880	0.7396	0.6309
COL SSIE (+ Blur + Only amp + Median)	0.8219	0.7631	0.6414
Mean filter SSIE	0.8240	0.7367	0.5791
Median filter SSIE	0.8167	0.6800	0.4929
BayesShrink SSIE	0.8580	0.6977	0.5633
VisuShrink SSIE	0.6246	0.5045	0.4361
Origial SSIE	0.7213	0.4130	0.2161

Newcastle University

Final result (Colored MSE)

Results produced are averaged over the whole dataset we have obtained

Dataset	noisy10	noisy25	noisy50
COL MSE	200.51	264.01	513.69
COL MSE (+ Only amp)	200.51	264.01	513.69
COL MSE (+ Median)	151.29	217.95	395.29
COL MSE (+ Only amp + Median)	151.29	217.95	395.29
COL MSE (+ Blur)	172.40	185.03	304.03
COL MSE (+ Blur + Only amp)	77.886	120.08	301.61
COL MSE (+ Blur + Median)	199.90	218.40	282.58
COL MSE (+ Blur + Only amp + Median)	168.57	193.11	395.29
Mean filter MSE	270.84	302.42	416.67
Median filter MSE	151.29	217.95	395.29
BayesShrink MSE	57.98	178.77	360.72
VisuShrink MSE	279.53	491.75	736.98
Origial MSE	97.280	580.35	2067.19

Overview

- A more general shape
- A method to predict the shape correctly, this was done on a simple grid search with few values and exceeding runtime ($\sim$ 1 minute to generate the filter per image)
- Problems with arbitary rotated images
- Preferential petals
- Highly detailed images

Newcastle University