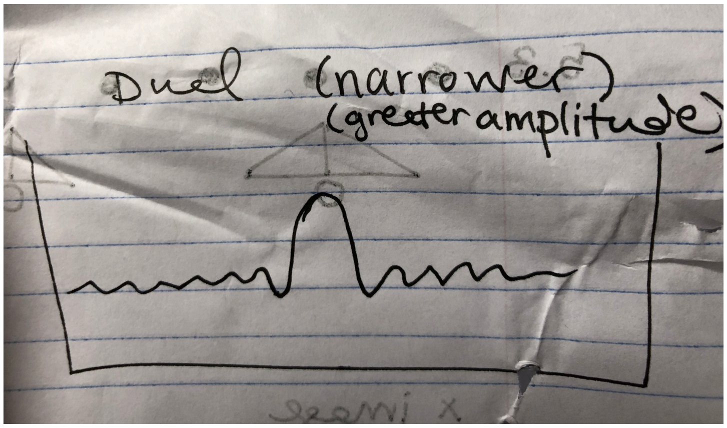
1.1

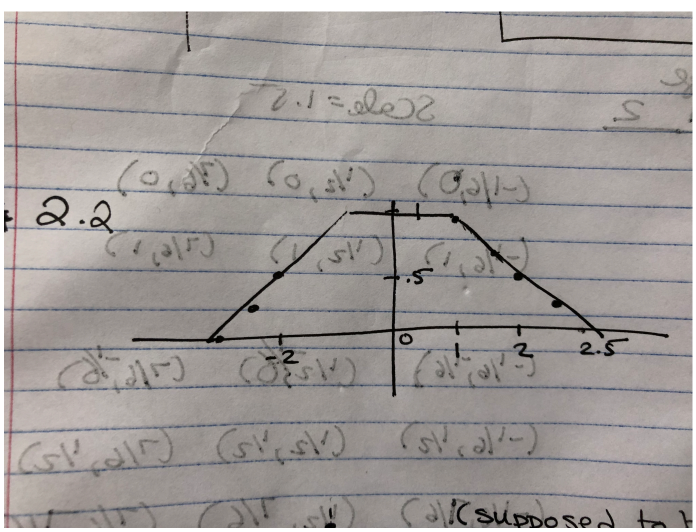


1.2 sinc is approximated with the Gaussian distribution or triangle distribution. In this case, we use finite extends and weights greater than or equal to zero. These distributions are easier to compute than sinc, which has an infinite support.

1.3 In Brush, a pixel was a colored square at a specific point on the canvas. Now, we see pixels as a sample of some function at a particular point (i,j). This function maps pixel coordinates to real numbers. In this perspective, pixels are point samples rather than squares, which are reconstructed using subpixels for display.

2.1 a) 0
b) 0.25
c) 0.75
d) 1

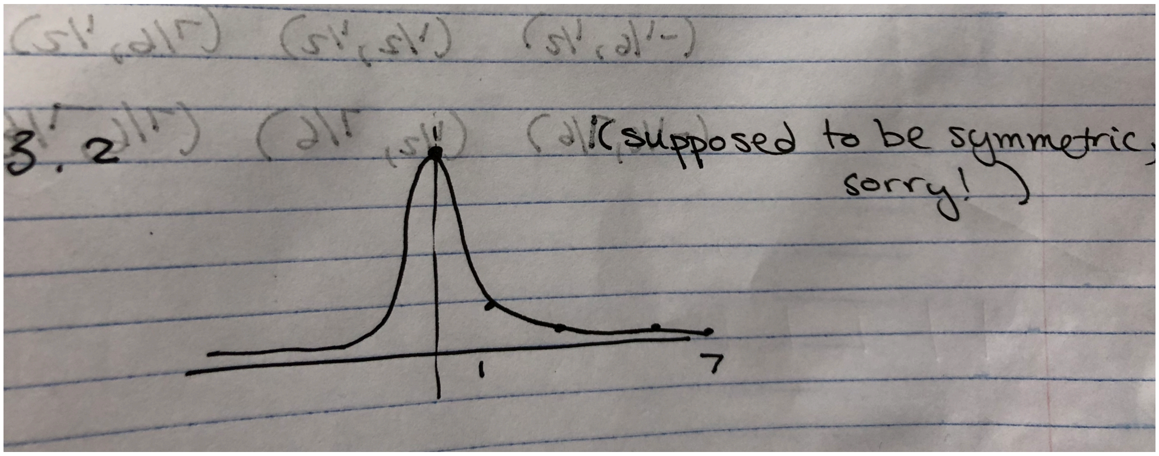
2.2



2.3 F(x) multiplied by G(x)

3.1 2 * n

3.2



4.1) The effect of an ideal blur is eliminating high frequencies. This is the same as multiplying by a box filter.
4.2) In the spatial domain, this is convolving with sinc(x), which is the same as sin(pi*x)/(pi*x)

4.3

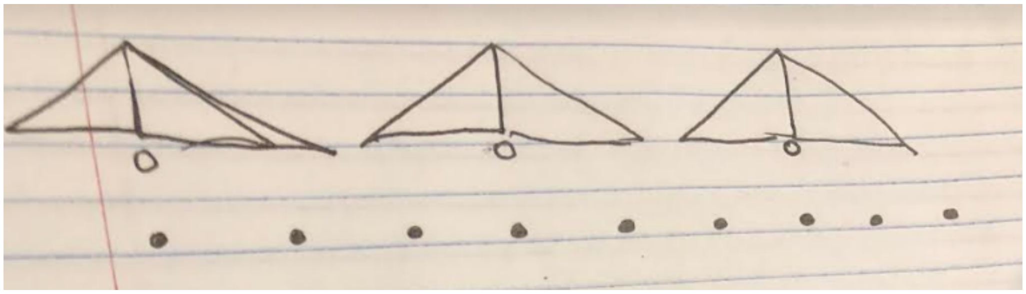
4.3

$$\sum_{-2 \leq W \leq 2} \frac{1}{5} I(y, x+\omega) = N(y, x)$$

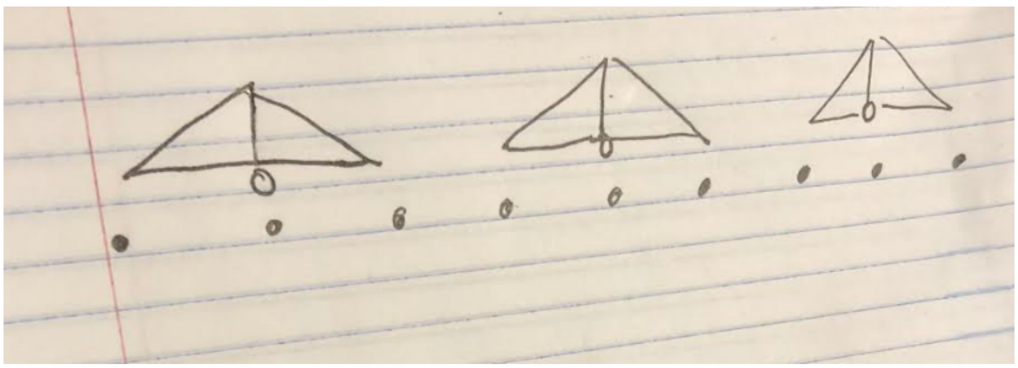
4.4 My idea is to pad the image with zeros on all sides; there should be the same amount of zeros on each side as the size of the filter. This way, “out of bounds” filters would just hit zeros rather than exceptions. To keep the image’s brightness constant, I would make sure that the filter was normalized so I’m not losing or gaining any information.

5.1 The filter support width is 2.
5.2 The filter support width is 2/n

5.3



5.4



5.5

(-1/6,-1/6), (1/2, -1/6), (7/6,-1/6), (-1/6,1/2), (1/2,1/2), (7/6,1/2), (-1/6,7/6), (1/2,7/6), (7/6,7/6)

5.6a) triangular
b) Gaussian
c) bad
d) linear