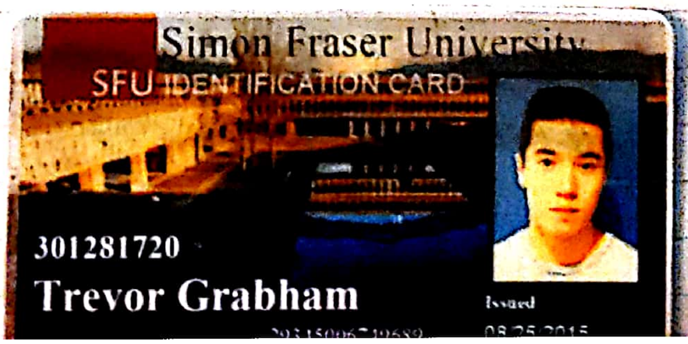
1.a) - load in the colour image
- create a second, grayscale copy
- convert both of these images to doubles.
- on the grayscale image, use canny edge detection to create a binary image of the edges
- dilate the binary image
- use an array of 3 of the dilated binary images as a mask on the original image and set all of those values to zero.
- the original image should now look like this

We need to convert the original image to a second grayscale copy to use canny edge detection properly. We dilate the image to get the wide strong edges. we use an array of 3 of the dilated binary images so that we change all of the RGB values, not just one.

1. b) Adjusting the algorithm to ensure that the binary mask that we use is changing all three of the RGB channels should fix the green patches

1.c) To create the output map of the noise, we could use a modified version of a median filter to move the centre pixel to the output array iff it is an outlier with respect to the local window.



for the child $\qquad\qquad\qquad\qquad$ for the tree

2.a) $y = 1m$ $\quad y' = \frac{y}{z} \cdot d$ $\quad y = 3m$ $\quad y' = \frac{y}{z} \cdot d$

$z = 4m$ $\quad 100px = \frac{1}{4} f$ $\quad z = 8m$ $\quad = \frac{3}{8} \cdot 400px$

$d = f$ $\quad\quad d = 400px$

$400px = f$ $\quad\quad = 150px$

2.b) $y_{tree} = 3m$ $\qquad y_{kid} = 1m$

$z_{tree} = z_{kid} + 4m$ $\qquad z_{kid} = ?$

$d = ?$ $\qquad d = ?$

$$y'_{tree} = \frac{3}{(z_{kid}+4)} \times d = 2y'_{kid}$$

$$= 2\left(\frac{y_{kid}}{z_{kid}} \times d\right)$$

$$\frac{3}{(z_{kid}+4)} \times \cancel{d} = 2\left(\frac{1}{z_{kid}} \times \cancel{d}\right)$$

$$\frac{3}{(z_{kid}+4)} = \frac{2}{z_{kid}}$$

$$3z_{kid} = 2z_{kid} + 8$$

$z_{kid} = 8m$ $\qquad$ c_2 should be 4m away from c_1

2.c) $y'_{kid} = \frac{y_{kid}}{z_{kid}} \times d$

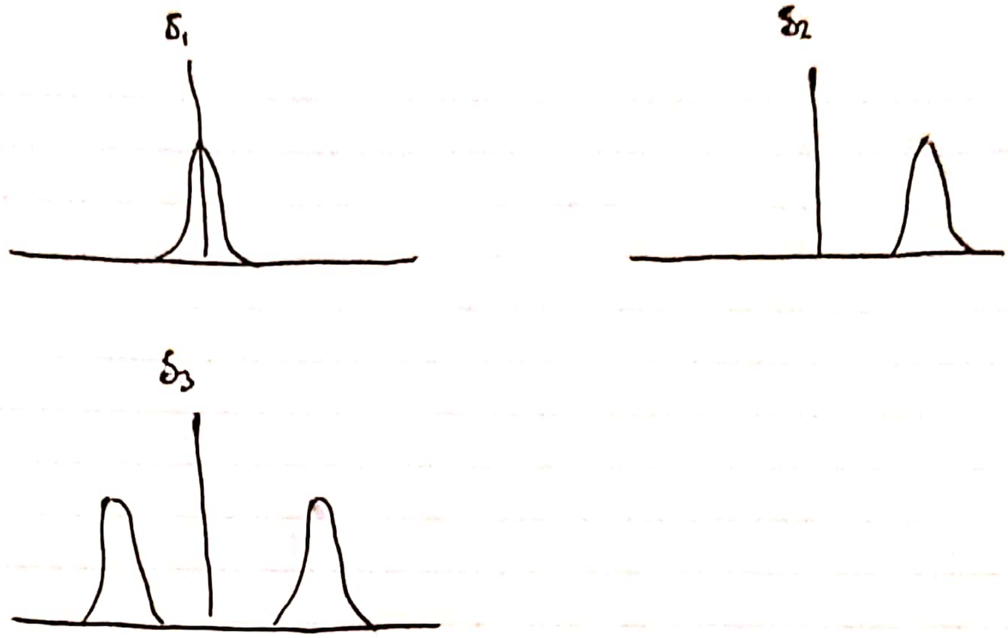
$100px = \frac{1}{8} \times d$

$800px = d$

$f = 400px$

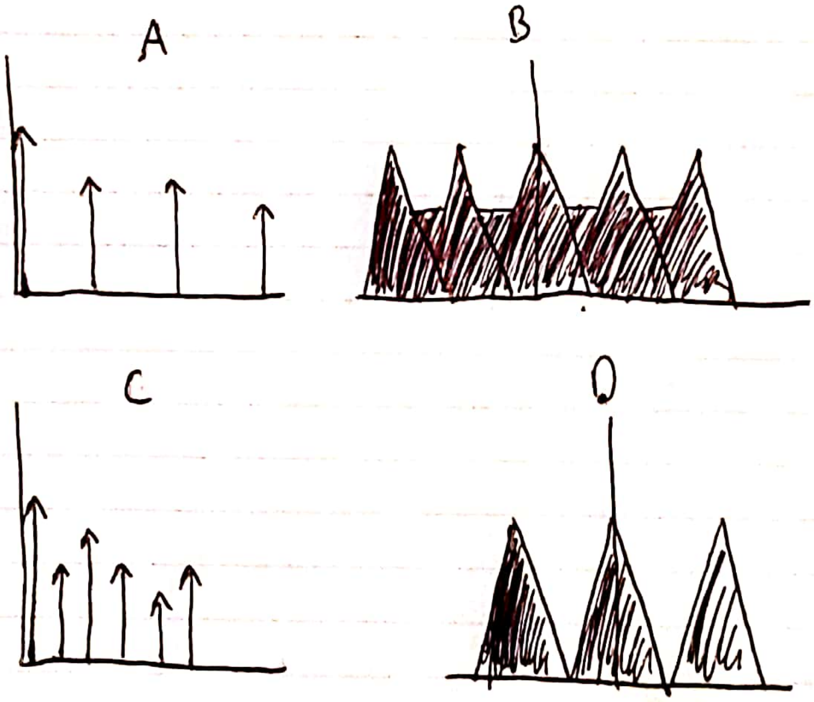
$d = 2f$

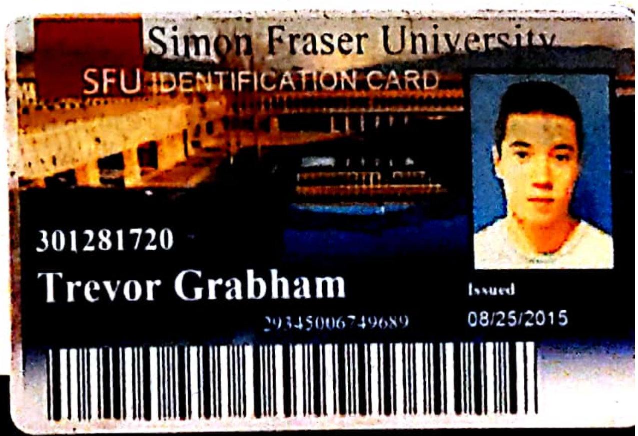
3.a)



3. b) convolution in the spatial domain is equivalent to multiplication in the frequency domain and vice versa.

3. c)





3. d) The two ways to get rid of aliasing are increasing our sampling rate like we did in graph C, or we use a low pass filter.