

**Introduction to Computer Vision**

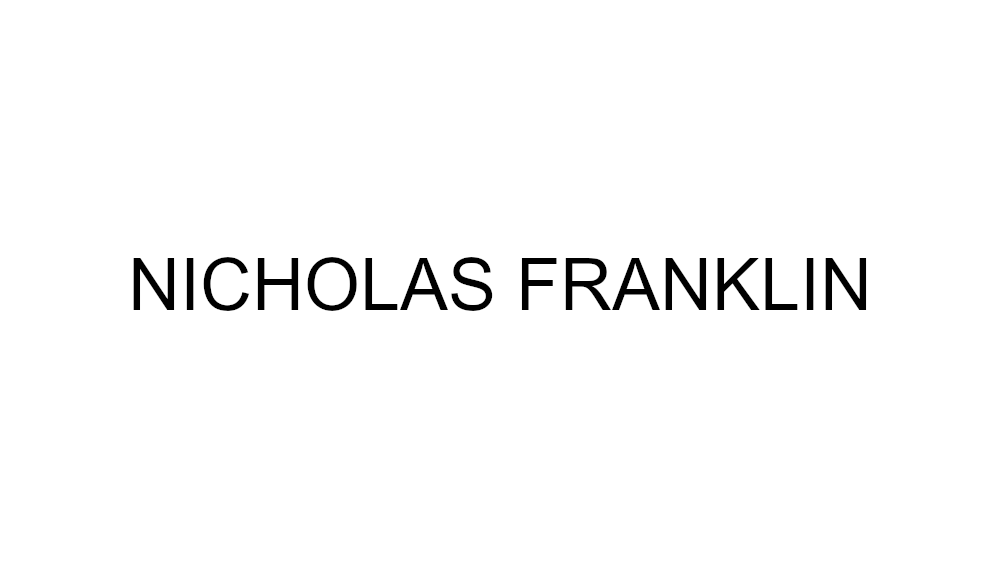
**Coursework**

**Submission 1**

**Your name Nicholas Franklin**

**Student number 150402149**

**Question 1(a):**



**Rotated images:**

θ = -50 deg

θ = 60 deg

θ = 30 deg

θ = 120 deg

**Skewed images:**

θ = 10 deg

θ = 40 deg

θ = 60 deg

**Your comments:**

**At first, I skew very much in the wrong dimentio. Resulting in a matrix that implied I had skew. Turns out this was actually that I was multiply the matrix in the wrong order. Resulting in a y skew instead of an x skew.**

**I was skewing with the bottom left pixel at 1,1 not 0,0**

**In addition, because matlab treats the top left as the origin. The skew matrix is the symtrical matrix of the standard bottom left origin based matrix.**

**New problem. When I use a negative value for the angel it errors out. As the position of some points is below 0.**

**Solved by sizing the image based on the difference in location of the max and min points in the image. Then shifting the image into positive space to display the image.**

**Because matlab matrix origin is in the top left then to get then to get a right shifting skew from the bottom left we need to**

**I had no problems with rotation until I got to gap filling. The main problem I had was differentiating between a hole and a piece of the extension to the image so the rotation is not cut off.**

**I made an array of -1’s with dimension large enough to fit the whole output image. Then I output the new pixels into their required positions. Next I loop thought each pixel in the image ignoring any pixels where they had 1 or less non -1 in there 4-neightborhood. This results in my images being very slightly enlarged on the edges.**

**My hole filling is failing to fill anything with white.**

**Learnt how cell arrays worked so that I could find overlayed pixels.**

**Question 1(b):**

θ2=50 and θ1=20 clockwise

θ1=20 clockwise and θ2=50

**Your comments:**

**Question 2(a)**:

**Designed kernel:**

**A box blur**

Averaged image



Original image

**Your comments:**

This box blur kernel averages the value of the target pixel by the pixels surrounding it. The matrix constant 1/9 reduces the floating maths to a multiplication with he output of the mask on a target pixel.

The pixel is added together by this mask and divided by 9 (the weight of the mask) so that the pixels are in the range 0-255.

I chose to convert the image to grey scale because of the comment in the instruction stating that the input array is normally a grey level image.

**Question 2(b):**

**Filtered image with kernel A**

**Filtered image with kernel B**

**Your comments:**

Kernel A is a gaussian blur approximation. It has blurred the high frequency noise from the image while reasonable painting the lower frequencies.

This can be seen in the number plate and the car manufacturer logon on the boot. The letters on the number plate are reasonably crisp because they are a large area of the same colour and gaussian blur removes high frequency noise. Whereas the logo is made up of a lot of pixel of extremely different values to there neighbours. Therefore, is similar to high frequency noise and this has been blur to an approximate single colour.

Kernel B is an edge detection kernel. It has made any vertical or horizontal edge in the image white while the rest is black.

Its design means that it responds to edges on the horizontal and the vertical. The total average output of the kernel is 0 hence most of the image being black as the kernel outputs 0 when all the pixels in the masked area are the same value. Where is if there is an edge there is a change in intensity which this filter response to creating bright areas.

The response on the vertical and horizontal is best seen on the 5 on the number place the top horizontal line is bright white as is the vertical down stroke.



The lack of response on the diagonals is seen on the bumper as is appears in the transformation image to be made up of several discounted white horizontal lines as the edge is slightly angled so only parts of the line are responding to the filter.

**Question 2(c):**

A followed by A

**A followed by B**

**B followed by A:**

**Your comments:**

i) Kernel A then kernel A further blurs the image compared to a single application of the blur. The glare on the too of the car shows this best as you see the circle of the glare growing compared to a single application of the blur and a further lose of detail in place like the car name which is no longer readable as ‘Quick’.

ii) Kernel A then kernel B. Blurs the original image then applies the edge detection filter. This results in the response on the edges being reduced. This can be seen by less intense lines on the edges. However, the edges detected are more continues comparted to just applying B.

iii) Kernel B then kernel A. Applies an edge detection then gaussian blurring to the image. This results in stronger responding points in the image compared to A then B (they are further to white). However, the lines in this image are often not continues. This is because the edge detection responses best to large changes in the intensity between the target pixel and the neighbourhood which are more common before the blur. In addition, the responses in B then A are thicker because the blur averages some of the adjacent pixel from black closer to the value of the detected edge.

**Question 2(d):**

**Extended kernels of A and B (5x5):**

**Results obtained by applying 5x5 kernel:**

**B followed by A**

**A followed by B**

**A followed by A**

**Extended kernels of A and B (7x7):**

**Results obtained by applying 7x7 kernel:**

**A followed by A**

**B followed by A**

**A followed by B**

**Your comments:**

b) I identified kernel A as a gaussian blur so my 5x5 kernel is what I remember a 5x5 gaussian blur to be. In additional a gaussian blur is approximately a blur curve so I model my 7x7 kernel A on a bell curve that can be seen to grow slowly near the edge growing faster as you get to the centre with a peak.

5x5

I created a 5x5 and 7x7 of kernel B by identifying that the average output of the kernel would be 0 so I decided to maintain that. In addition, we wanted the effect to be the same therefore we wanted it to continue to respond to horizontal and vertical lines only. So, when I extended the kernel, I increased the value the centre pixel when I added additional 1s on the horizontal and vertical axis of the mask. Therefore in all my kernel B extended masks, they result in a 0 if applied to an areas of the same colour.

The 5x5 mask extends response even more strongly to vertical and horizonalt edges.

**Question 3(a):**

**Two non-consecutive frames:**

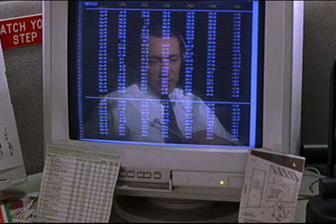
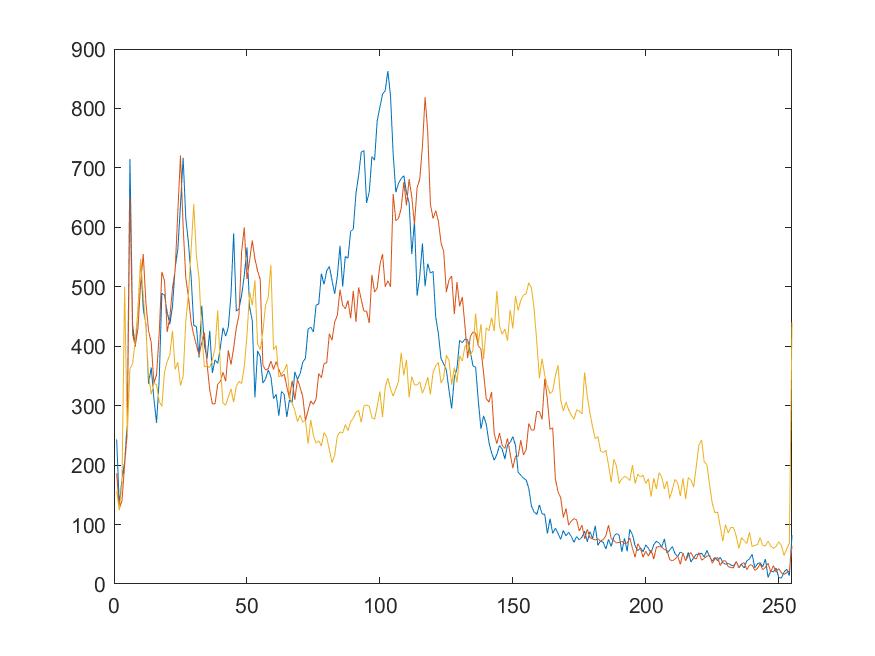
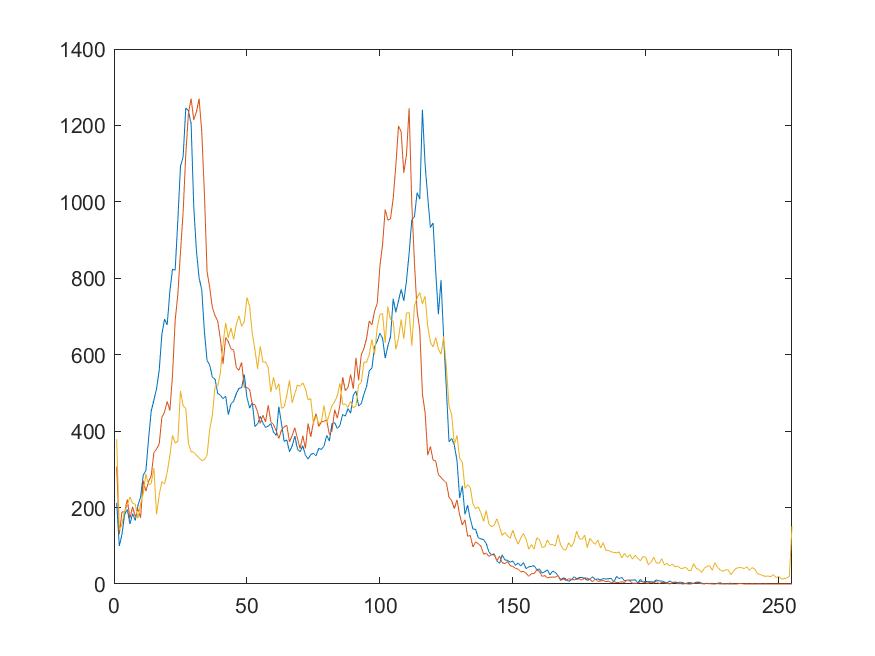
Image 1

Image 2

**Corresponding colour histograms:**

Histogram 2

Histogram 1

**Your comments:**

**Question 3(b):**

**Example 1:**

It

It+1

**Histograms:**

Histogram of It

Histogram of It+1

Intersection result

**Example 2:**

It+1

It

**Histograms:**

Histogram of It

Histogram of It+1

Intersection result

**Your Comments:**

I was unsure how to deal with the multiple channels, so I decided to find the minimum of each channel at its j histogram value.

I also choose to normalize the intersection value.

**Question 3(c):**

**Comments:**

**What does the intersection value represent?**

The intersection value represents the number of pixels

**Can you make decisions about where the scene in the video changes?**

**How robust is the histogram intersection technique to changes?**

**Where does it fail?**

**What would be other applications areas where histogram calculations and histogram intersection can be used?**