Queen Mary
University of London

Introduction to Computer Vision

Coursework

Submission 1

Your name Wei-Shiang Lian

Student number 170509682

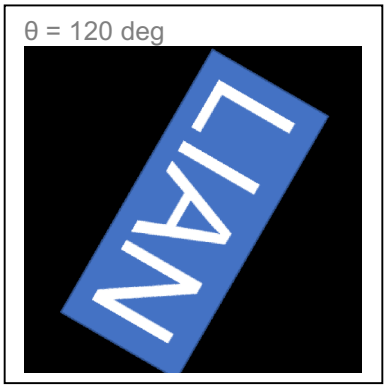
Question 1(a):

Rotated images:

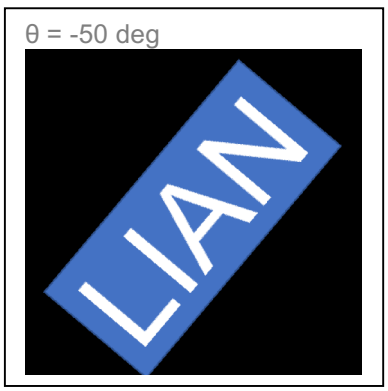
θ = 30 deg

θ = 60 deg

θ = 120 deg



θ = -50 deg



Fill the available spaces for your answers. Do not extend the spaces and do not change the formatting of the pages. Use the same font size and the same number of lines as in the original file.

Skewed images:

θ = 10 deg

θ = 40 deg

θ = 60 deg

Your comments:

Rotating Images:

According to the rotated images above, we can understand they are rotated clockwise from the horizontal axis (x-axis) with 30, 60, 120 and -50 degrees. And there were some problems occurred when I tried to rotate the images as below.

Firstly, if the images are rotated or skewed without expanding the images' size, the edges of the images will be cut because spatial transformations will change the domain of the images. To solve this problem, I expanded the images' heights and widths by their diagonals.

Secondly, the rotation centre of the image in Matlab is posed in the upper left corner. Before rotating the image, its rotation centre should be translated to the middle position of the images and moved to the original location after rotating.

Finally, there were some black bubbles on the rotated image at first, because its' size was bigger than original one, and it caused some spaces were not filled with pixels. However, I solved this problem by filling the similar pixel from the original image to the new image.

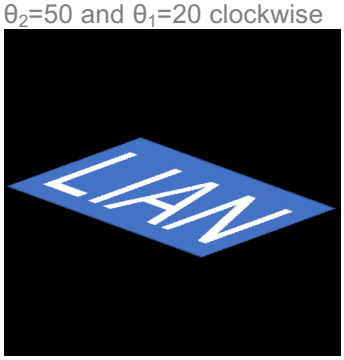
Skewing Images:

Skewed images are skewed horizontally from the vertical axis (y-axis) with 10, 40 and 60 degrees. And they have faced the same situation when skewing images without expanding the images' size, but it was easier to skew images without translation.

Question 1(b):

θ_1=20 clockwise and θ_2=50

θ_2=50 and θ_1=20 clockwise



Your comments:

Indeed, we can see different results with rotating and skewing images by exchanging their order. To begin with rotating and skewing subsequently, the image looks like a uniform diamond. Next, skewing and rotating sequentially, the image looks like a tilting parallelogram. There look totally different because skewing just shears the image along the horizontal axis and rotating changes the image's both of vertical and horizontal positions.

Question 2(a):

Designed kernel:

$$\textit{Kernel Average} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Original image

Averaged image

Your comments:

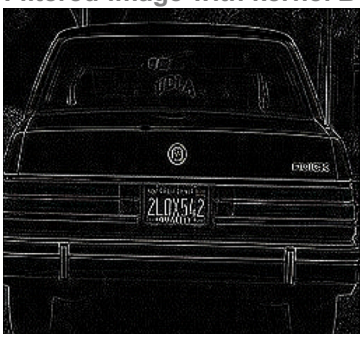
A convolution kernel that can average intensity value in a 3x3 region means each pixel's value should be added up with its original value and divided by the count of the kernel values. So, I designed a kernel that each element contains one to ensure every element in the image array can present the original state and divided by nine (the count of the kernel values) to prevent the pixel values overflow.

Furthermore, the effect of multiplying average kernel with the pixel array will blur the image, making it looks not so clear. If we implement average kernel with a larger size such as 5x5 and 7x7, the image will be more blurred due to including more pixel values to average.

Question 2(b):

Filtered image with kernel A

Filtered image with kernel B



Your comments:

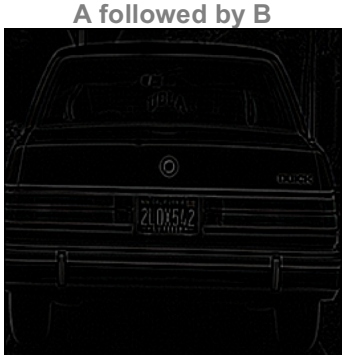
The effect of applying kernel A blurring the image looks like applying average kernel, but we can see the difference between the degrees of blurring. The effect of average kernel just like photographing an image by an out-of-focus lens and the visual effect of Kernel A makes the image resembled through a translucent screen. The Kernel A is also known as the operation of "Gaussian blur", used widely in graphics software, and the aim of it is to reduce image noise and detail.

The aim of applying kernel B is to emphasize the boundary of the objects in the image and the principle of this effect that is trying to identify points which its brightness changes sharply.

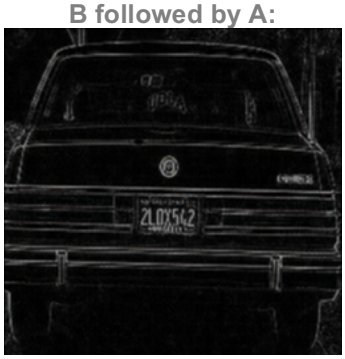
Question 2(c):

A followed by A

A followed by B



B followed by A:



Your comments:

These results altered by exchanging the order of applying filter functions. In the first case, the image applying Kernel A twice that looks smoother than applying Kernel A once. Moreover, the boundary of objects in the image applying Kernel A followed by applying Kernel B is not clearer than the image applied by Kernel B due to blurring effect from Kernel A. Finally, the image's edge is highlighted by applying Kernel B and then is smoothed by applying Kernel A so that the effect of edge detection is not sharp as the previous effect.

Question 2(d):

Extended kernels of A and B (5x5):

Kernal A 5x5

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Kernal B 5x5

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Results obtained by applying 5x5 kernel:

A followed by A

A followed by B

B followed by A

Extended kernels of A and B (7x7):

Kernal A 7x7

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 4 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

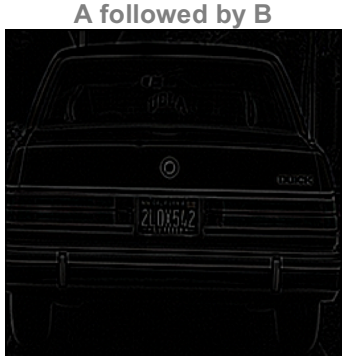
Kernal B 7x7

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

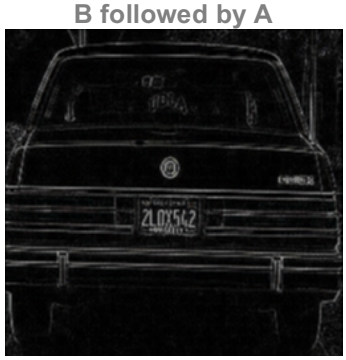
Results obtained by applying 7x7 kernel:

A followed by A

A followed by B



B followed by A



Your comments:

To keep the same effects with the original kernels, what I find out is expanding the kernel with empty pad such as filling zeros to the additional spaces in the new larger filter kernels 5x5 and 7x7. Because I have to keep the same product of the pixel values multiplying with kernels, I need to avoid any effects increasing the value due to a larger region.

Comparing the results obtained by applying 5x5 and 7x7 kernels, it looks all the same from the original kernel 3x3, and that is the aim what we should implement it. However, we may face some problems that larger size kernel needs more border space from an image, and there is not enough specific data to calculate a correct value and what we should do is filling the image pad with zero to make sure that every pixel can be multiplied by the larger kernel.

Question 3(a):

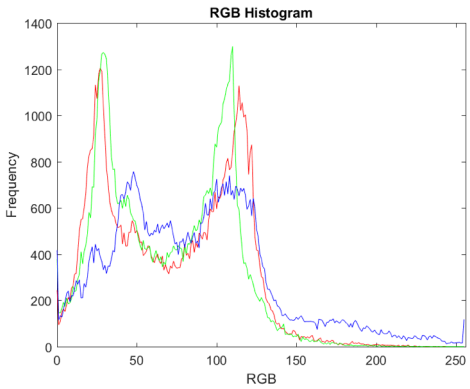
Two non-consecutive frames:

Image 1

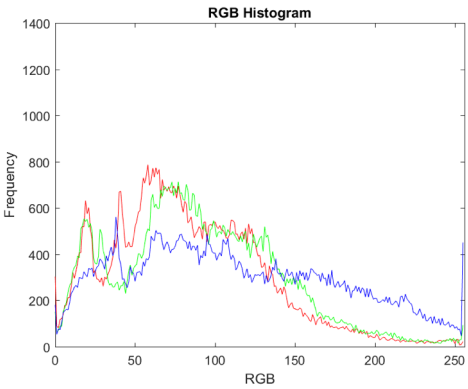
Image 2

Corresponding colour histograms:

Histogram 1



Histogram 2



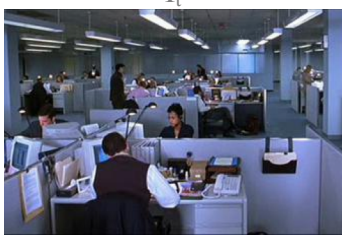
Your comments:

The video can be regarded as that is composed of many frames of an image, and every image extracted from it can constructs a RGB histogram by their RGB channel values. So, we have a series of RGB histograms from consecutive frames, and they can be analysed for different ways. For example, we can judge every frame is in the same scene or not by its RGB histogram curve shape is same with the others. Furthermore, it is obvious to see a different histogram of RGB channel value distributions from different images from the video.

Question 3(b):

Example 1:

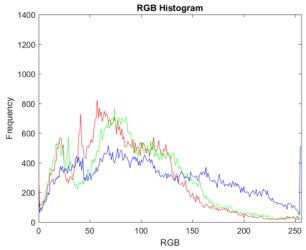
I_t



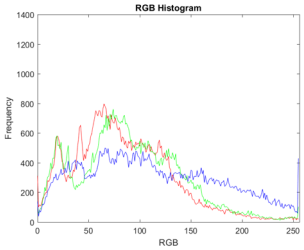
I_{t+1}

Histograms:

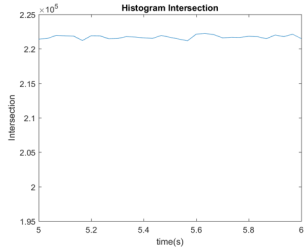
Histogram of I_t



Histogram of I_{t+1}



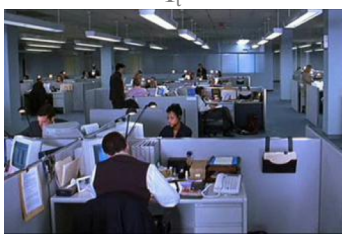
Intersection result



Fluctuating between 221200 and 222200

Example 2:

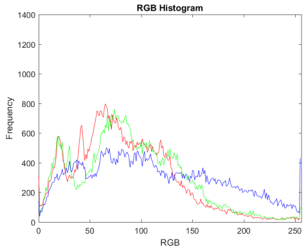
I_t



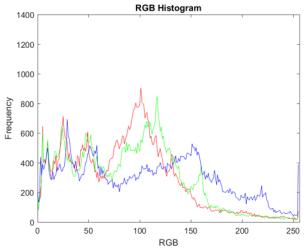
I_{t+1}

Histograms:

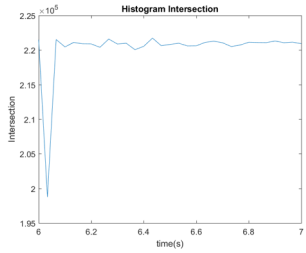
Histogram of I_t



Histogram of I_{t+1}



Intersection result



A sharp fall from
220000 to 199000

Your Comments:

For calculating histogram intersection, I set bin locations to 256 groups from 0 to 255 and add up three channel values. Moreover, I extract two sets of images from the given video sequence and generate their histograms and intersection results. One of them is from 5 seconds to 6 seconds by 30 frames per second; its images are quite similar so that the histograms look alike. The other one is from 6 seconds to 7 seconds by 30 frames per second; its images are different so that the histograms have a significant difference.

Furthermore, I can discriminate the intersection results between the different range what they change, and the first one's values are fluctuating from 221200 and 222200, its shape looks shifting smoothly; the other one's shape has a deeply decrease between 6 seconds and 6.1 seconds, and its values decline below 200000 and rise over 220000 suddenly.

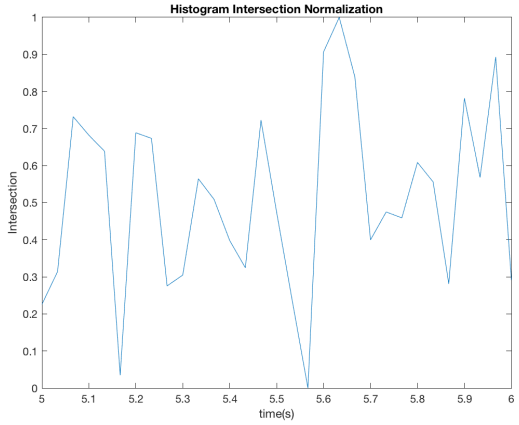
According to the results what I observe, it can be surmised that if intersection values have a big change comparing to the others, the video could have a scene translation.

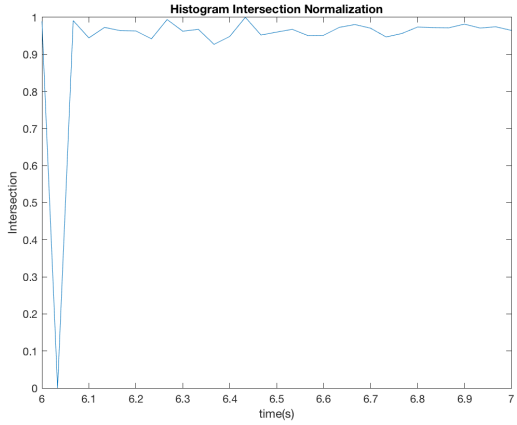
I try to normalize the intersection by calculating each intersection value subtracts the minimum of the intersection values and divide the distance between the minimum of the intersection values and the maximum of intersection. It can be displayed as formulas as follows:

$$\mathbf{I}(\boldsymbol{h}) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{min}(\boldsymbol{I_j}, \boldsymbol{M_j})$$

$$\mathbf{N}(\mathbf{I}) = \frac{\boldsymbol{I}(\boldsymbol{h}) - \boldsymbol{min}(\boldsymbol{I}(\boldsymbol{h}))}{\boldsymbol{max}\big(\boldsymbol{I}(\boldsymbol{h})\big) - \boldsymbol{min}(\boldsymbol{I}(\boldsymbol{h}))}$$

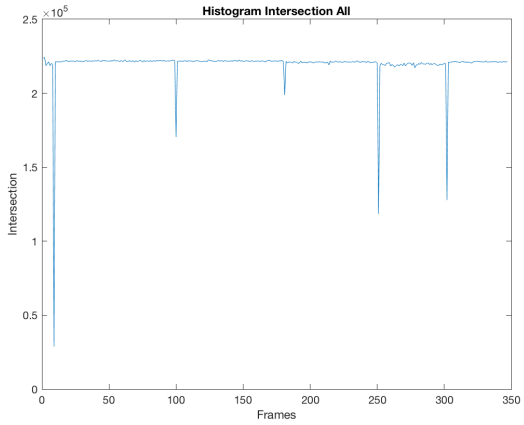
Therefore, the y-axis range of the intersection results is from 0 to 1, and the first result can't recognize scene translation easily after normalizing. However, the second result keeps its original figure that can recognize scene translation easily due to a sharp fall.

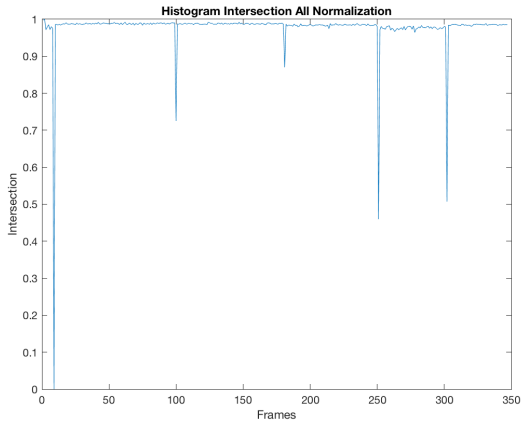




Question 3(c):

Comments:





Theoretically, the intersection values represent an overview of the frame images' colour and it can be regard as the way to find out if frame images' colour are similar. As mentioned previously, it can be distinguished that the scene in the video changes from the intersection values changes. For example, we can see that a few deeply fall in the histogram intersection plot so that we can make a decision about when the scene in the video changes. Therefore, it can be a strong way to judge when the scene changes. However, it may happen some situation that different images have the same histogram intersection, and it should be aware even its probability of occurrence is not too high.