Image Processing Coursework

Part A – Bilateral Filter

Question 1

The bilateral filter is a non-linear, edge-preserving filter for images. It is a spatial filtering approach that is an extension of a Gaussian filter that removes noise from images. An (n ***x*** n) neighbourhood Ω is chosen where n is odd so that the neighbourhood has a clearly defined centre pixel. The Gaussian function (detailed below) is performed twice, with two inputs described below, on each pixel.

The bilateral filter is different to the Gaussian filter in that the similarity between two pixels is considered as well as the spatial proximity between them. The similarity between two pixels is their intensity difference (where a smaller intensity difference means a higher influence on the new value of the pixel), and the spatial proximity is the distance between them (where a smaller distance difference means a higher influence on the new value of the pixel). A Gaussian function is used on these values, which is shown to the right.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

The sigmas (σ) used in the Gaussian function are called sigma colour and sigma space for intensity and distance respectively. Increasing each sigma will increase the effect of the bilateral filter on the image. The sigmas are normally the same value in practice.

The value ***x*** is given by one of the two differences used to calculate the new pixel's value in the bilateral filter. The first is the distance difference (spatial proximity) |p – p'|, which is the distance of pixel p' from the pixel p at the centre of the neighbourhood. The second value ***x*** can take is the intensity difference between the two pixels: $I_p - I_p'$. It does not matter what sign this value takes as the Gaussian function will return the same value either way.

The actual filter is defined in the formula to the right. The denominator of the expression is used to normalise the result which ensures that the filter preserves image energy. The functions g_1 and g_2 are the same Gaussian function but with different inputs, and after the summation the numerator is multiplied by the intensity of the middle pixel of the neighbourhood. For greyscale images, the filter is performed on one of the RGB intensities (since each one is the same), but for colour images the filter is performed on each colour channel separately with the same parameters.

$$I_p^{output} = \frac{\sum_{p' \in \Omega} g_1(|p - p'|)\, g_2(I_p - I_{p'})\, I_{p'}}{\sum_{p' \in \Omega} g_1(|p - p'|)\, g_2(I_p - I_{p'})}$$

The bilateral filter is a great way to remove noise from images while maintaining detail and edge sharpness, and it is because of these properties that it is used in movie restoration where for example it could be used to improve the quality of a CCTV image of a suspect.

Question 2

The bilateral filter takes four inputs: the image to be filtered, the neighbourhood size, the standard deviation of the first Gaussian function (sigma colour), and the standard deviation of the second Gaussian function (sigma space). The challenge in fine tuning the parameters is that we want to get a result where there is no noise left in the image with as little detail removed as possible, and the results I achieved are discussed below.

Question 3

Firstly, I experimented with the neighbourhood size keeping sigma colour and sigma space equal. I found that for small values of the sigmas, the bilateral filter did not affect the image very much for small neighbourhoods. The images below are before and after I applied the bilateral filter with d = 3 and sigmas = 20.


Figure 1a


Figure 1b

They appear to be almost identical. The effect increased when I increased the neighbourhood size to d = 9, keeping the sigmas equal at low values.

We can see that there is a much more noticeable difference when the neighbourhood size is increased as in these images (below on the right) where d = 9. The filtered image is smoother and has lost detail (the forehead has fewer wrinkles and there are fewer single hairs near the neck area), and has lost some of its edge sharpness, though this is not noticeable at a glance. As the neighbourhood size increases, so does the loss of detail and the smoothing of the filtered image. However, the greater the neighbourhood size, the longer it takes to run the bilateral filter so there needs to be a trade-off.


Figure 2a


Figure 2b

Now, increasing the sigmas while keeping the neighbourhood size constant yielded the results below on the left, with values of d = 5 and sigmas = 20.


Figure 3a


Figure 3b

As we can clearly see, the filtered image is again nearly identical to the original, apart from a few forehead wrinkles removed from the filtered image. If we increase both sigmas significantly, we can see a much greater difference, as shown below with values of d = 5 and sigmas = 10000.

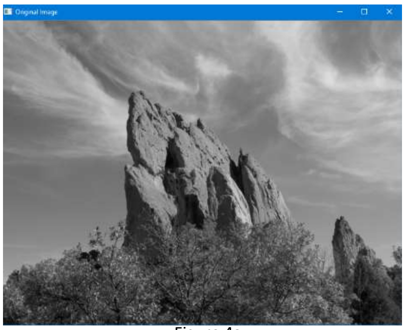
Figure 4a

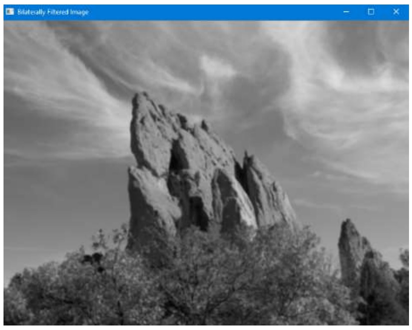
Figure 4b

We finally have a noticeable difference. The filtered image is much more blurred than the original and further increasing of the sigma values would have a smaller and smaller effect, since the correlation is logarithmic. We can conclude from these results that increasing the neighbourhood size along with the sigmas would further blur the image. We can infer from this that increasing the sigmas produces a more "cartoonish" effect.

Part B – Joint Bilateral Filter

Question 1

The joint bilateral filter is an extension of the bilateral filter. It takes an input of two images, where one image is a no-flash version of the other image. The second image is taken using flash photography. The only difference between the two filters is that the joint bilateral filter uses the difference in the flashed image's intensities, instead of the normal no-flash image's difference in intensities. This causes the last part of the formula to change from $\frac{g_2(I_p - I_{p'})}{)\, g_2(I_p - I_{p'})}$ to $\frac{g_r(F_p - F_{p'})}{g_r(F_p - F_{p'})}$ where g_r is the same Gaussian function and $F_p - F_p'$ is the difference in the pixels' intensities in the flash image. The numerator is still multiplied by the no-flash image neighbour's intensity.

The goal in photography is to reproduce the richness of the environment where the photo was taken. When there is little light in the environment, the photo appears unlike the environment, and it is difficult to see the details of the image as they are darkened. Flash photography solves this problem by adding artificial light to nearby objects in environments, so that cameras can use shorter exposure times and smaller aperture widths to capture enough light to produce flashed images. These images are more illuminated, their sharpness is increased, and noise is removed. However, flash photos have drawbacks such as disproportioned brightness of objects near the camera and red eyes in people. The joint bilateral filter uses the advantages of both the no-flash and flash images to produce a clear noise-free no-flash image. $g_r(F_p - F_p')$ accounts from removing the noise as there is no noise in the flashed image and multiplying by the no-flash image neighbour's intensity results in the no-flash image being recreated with more detail and less noise.

There are many applications of the joint bilateral filter, as it is a very effective filter. One example is fog removal from images – the filter will remove most of the fog from an image so that it is easier to see the surroundings. This works because the flashed image has higher visibility through the fog. Another example is with cameras in medical imaging where it can be used to sharpen the quality of images to see organs or internal cell structures more clearly to derive an accurate diagnosis. This works because the filter, when used with efficient parameters, removes noise from the images and sharpens the finer details. A final example is in visible or infrared filtering.

Question 2

Implemented in the code attached.

Question 3

Testing the joint bilateral filter on the test images yielded some interesting results. I used the no-flash and the flash images and began by making d constant and increasing the values of sigmas together equally. For example, the images on the next page (left) were achieved with values of d = 5 and sigmas = 5.

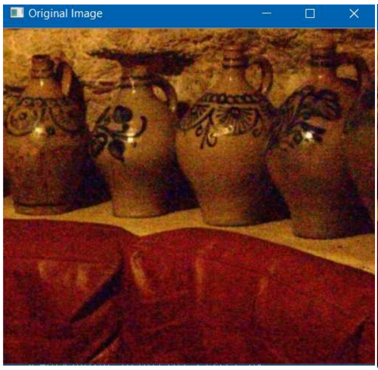

Figure 5a

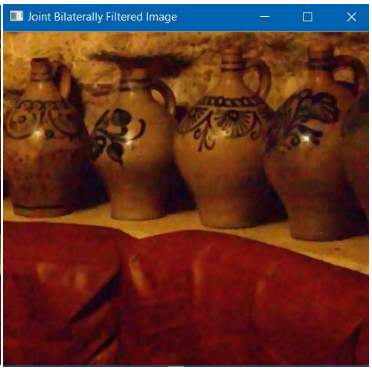

Figure 5b

As we can see, there appears to be a small effect of the filter on these images. There is less noise and the image returned is a no-flash image.

If we increase the sigma values to 20, then we get this:

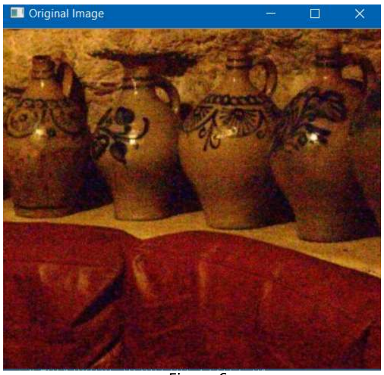

Figure 6a

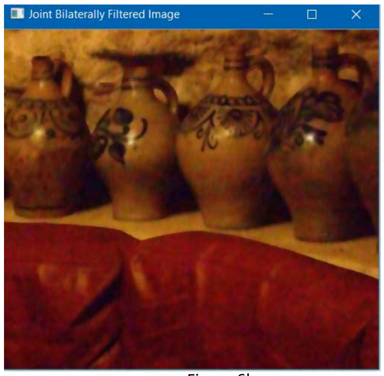

Figure 6b

There appears to be more blurring of the image than before, and we find that this in fact is the case for increasing the sigmas further while keeping d constant. Since we do not want too much blurring, we will experiment further with various values of d while keeping the sigmas constant at 5. The first value I tried with this for d was 7, and the results are shown below:

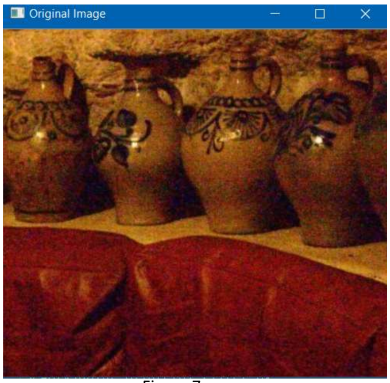

Figure 7a

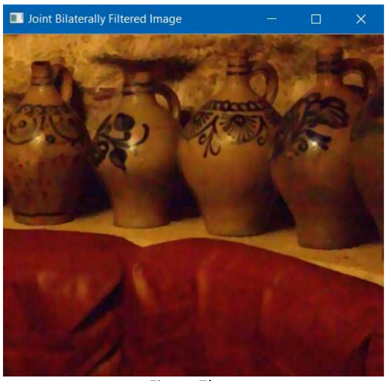

Figure 7b

Now we get a much smoother result, with less noise. It seems that the further we increase d, the more noise we remove from the image, so I tried again with d = 13:

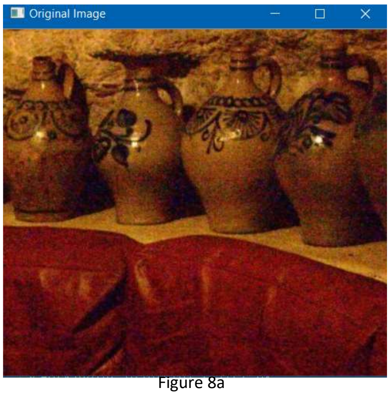

Figure 8a

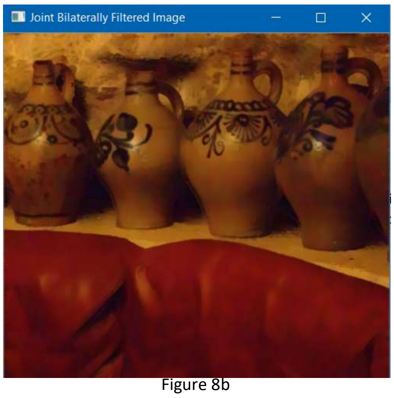

Figure 8b

The difference between d = 7 and d = 13 is noticeable. The latter is clearer and has less noise. It follows that we would want to use higher values of d and lower values of sigma when using the joint bilateral filter.

References

The Guided Bilateral Filter - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7004807

Bilateral Filtering for Grey and Colour Images - *https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=710815*

Digital Photography with Flash and No-Flash Image Pairs - http://hhoppe.com/flash.pdf

Flash Photography Enhancement via Intrinsic Relighting - https://people.csail.mit.edu/fredo/PUBLI/flash/flash.pdf

https://en.wikipedia.org/wiki/Bilateral_filter

https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html

Appendix

Images from report (**Part A**) in larger sizes:

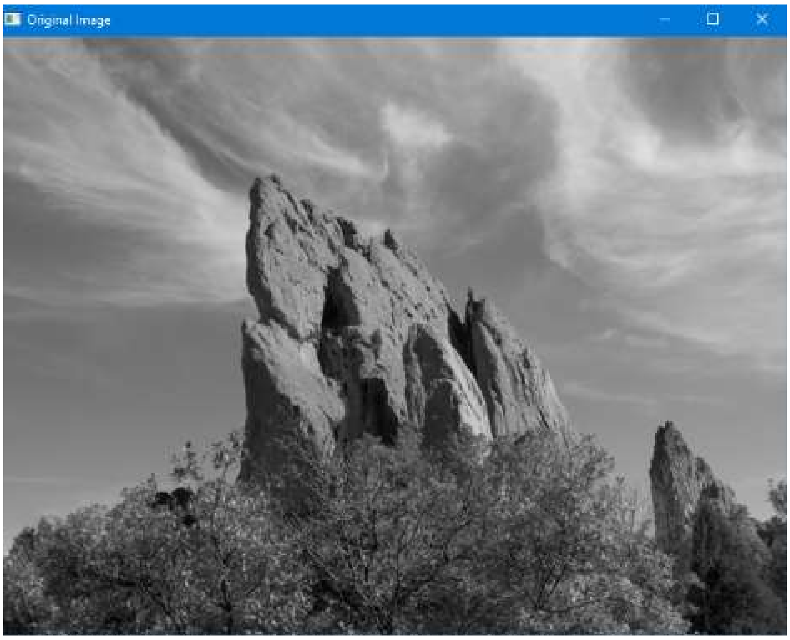

Figure 1a

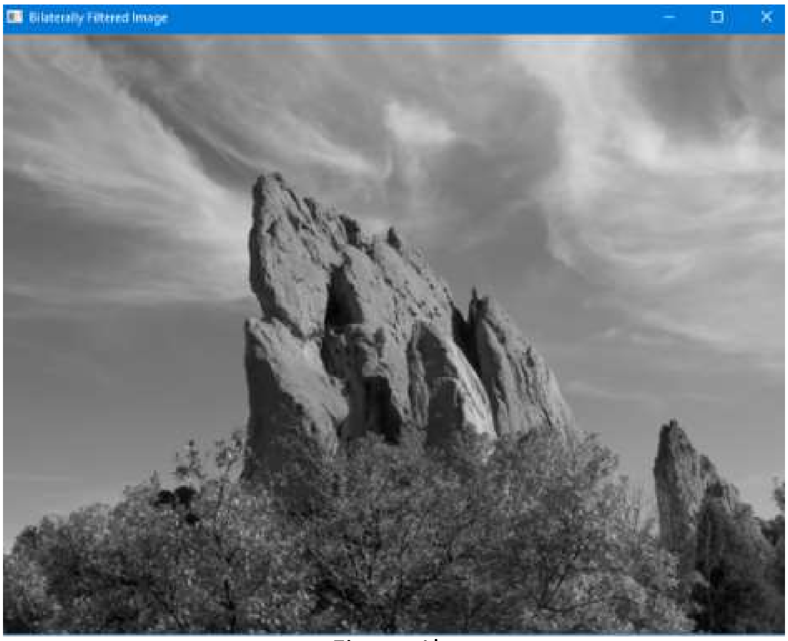

Figure 1b

Figure 2a

Figure 2b

Figure 3a

Figure 3b

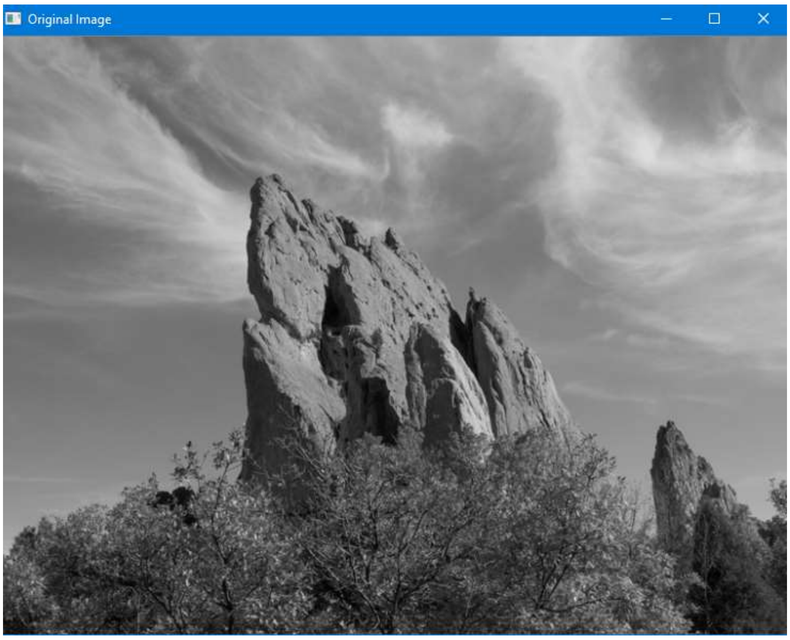

Figure 4a

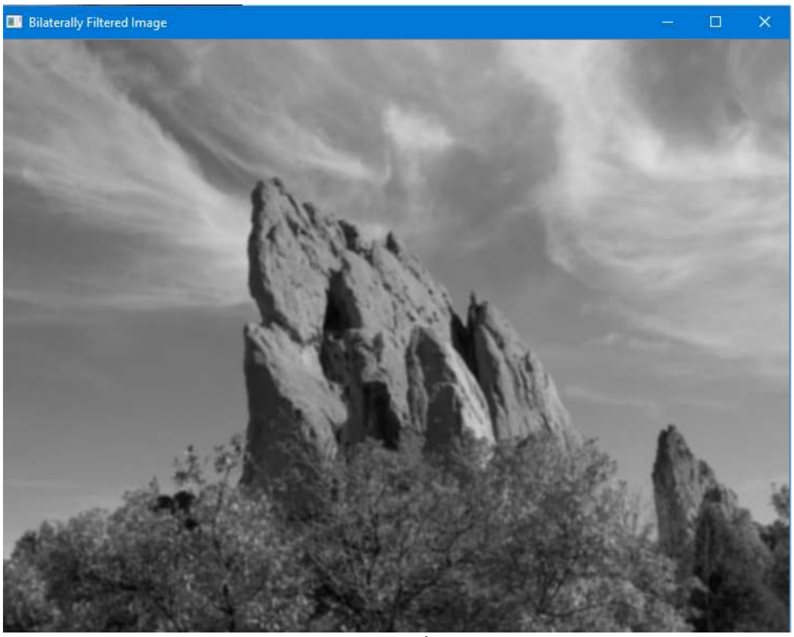

Figure 4b

Images from report (**Part B**) in larger sizes:

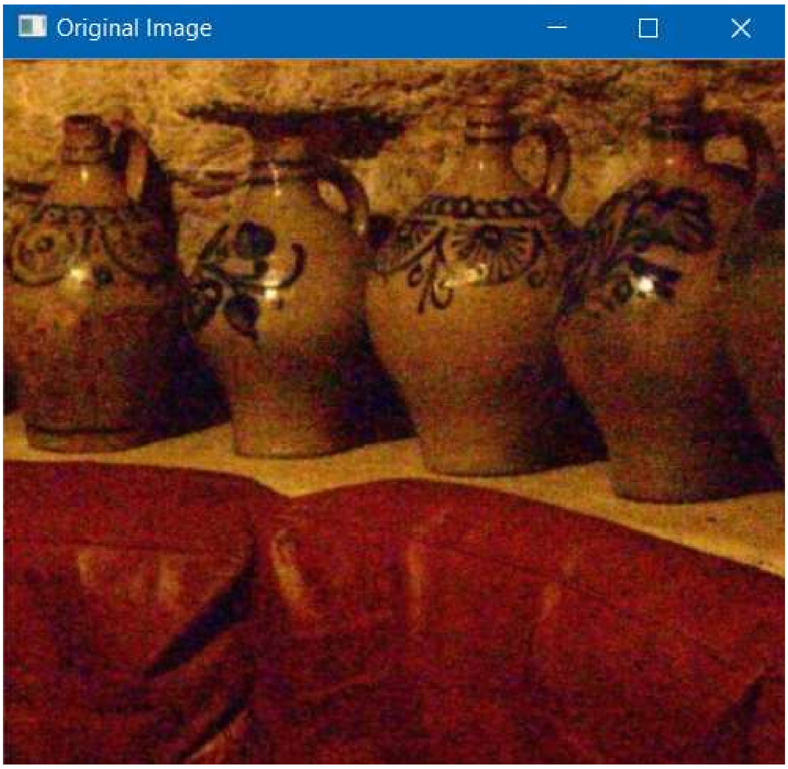

Figure 5a

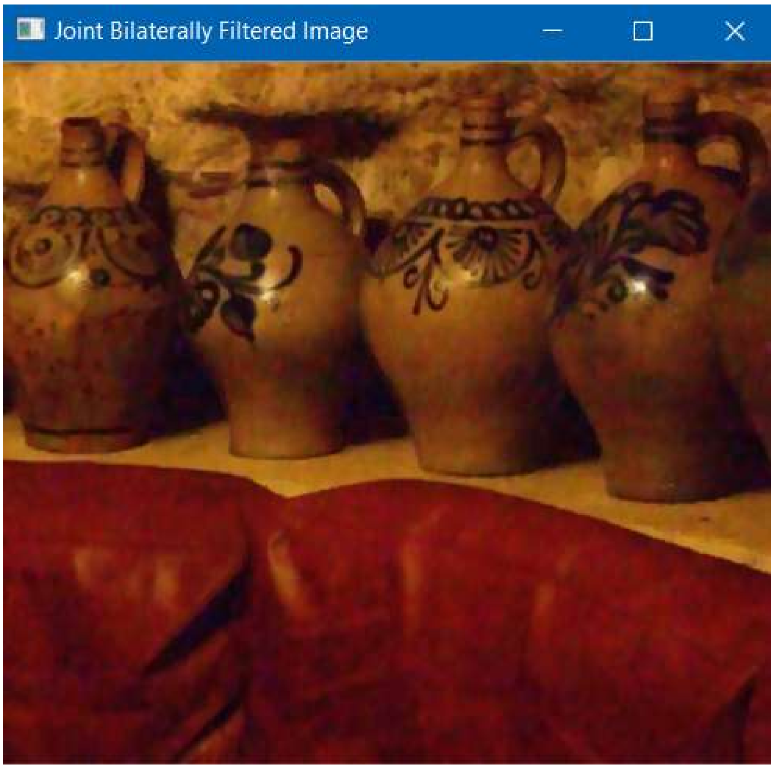

Figure 5b

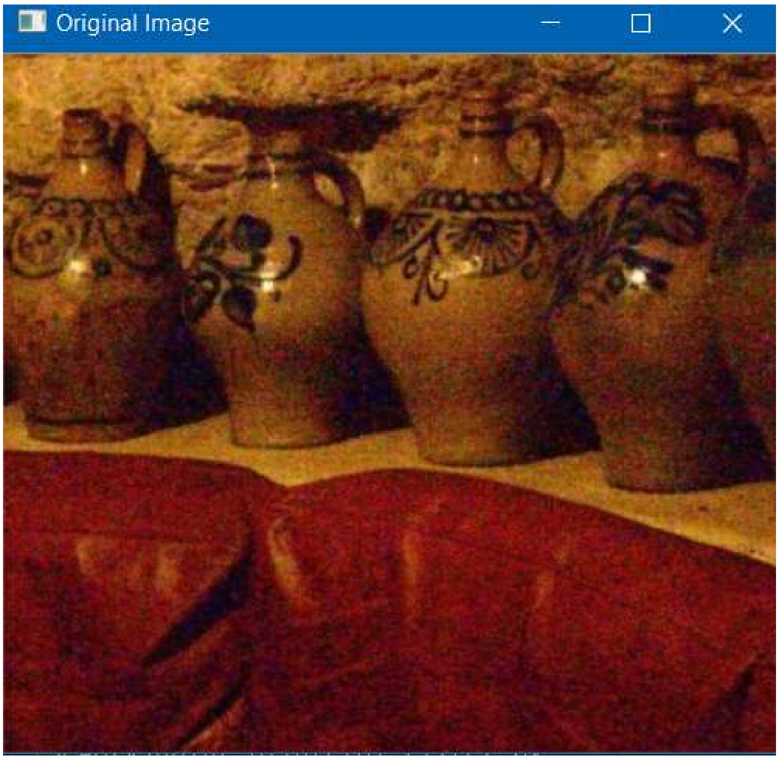

Figure 6a

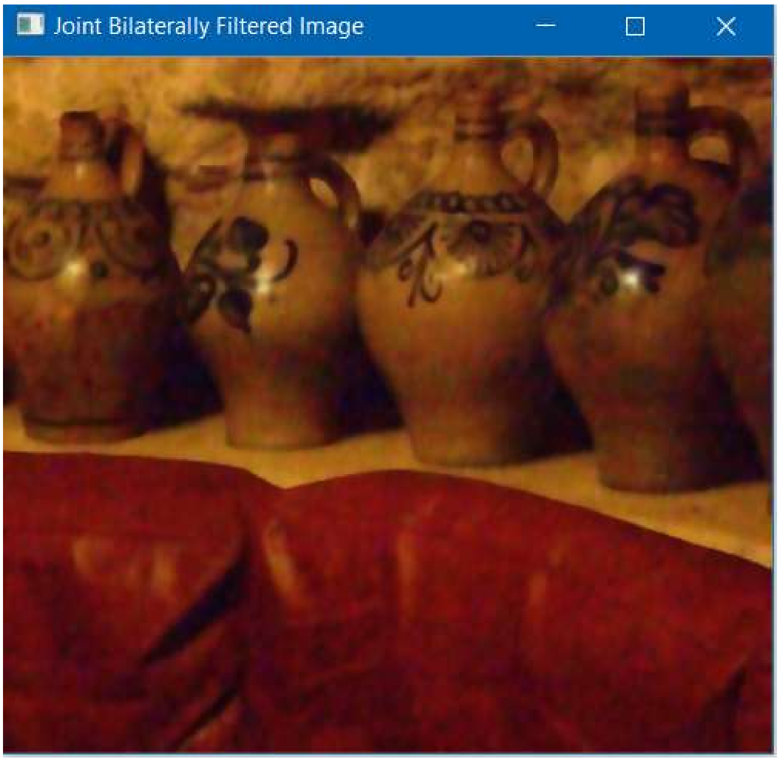

Figure 6b

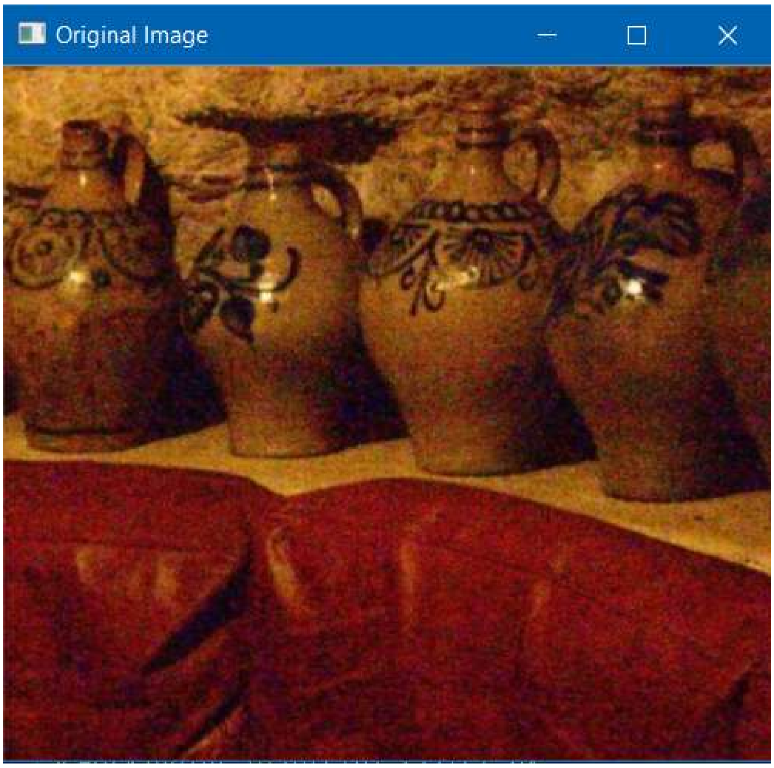

Figure 7a

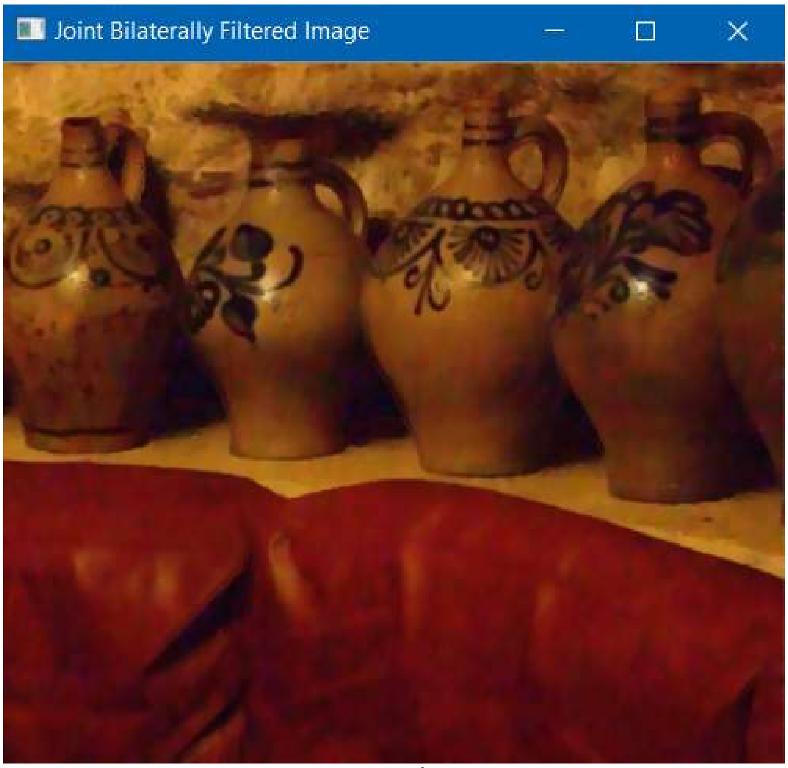

Figure 7b

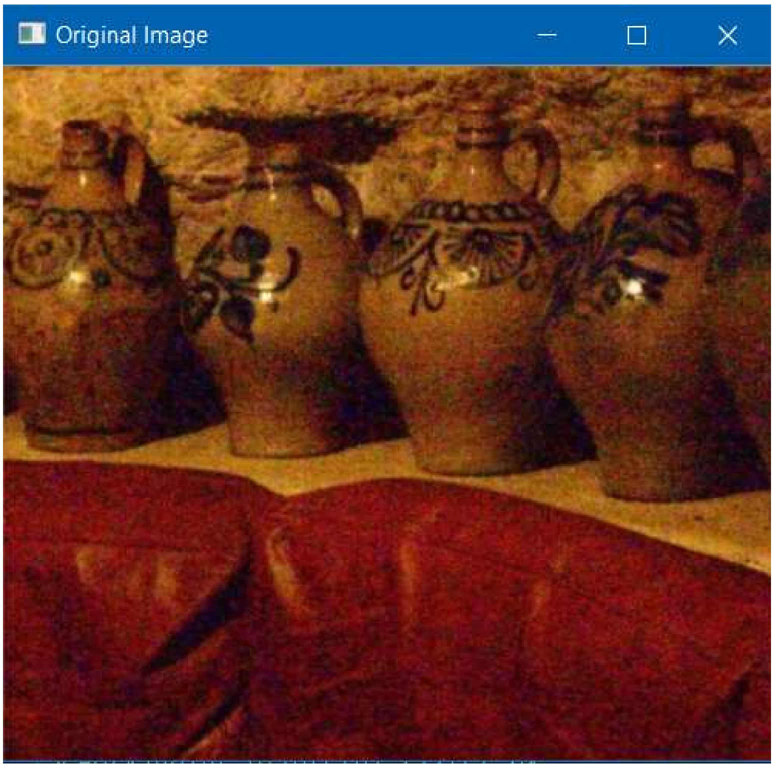

Figure 8a

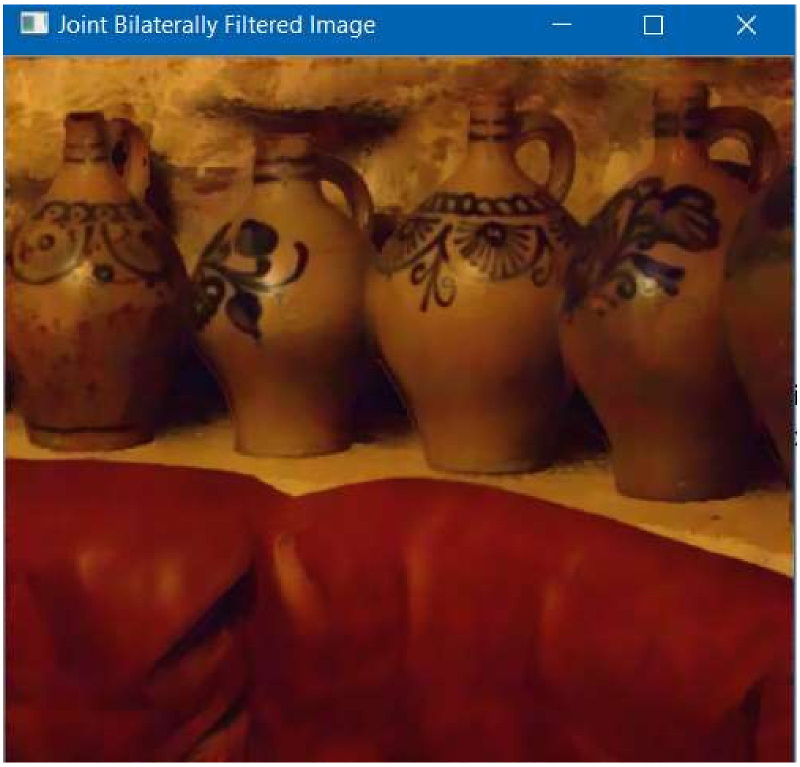

Figure 8b