Bilateral Filter – Digital Imaging Summative Assignment

This assignment implemented a bilateral filter for both grayscale and colour images in Python.

1.0 Introduction

Unlike traditional filters, bilateral filters focus on both spatial proximity and intensity similarity, that is, the influence of one pixel over another in a bilateral mask is determined by not only their closeness in spatial location, but also their differentness in intensity value. The bilateral filtering with a Gaussian kernel is described by equation (Ivrissimtzis, 2019) below

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

As the introducing of range domain, bilateral filters are able to preserve the edges of objects while smoothing the surfaces and reducing the noise: Pixels around edges diverse significantly in intensity and therefore influence each other much less during filtering, therefore preserving edges from being smoothed.

However, the advantage on edge preserving brings a disadvantage on filtering high-frequency components (where intensity changes rapidly on a short distance scale) because bilateral filters preserve too much high-frequency information such as edges and "salt and pepper noise" (Rasti, 2011). As a result, the main application of bilateral filtering acts on low-frequency denoising such as tone management, data fusion and 3D fairing (Paris et al., 2009).

2.0 Experiment on Grayscale Images

For grayscale images, the implementation of bilateral filtering is quite similar to traditional Gaussian filtering, except the introducing of range domain: Intensities of pixels are scalar value, the difference of which can be simply calculated by subtraction.

Meanwhile, a padding strategy named replication by extrapolating values from nearest pixels of the source image into enlarged area is utilized in this assignment.

All outputs of experiment on grayscale images can be found in **Appendix A**. The experiment uses a combination of proximity (σ_s) and similarity (σ_i) standard deviation as parameters of Gaussian function to test how these parameters control the smoothing effects.

(a) $\sigma_i = 1, \sigma_s = 1$ (b) $\sigma_i = 1, \sigma_s = 1000$ (c) $\sigma_i = 1000, \sigma_s = 15$ (d) $\sigma_i = 15, \sigma_s = 1000$

Figure 2.1 Parts of Filtered Grayscale Images

To illustrate the functionality of standard deviation parameters, detailed parts in **Figure 2.1** above are cropped from the output images. Comparing **(c)** and **(d)**, it is obvious that σ_s will enlarge the influence of one pixel over its spatial neighbourhood, causing a universal blurring across the image **(d)** similar to a traditional Gaussian filtering, while σ_i controls the influence of a pixel over pixels with close intensity, and therefore the introducing of range domain smooths surface significantly but still preserves edges in **(c)**. Similarly, in image **(b)**, a large σ_i with a small σ_s does nearly no blurring because σ_i only makes pixels with similar intensity more similar, which causes no obvious effect after filtering.

3.0 Experiment on Colour Images

Another advantage of bilateral filtering is its high applicability on colour images. The implementation for colour images only differs from which for grayscale images in that pixel intensities in colour images are vectors instead of scalar values, and therefore the similarity between them should be calculated as their Euclidean distance, a.k.a. colour space distance. Spontaneously, it is critical to choose the most meaningful colour space in colour image bilateral filtering.

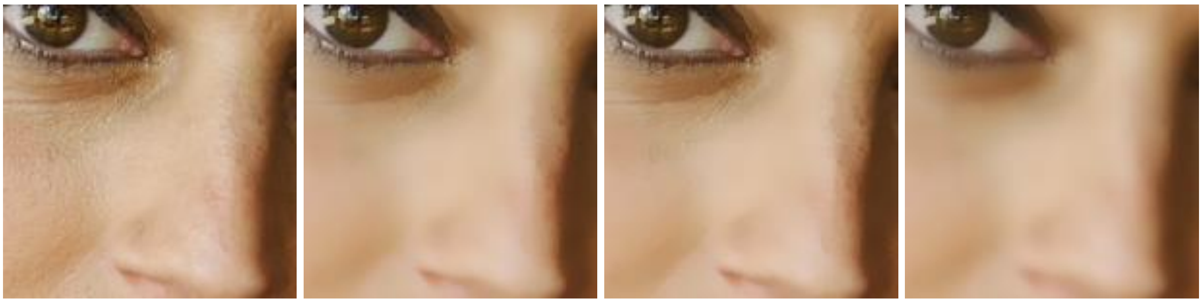
In this assignment specifically, three different solutions of colour image bilateral filtering are implemented, and all outputs of experiment on colour images can be found in **Appendix B**. Detailed parts of outputs from these solutions are listed in **Figure 3.1**:

(a) Source image, the original input.

(b) Splitting the RGB image into three "grayscale" images in R, G, B channel, filtering them by grayscale algorithm separately in each channel, and stacking filtered images back to an RGB image.

(c) Filtering each pixel in RGB image as an RGB vector by calculating their colour space distance in Euclidean way $\Delta E = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$.

(d) Converting the RGB image into CIE-L*ab colour space, filtering each pixel as a L*ab vector by calculating their colour difference with CIE94 formula (Lindbloom, 2017), and converting the filtered image back to RGB image.



(a) source image (b) in RGB channels (c) as an RGB vector (d) as a L*ab vector

Figure 3.1 Parts of Filtered Colour Images where $\sigma_s = \sigma_i = 15$

Although three outputs above differs only on a minor smoothing degree, however, in some scenarios, the implementation way may matter significantly (Tomasi and Manduch, 2009):

Table 3.1 Comparison between Three Implementations of Colour Image Bilateral Filtering

Filter Solution	Performance	Description
In RGB channels	High	As bilateral filters are non-linear, filtering in separate channels will cause false colours in the edge of objects where different colours insect.
As RGB vector	Medium	Significantly different colours in human perception might be very close colours in RGB colour space, causing unexpected smoothing effects on edges and surfaces.
As L*ab vector	Low	CIE-L*ab colour space is specifically designed for the human perceptual non-uniformities (because human eyes are more sensitive to some colours than others), and thus gives the best filtering result. However, the process involves too many float point calculations and gives the worst performance.

4.0 References

[1] Ivrissimtzis, I. (2019). Digital Imaging - Lecture 6 Spatial Filtering II.

[2] Paris, S., Kornprobst, P., Tumblin, J. and a, F. (2009). Bilateral Filtering: Theory and Applications. [ebook] Available at: https://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf [Accessed 14 Jan. 2019].

[3] Rasti, B. (2011). Image Noise of classifications in Remote Sensing.. [ebook] Available at: http://www.dig.cs.gc.cuny.edu/seminars/IPCV/pres12.pdf [Accessed 15 Jan. 2019].

[4] Lindbloom, B. (2017). Delta E (CIE 1994). [online] brucelindbloom.com. Available at: http://www.brucelindbloom.com/index.html?Eqn_DeltaE_CIE94.html [Accessed 15 Jan. 2019].

[5] Tomasi, C. and Manduch, R. (2009). Bilateral Filtering for Gray and Color Images. [ebook] Bombay, India: 1998 IEEE International Conference on Computer Visio. Available at: http://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Tomasi98.pdf [Accessed 15 Jan. 2019].

Appendix A. Filtered Grayscale Images

The following images are a set of filtered grayscale images by a combination of proximity and similarity standard deviation in gaussian function. More detailed images in PNG format can be found under *./output* folder.

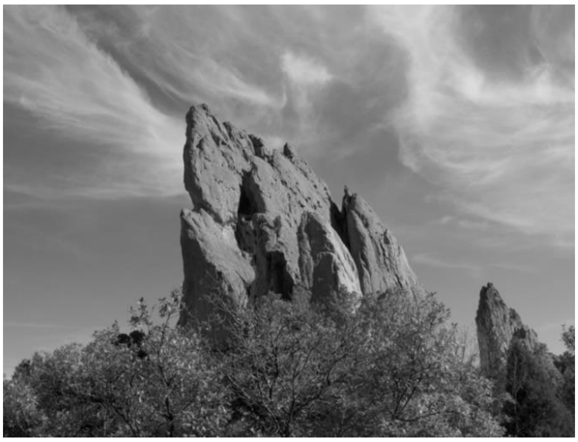
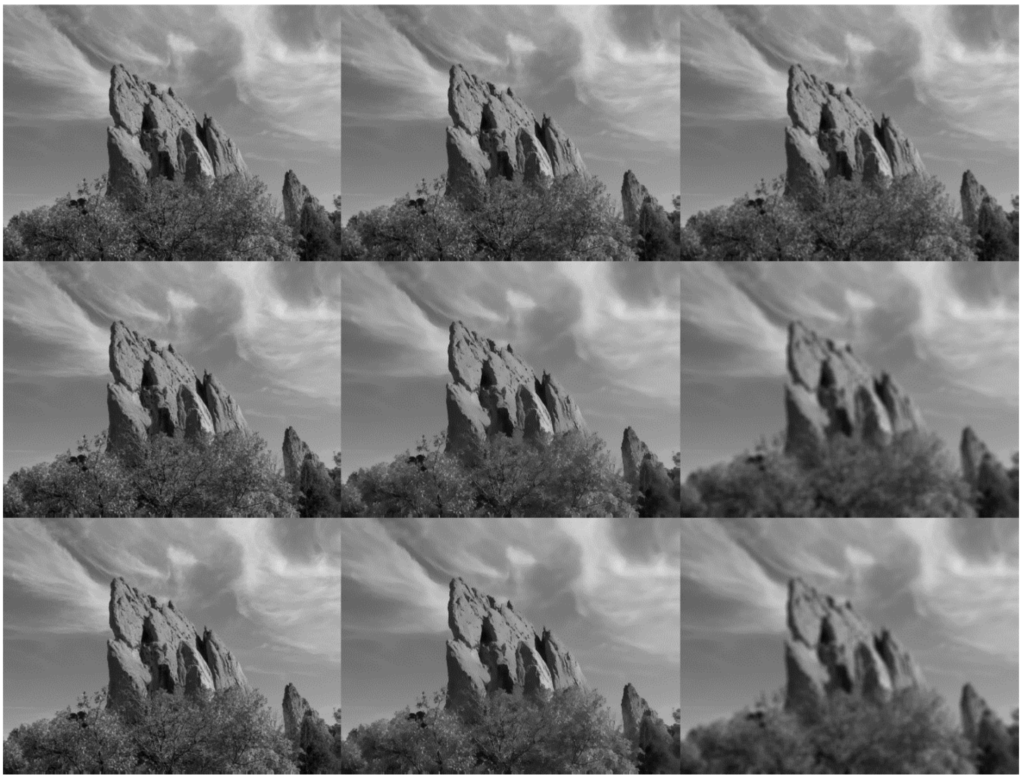


Figure A.1 Grayscale Source Image

$\sigma_i = 1$ $\sigma_i = 15$ $\sigma_i = 1000$



$\sigma_s = 1$

$\sigma_s = 15$

$\sigma_s = 1000$

Figure A.2 Grayscale Filtered Images

Appendix B. Filtered Colour Images

The following images are a set of filtered colour images by different colour distance functions and parameters. More detailed images in PNG format can be found under *./output* folder.

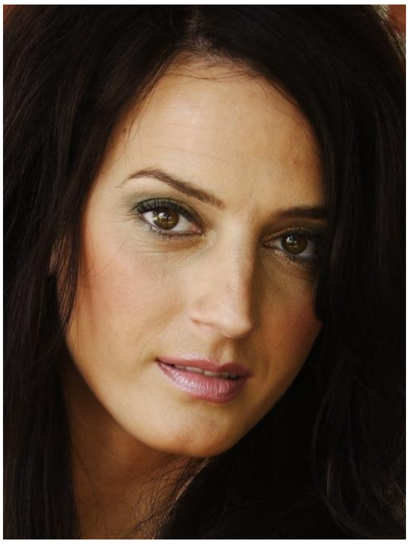


Figure B.1 [left] Colour Source Image

Figure B.2 [below] Colour Filtered Images : A set of filtered colour images by different distance functions and parameters.

The first column calculates colour distance by filtering three RGB channels separately, the second column filters a pixel by calculating its Euclidean RGB distance as a vector, the third filters a pixel as a L*ab vector by CIE76 formula.

The first row uses 15 as standard deviation in its proximity and similarity gaussian function, while the second row uses 1000.