Comp3302 Unit1 Spatial Filtering (smoothing)

C.B.Price September 2020

Purpose	Images may contain noise. When they are transformed (e.g., contrast enhancement) then the noise may become more visible. It is usual to "smooth" images, reducing the amount of noise in the image. This makes them more agreeable to the viewer. This process is called "Image Enhancement"
Files Required	Octave scripts in the zipped folder "Octave_ImProc_Release" on the module site.
ILO Contribution	4
Send to Me	
Homework	

Activities

1 The "Mean" Filter (Part-1)

This filter sweeps an NxN kernel across the image using the correlation operation as explained in class to produce the smoothed output image

(a) Open up the script "FilterMean". There's quite a lot going on here. Noise will be added to the image in two ways, then the mean filter will be applied.
Look for the following in the script

(i) Two functions which add different sorts of noise to the image
(ii) The calculation of the NxN smoothing kernel
(iii) Filtering operation applied to original and two noisy images

(b) Run the script, and when asked set the kernel size to 3. In the Command Window type **kernel** and you will see the smoothing kernel displayed.

(c) Check out the original and smoothed images. What has smoothing done to each image? To what extent has the noise been reduced? Any other visible changes.

(d) Now repeat with a kernel size of 11. Dump the kernel values in Command Window. Now check out the images. Has smoothing worked to reduce the noise? Are there any other visible changes? (Hint – look at the transformation on the *original* image).

(e) Any conclusions on how to choose the size of the kernel?

(f) You may wish to experiment with other images

2 The "Mean" Filter (Part-2)

Here we shall work with a small synthetic image to try and increase our understanding of what is going on. The synthetic image has a block of low values on the left, and high values on the right, i.e., there is a vertical edge in the image.

(b) Run the script NumFilerMeanEdge and when asked set (i) input size = 10, (ii) noise = 0, (iii) Input kernel size 3. Type the following in the Command Window

(i) "**I**" which will show the numeric values of the input image
(ii) "**I_f**" which will show the numeric values of the input image
(iii) "**kernel**" which will show the numeric values of the kernel.

(c) Looking at the input image and the kernel, calculate the output values when the kernel is centred at the locations
(i) row = 5, column = 3
(ii) row = 5, column = 4
(iii) row = 5, column = 5

This should help you understand the operation of correlation. You may wish to take notes and records of these numbers

(d) Run the script again but now with (i) input size = 100, (ii) noise = 0 and let's change the kernel size. Look at the graphs which show you what is happening at the *middle row* of input and smoothed image.
So try kernel sizes of 3,5,7,11,15 and look at what happens to the edge. Does this agree with your results from Activity 1?

(e) Now let's work with noise. Run some experiments with (i) input size = 100, (ii) noise = 0.01 and change the kernel sizes 3,5,7,9,11,15.
What happens to the noise when you increase the kernel size? What happens to the edge when you increase the kernel size.

(f) Can you find the *optimal* kernel size for this image? What does "optimal" mean?

3 Gaussian Filter applied to a noisy edge.

Let's investigate Gaussian smoothing using the script "NumFilterGaussEdge" to look at smoothing of a synthetic noisy edge.

(a) Investigate the following four combinations. Remember after each run you can type **kernel** to see the 2D correlation kernel.

(i) image size = 100, (ii) kernel linear size = 11, (iii) sigma = 2
(i) image size = 100, (ii) kernel linear size = 11, (iii) sigma = 9
(i) image size = 100, (ii) kernel linear size = 5, (iii) sigma = 2
(i) image size = 100, (ii) kernel linear size = 5, (iii) sigma = 9

Look in detail at two things – how much the noise is reduced, how much the edge is blurred. Remember we want removal of noise, but not too much blurring of edges.

(b) Which combination of kernel size and sigma works best?

5 Gaussian Filter applied to real images.

Let's run the spatial filter "FilterGauss" on some images and see if we can recognise the behaviour we saw in Activity 4.

(a) Open up the script in the Octave Editor, and make sure you understand what is going on, especially where the Gaussian kernel is being set up.

(b) Now investigate the four combinations presented in Activity 4 and make some conclusions about which combination works best

(c) Either investigate other combinations of kernel size and sigma, or try the filter out on some other images.