

Topic 4: Image Filtering

Workshop Solutions

Workshop Questions

5.1 At the Edge of an Image

When an image is convolved in real space with a $M \times M$ filter there is a problem of how deal with the edge of the image. Show, with the aid of diagrams how this problem arises.

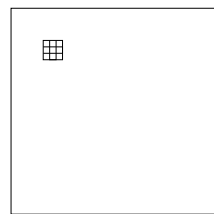
There are three conventional schemes for dealing with this problem, there being

1. Cyclic wrap-around of the image.
2. Extend the image with a constant, (zero or image mean).
3. Replicate pixels at the edge.

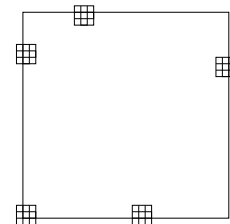
discuss the merits and de-merits of these three techniques.

Solution

Real space filter involved convolving the image by performing a *shift-multiply-add* of the small filter across the image. There is no problem when the filter is totally within the image, but at the edge then part of the filter will be outwith the available image data as shown below:



Filter inside image



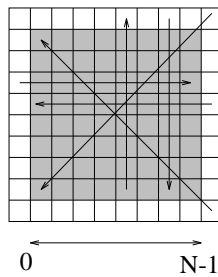
Partially outside Image

So for a $M \times M$ filter and a $N \times N$ image you need image data in the range $-M/2 \rightarrow N + M/2 - 1$.

For Example: for a 3×3 filter and a 128×128 image you need data for image pixel locations in the range $-1 \rightarrow 128$, but you only have data in the range $0 \rightarrow 127$.

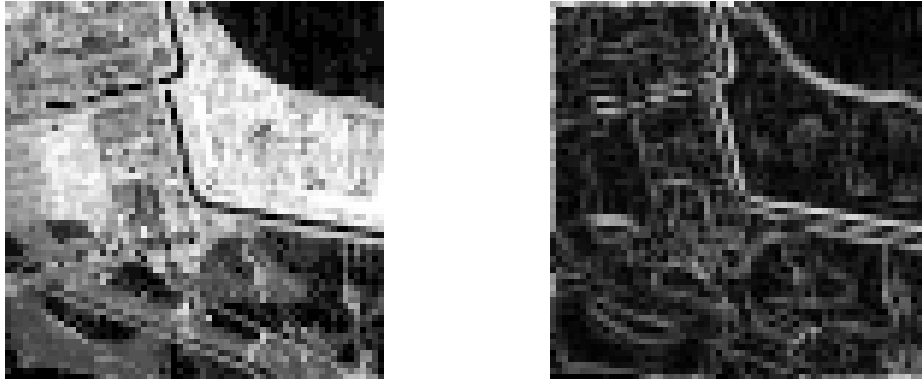
There are a range of solutions to this problem:

Cyclic Wrap-Round: We are dealing with samples data, so as shown in the section on sampling, then the if the image is of size $N \times N$ then the Fourier transform of the image is periodic of period N . We can then apply the same argument to from Fourier space to Real space to show that the image data must also be cyclic of period N . (See next question for details of this), so the mathematically correct scheme is to use a cyclic wrap-around as shown below,



In practice there are problems if there is an intensity discontinuity at the edge of the picture then if you “edge detect” you will get a spurious “edge”.

For example, shown below are a 64×64 part of a satellite image and the result of applying the “edge detect” option from xv.



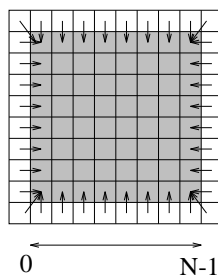
The *edge detect* is a 3×3 filter that calculates the modulus of the gradient with cyclic wrap-around.

There are “spurious edges” all round this image where there is a light region at one edge and a dark region at the other. This is typical for cyclic wrap-around filtering.

Pad with Constant: Take all pixels outside the image to be a constant, with this being either Zero or the Mean of the image. This again will tend to give “spurious edges” round the image when the pixel values at the edge of the real image differ significantly for the value chosen to extend the image. Under these conditions the Mean will give the best results.

This method is often used in image processing packages but I consider it to be the “worst possible option” it is neither mathematically justifiable or gives good results.

Pixel Replicate at Edge: Replicate the edge pixels as shown below for a 3×3 filter.

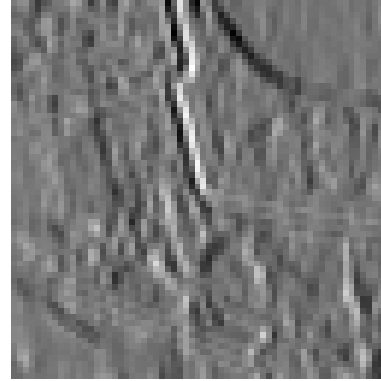


This will remove the “spurious edge” effects noted above but will not produce a mathematically correct image at the edge. Below is shown the same small section of satellite image and a

vertical differential by convolving the image with

-1	0	1
-1	0	1
-1	0	1

using edge pixel replication as implemented in the convolve demonstration program.



This results in the expected vertical differential with no spurious edges at the image boundary. This method gives sensible results but is not strictly speaking mathematically justifiable. In particular the operation is now not a convolution but has a small non-linear component due to the pixel replication which is a non-linear operation.

The only totally *safe* method is to except that for a $M \times M$ filter the filtered image is smaller than the original by $M - 1$ pixels. This is usually, however rather inconvenient, specially if you wish to apply Fourier transformations which restricts the size of the image.

5.2 Edge of image: the Fourier Model

There is the same edge problem at the edge of an image when the filter is applied in Fourier space. Which of the above solution to the problem applies in this case.

Solution

If we have an image $f(i, j)$ with Fourier transform $F(k, l)$ then we have that,

$$f(i, j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) \exp \left(i2\pi \left(\frac{ik}{N} + \frac{jl}{N} \right) \right)$$

Thus $f(i, j)$ is cyclic in two dimensions of period N so if a filter is implemented by the Fourier relation then cyclic wrap-around will occur automatically.



5.3 In Real and Fourier Space

The convolution theorem states that real space and Fourier space convolution is equivalent. If in real space you apply a 3×3 averaging filter calculate the effect in Fourier space.

Use the convolve and fourier programs to show confirm this result.

Solution

In real space we have convolution of

$$g(i, j) = f(i, j) \odot h(i, j)$$

where $h(i, j)$ is a 3×3 mask with all 9 elements being 1. In Fourier space this becomes,

$$G(k, l) = F(k, l) H(k, l)$$

where $H(k, l)$ is the DFT of $h(i, j)$, which is of size $N \times N$ where N is the size of the image $f(i, j)$.

Note that $h(i, j)$ is separable, and can be written as

$$h(i, j) = h(i) h(j) \quad \text{so that} \quad H(k, l) = H(k) H(l).$$

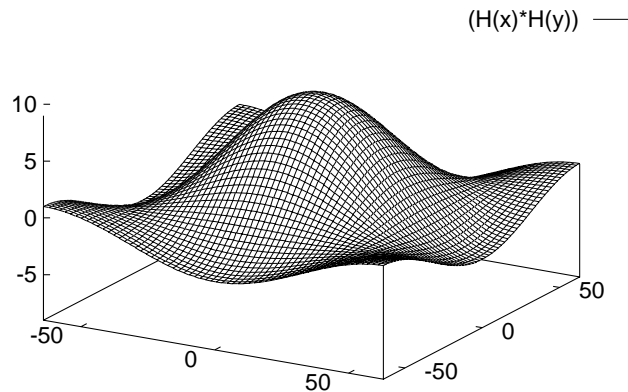
$h(0), h(\pm 1) = 1$, so in Fourier space we have

$$\begin{aligned} H(k) &= \sum_{i=-1}^1 \exp\left(-i \frac{2\pi k i}{N}\right) \\ &= 1 + \exp\left(i \frac{2\pi k}{N}\right) + \exp\left(-i \frac{2\pi k}{N}\right) \\ &= 1 + 2 \cos\left(\frac{2\pi k}{N}\right) \end{aligned}$$

so in two dimensions we get that

$$H(k, l) = \left(1 + 2 \cos\left(\frac{2\pi k}{N}\right)\right) \left(1 + 2 \cos\left(\frac{2\pi l}{N}\right)\right)$$

which in surface plot is:

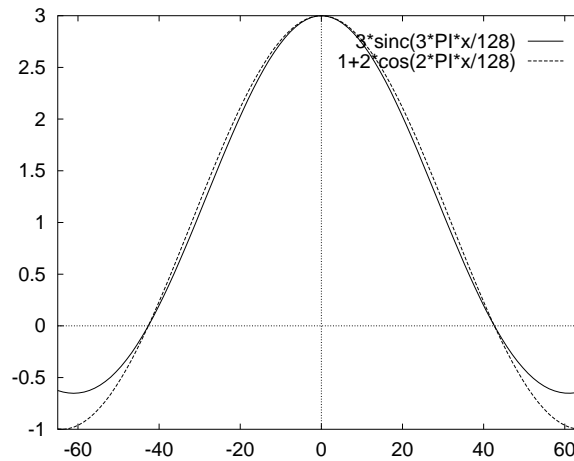


This is not the expression that you would expect, you would expect a sinc() shape from the Fourier transform of a “top-hat” of length 3 pixels. This should give an expression of,

$$H(k) = \text{sinc}\left(\frac{3\pi k}{N}\right)$$

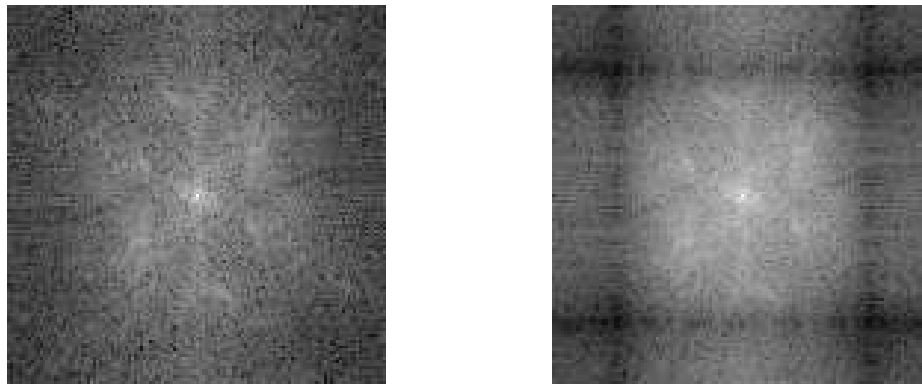
however the digital “top-hat” implemented by the filter is three δ -functions at $0, \pm 1$ which is a poor representation of a “top-hat”.

In fact the $1 + 2 \cos()$ expression for $H(k)$ is similar shape to the ideal $\text{sinc}()$ relation, both of which are plotted below

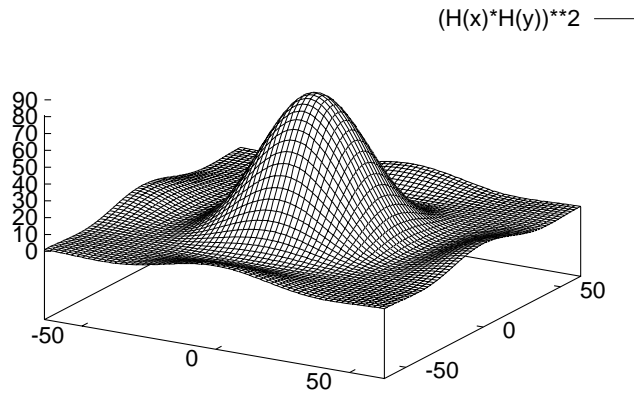


In particular they have zeros at the same location and only differ significantly at high spatial frequencies.

Part 2: Use the usual toucan.pgm and use the convolve programme to form a 3×3 average. The Power Spectrum of the Toucan image and the filtered Toucan image are shown below,



From above we see that the Power spectrum should be weighted by $|H(k, l)|^2$, which is of shape,



which is exactly what the two power spectrum plots above show.

Note: The horizontal and vertical “striping” on the power spectrum images is due to the edge pixel replication scheme used at the image edges as described in the above solution.

5.4 Applying Two Filters

You wish to smooth an image by applying a 3×3 9 point average filter followed by a 3×3 laplacian filter. Show that this can be implemented in a single convolution using a 5×5 filter and calculate the elements of this filter.

Solution

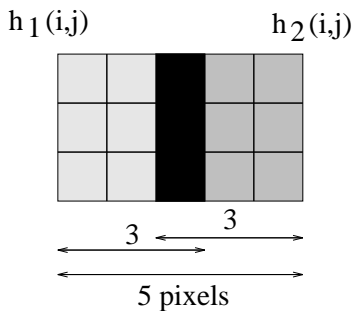
If we have two filters $h_1(i, j)$ and $h_2(i, j)$ then if you apply the both filters one after the other you get,

$$g(i, j) = h_2(i, j) \odot (h_1(i, j) \odot f(i, j))$$

but convolution is a linear operation, so the order of the convolution does not matter, so that

$$g(i, j) = (h_1(i, j) \odot h_2(i, j)) \odot f(i, j)$$

If both $h_1(i, j)$ and $h_2(i, j)$ are 3×3 filters, then the convolution is a 5×5 array.



You calculate the 5×5 array by *shift-multiply-add* which is given by:

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

5.5 Taking Median Filters

Select a standard image and use the program `median` to form Median filters of various sizes. Confirm that you get a smoothed image but retain the edges as theory predicts.

View the Fourier transform of each of these Median Filtered images and comment on what you find. (Save the image from the median program from within `xv`, and input this into the `fourier` program.)

Solution

I will use the *toucan* you can use any other `pgm` image you like. The images and their Fourier transform as shown below.

Original image and its power spectrum:

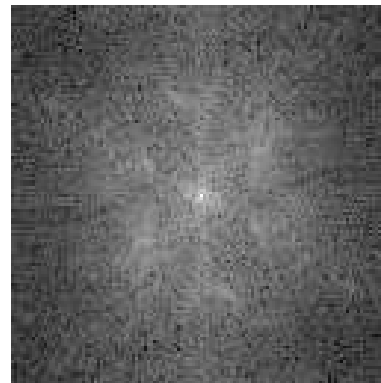
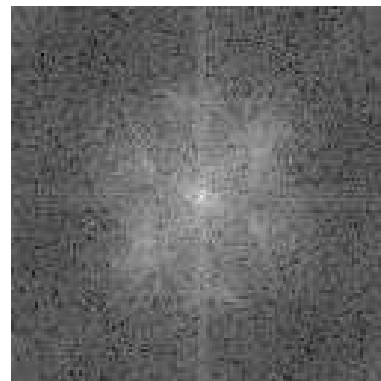


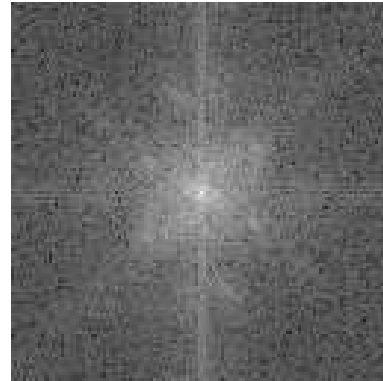
Image you have seen many times before. Most of the power at low frequencies, but significant power at all spatial frequencies from the sharp edges in the image.

3×3 Median Filter:



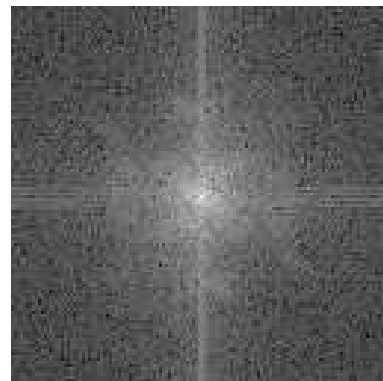
Real space image has slightly “blocky” appearance but retains the sharp edges. Power spectrum has same basic shape as original but has significantly reduced power at high spatial frequencies. However high frequencies are not totally removed and there are no areas of zeros. (Compare this to the 3×3 average above.)

5×5 Median Filter:



Real space image showing significant “blocky” appearance with areas of constant intensity with sharp boundaries. In Fourier space there is further reduction of the high spatial frequencies but again not total removal, and no problematic regions of zero. (The horizontal and vertical line of spatial frequencies is coming from edge effects, in particular this median filter uses the “repeat at edge” scheme which is not well matched to the cyclic nature of the Fourier transform.)

7×7 Median Filter:



Very blocky image in real space with relatively large areas of constant intensity and most of the small features lost. Again note that the edges are still sharp. In Fourier space you get further smoothing. The edge effects are now becoming significant since the filter is becoming a significant fraction of the size of the image.