

Advanced Image Enhancement (Chapter 14)

Among the most useful tools in image processing are those that generate a new pixel value based on the relationship between the value of the pixel of interest and the values of those that surround it. These processes are called “neighborhood” operations and involve the pixel’s immediate (and not so immediate) neighbors.

Convolution Filtering

Convolutions are operations in which new pixel values are determined from linear combinations (for example addition and multiplication) of a pixel value with its neighbors. Convolution can be used to sharpen or soften an image. It can be used for sharpening images with bad seeing, and works best on images of planets.

In convolution, two functions are overlaid and multiplied by one another. One of the functions is your image and the other is a “convolution kernel”. You can represent the image by a large array of numbers (pixel values in x and y) and the 2nd function by a relatively small array. In convolution, the kernel (the small array) operates on the image. Hence the kernel is frequently called an operator.

Here is a simple example.

Step 1: Open one of your calibrated images from your previous work on image enhancement. Choose a fairly faint star in the image, and record the pixel values of 10 pixels along a row that bisect the star. You can use this array of numbers to represent your image in this example.

Step 2: On nights of good seeing, the light from a star can be considered a point source, which means that all light from the star falls all in one pixel. On good nights, we could represent the distribution of light falling on the detector as an array of numbers [0.00 **1.00** 0.00]. But on nights of bad seeing, the light is spread out by the turbulent atmosphere. The light might have spread so that only half of it reaches the target pixel, and the other half falls on pixels to either side. This distribution is sometimes called the “point spread function”. For this example, the point spread function can be represented as [0.25 **0.50** 0.25]. It says that the light from the star spreads across three pixels, with 50% going to the central pixel, and 25% to the pixels on either side. Let’s imagine that for some reason we want to blur our original image a bit. So this little array [0.25 **0.50** 0.25] will be our operator.

Step 3: Now how do you apply this operator to your “image”? In one dimensional convolution, each pixel in the convolved image is the sum of the product of each element in the operator times the corresponding pixel value in the image.

$$N_x = (\sum k_i O_{x+i}) / \sum k_i$$

Where N_x is the x-th pixel in the new image, O_x is the x-th pixel in the original image, and k_i is the i-th element in the operator. The summation is done over all the elements of the operator, and repeated for all elements of the original image. Since the convolution is carried out for every pixel in the final image, as you move from pixel to pixel in the original image, the operator “steps” along the original image generating a new pixel value for the new image.

One final thing: the traditional convention is to do the summation over the index i from $-r$ to r , instead of starting at 1 (but I can’t write it out mathematically in Word). This implies that your operator will consist of a symmetrical arrangement of $2r+1$ elements with the 0th element in the center. So the summation for our little operator with three values will run from -1, 0, +1.

Step 3b: Calculate the new values using your image and operator. The point of this exercise is to show you how convolution works on an image. Here’s an example. Let’s say your single row image is [4 4 4 4 100 4 4 4 4]. Your operator is [0.25 0.50 0.25]. The first pixel in the new image will be:

$$N_1 = [(0.25 \times 4) + (0.5 \times 4) + (0.25 \times 4)] / (0.25 + 0.50 + 0.25)$$

$$N_1 = 4.0$$

$$N_2 = 4.0$$

$$N_3 = 4.0$$

$$N_4 = 28.0$$

If you continue on, your new image will look like [4 4 4 28 51 28 4 4 4]. The original sharp star image is now spread out over three pixels.

Now lets work on an actual image using AIP’s features.

Step 4: Go to the directory labeled “marspics” and open the image labeled “moon.fts”. This is an image of the moon taken through a 6mm telescope. Click Enhance → Convolution Filters → Crispen. A new image will appear. Crispen is an example of a sharpening operator. Its function is to increase the difference between a central pixel and the pixels in its immediate neighborhood. These high-pass filters are designed to enhance contrast, but also enhance any accompanying noise. A useful measure of the filter’s strength is its contrast enhancement, which equals the value of the central pixel divided by the sum of the elements in the filter. Most sharpening operators are designed so that the sum of the elements of the operator array is 1. Here are typical crispening operators:

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{array} \quad \text{and} \quad \begin{array}{ccc} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{array}$$

- 1) Compare this image with the original. How has the contrast and detail changed?
- 2) Compare the “hot” pixels in the original image with those in the enhanced image. Crispen enhances everything, it has no way of knowing which pixels are noise.

Step 5: Go to the directory labeled “marspics” and open the mars images (there should be 20 or so). View the fits headers of two of the images and record the exposure time.

- 1) Are they the same exposure time?
- 2) Do the images all look the same?
- 3) How can you explain the differences?
- 4) Record which Mars images are clearer than the others.

Step 5: Pick one of the images that is fairly sharp and another that is of medium quality. Start with the fairly sharp image. Click Enhance → Convolution Filters → Crispen. A new image will appear. Compare this image with the original. How has the contrast and detail changed?

Step 6: Return to the original Mars images. Now try the “Enhance → Convolution Filters → Sharpen”

- 1) How does “sharpen” compare with “crispen”?
- 2) Which filter (crispen or sharpen) do you feel is stronger?

Step 7: Open one of the calibrated images of an object we’ve observed in class. Make sure it is an image that contains stars. Try “Enhance → Convolution Filters → Sharpen” on this image.

- 1) What happens to the stars?
- 2) Looking at the crispening operators given above, why do you think this happens to stars?

Step 8: Return to the original Moon image. Apply “Enhance → Convolution Filters → Smooth” and “Enhance → Convolution Filters → Blur” to the original images. The crispen and sharpen filters are called “high pass” filters. They are designed to bring out details. The Smooth filter is an example of a “low pass” filter.

- 1) Compare the smoothed image with the original. What are the differences?
- 2) What do you think “low pass” filters are designed to do?

Step 9: Sometimes giving objects in an image a three dimensional feel helps to isolate faint objects. The bas relief filter makes a copy of the images, shifts it by one pixel, and subtracts that from the original image. Try the “Enhance → Convolution Filters → Bas Relief” on the original moon and Mars images. What happens to each?

Step 10: Experiment with the Gradient, 3x3, and 5x5 Emboss filters. What do you think these filters are designed to detect?

Step 11: More mathematically complex convolutions can reveal intricate detail in images.

Experiment with the Sobel, Kirsch, and Prewitt Operators on the moon images.

- 1) What do you think these filters are designed to enhance?

Close all of your open images.

Convolution by Unsharp Masking

Unsharp masking is an analysis technique used to bring out small-scale variations in brightness. The technique works as follows: A copy is made of the original and convolved with an operator that averages pixel values over a region. This blurred, or soft, image is called the unsharp mask, after a masking process used during the last century by photographers. The unsharp mask is then convolved with another operator called the “unity kernel” (see chapter 14, page 388 for the exact structure of the unity kernel). Finally, the two images are summed. Unsharp masking acts as a high frequency enhancement filter. Information about large scale structure is suppressed, and information about details has been multiplied.

Step 1: Open the file labeled “moon2.CCD”, and a fairly sharp Mars image. Start with the moon image. Click on “Enhance → Unsharp Mask”. Leave the Mask Type set to Gaussian with a Radius of 2 and set the Contrast to 5 and click apply.

- 1) What happens to the image?

Step 2: Repeat for the Mars image. What happens to this image?

Step 2a: Open the image labeled ‘jupiter.ccd’ Invoke the unsharp mask tool and apply a Gaussian mask with a radius of 3 and a contrast of 4. Click the “AutoMin/Max” button on the Image Display Control window to see the entire range of brightness in the image. What happens to the image?

Step 3: Return to the original Mars image. Click on “Enhance→Unsharp Mask”. Change the Mask Type to Parabolic. Be sure to click on the original image you want to use. Compare with your Gaussian results.

- 1) What happens when you lower the Radius value?
- 2) What happens when you increase the Radius value?
- 3) What happens when you lower the contrast?
- 4) What happens when you increase the contrast?

Step 4: Experiment with the Mars image and different types of unsharp masks.

- 1) Which do you think best brings out the detail in the planet?
- 2) What are some of the disadvantages?

Step 5: Open the calibrated image of one of the objects we’ve observed in class (an image that contains stars). Experiment with the various types of unsharp masks.

- 1) What does each unsharp mask do?

