# Lab 5: Spatial Filtering

*due TBD*

**Goals:**

1. To gain an understanding of convolution filtering.
2. Learn spatial filtering procedures and applications.

**Introduction:**

A common and important practice in digital image processing is to enhance imagery using spatial filters. Filters are commonly used for edge enhancement, noise removal, and the smoothing of high frequency data. These filters work by enhancing or suppressing spatial details to improve visual interpretation in the final image. Spatial domain filters are "neighborhood" (or "local" or "group") operations, in which the DN of a given pixel is reassigned based on the values of a specified number of surrounding pixels.

Spatial frequency refers to the number of changes in brightness value, per unit distance, in the image. Low spatial frequency implies changes in brightness over many pixels and, therefore gradual transitions in contrast. An area with high spatial frequency will have rapid change in brightness, that is many, non-gradual contrast transitions over a given distance.

Low-pass filters are used to emphasize low spatial frequencies and are designed to smooth out the high spatial frequency content of an image. Since they de-emphasize high spatial frequency, and the smallest details in an image are recorded by high spatial frequencies, the resulting imagery will be slightly blurred. High-pass filters, on the other hand, are used to emphasize high spatial frequency data. Often they are used to enhance and sharpen features such as roads, land-water boundaries (and edges in general), and other high spatial frequency details. These filters are often referred to as sharpening filters because they generally enhance edges without affecting the low frequency portions of the image. Filtering can also help remove noise from an image by averaging out the effects of random noise fluctuations. Median filters are especially useful for this task.

In many remote sensing applications, the most valuable information derived from an image is contained in the edges of features. Specialized edge detection filters are used to emphasize these boundaries and make them easier to analyze. Edge detection filters can be either directional (e.g. gradient filters) or non-directional (e.g. Laplacian filters).

Spatial filtering works by *convolving* the image with a moving, two-dimensional rectangular array of weighted values (the "kernel" or "mask"). Think of the kernel as a moving window that covers a small patch of pixels as it traverses across the image. In most cases, a new brightness value for the image pixel location beneath the center of the kernel is calculated according to some function of the original pixel value and some neighborhood around it, for example, the weighted average of the pixel and the neighboring pixels also covered by the kernel. The weighting coefficients are the values in the kernel. Some filters are non-linear and do not rely on the weighed mean, but instead are based on other algorithms. In any case, once the new pixel value is calculated and the result is placed in the appropriate location in the output image, the kernel then shifts over by one pixel and performs the same operation again. Note that the input image remains unchanged! The results of the convolution at each pixel location are recorded in a new image that you are building as the process continues. This process is called a two dimensional convolution, and the filter is often called a convolution kernel. The size of the kernel and the values of the kernel coefficients determine the behavior of the filter.

**Instructions:**

1. You will first work in ENVI (Classic) with individual bands of the `test_ms.img` dataset. In ENVI, choose just one of the VNIR bands and just one of the TIR bands. You will work with these two bands individually, as separate, single-band, grayscale (not RGB) images.

2. Open the VNIR band and the TIR band in separate image windows. For both images, extract some spatial profiles using the: `Tools> Profiles> Arbitrary Profile (Transect)` tool in the image window menu. You can define the endpoints exactly by selecting `Options> Create Transect from Endpoints> Pixel Based` and entering start/stop endpoints. You will have plots of DN vs. distance in your image. From these plots, get a feeling for the meaning of "spatial frequency". What can you say about the frequency content of your image from these transects? What do edges look like? What do patches of (near) constant DN look like?

3. Navigate to `Filter> Convolutions and Morphology`. The `Convolutions and Morphology Tool` dialog box will open. Use the `Help` menu to read the ENVI help files on convolutions and morphologic processing before proceeding.

   In the `Convolutions and Morphology Tool` dialog box you have a variety of options. The `Convolutions` menu lists the various spatial filter types you can apply. Similarly, the `Morphology` menu lists various morphologic convolutions. Choosing one of these convolutions/morphologic operators will present you with a set of user-editable default parameters in the main part of the window. These parameters will vary depending on the convolution you choose to use. These parameters include the kernel size, an "add-back" factor, and the kernel coefficients themselves:

   a) The `Kernel Size` is the size of the filter window. You can specify square or non-square (though still rectangular) kernels by toggling the

`Square Kernel` selection under the `Options` menu. You can interactively increase or decrease the size of the kernel.

b) The `Image Add Back` allows you to specify how much of the original image to "add back" to the result of the convolution. This value is the percentage of the original image that is included in the final output image. For example, if you enter `40%`, then `40%` of the original image is added to `60%` of the convolution filter image to produce the final result. A value of `0%` will apply the pure kernel to the image. A value of `100%` will result in the unmodified, original image. "Adding back" a percentage of your original image helps preserve the spatial context and is typically done to "sharpen" an image or to improve the interpretability of the convolution output.

c) The `Editable Kernel` is the matrix of kernel coefficients that will be used in the convolution. To edit the kernel and change any of the weight values, double click on the coefficient you want to change, modify the value, and hit `<Enter>` to accept the change.

d) You can save and restore kernels you create or modify under the `File` menu.

e) You can apply a convolution to an image band by using the `Quick Apply` button. This will automatically create a new image window that shows the results. You can select the image to use as the default input for the `Quick Apply` in the `Options` menu. To save the results of a `Quick Apply` to a file or to memory, select `File > Save Quick Result`. Alternatively, if you would like to save the results of a convolution directly to a file or to memory (without automatically displaying the results), use the `Apply to File…` button.

4. You will now apply a `low-pass` filter.
   a) Use your VNIR band as the input image, and perform three different low-pass convolutions (with no add-back): one with a 3x3 kernel, one with a 5x5 kernel, and one with a 7x7 kernel. You can also experiment with larger kernels (and different add-back factors) as you see fit.
   b) When complete, you should have at least four images: the original VNIR band and at least three filtered versions.
   c) Load them all in separate image windows. Tip (optional): To link your display windows, select `Window> Link Displays` and select the display windows to link.
   d) Describe how the appearance of the image changes with increasing kernel size (and amounts of add-back if you applied different amounts).
   e) Extract statistics from each image (`Basic Tools > Statistics`) and determine how the minimum, maximum, and standard deviations change with kernel size.

f) How have sharp linear features/edges/other features in the original VNIR image changed after applying the low-pass filters? Comment on the enhancement effects of the low-pass filter.

5. Now apply a `High Pass` filter.
   a) Use your TIR band as the input image and perform three different high-pass convolutions (with no add-back): one with a 3x3 kernel, one with a 5x5 kernel, and one with a 7x7 kernel. You can also experiment with larger kernels (and different add-back factors) as you see fit.
   b) When complete, you should have at least four images: the original TIR band and at least filtered versions.
   c) Load them all in separate image windows (and link them if you like).
   d) Describe how the appearance of the image changes with increasing kernel size (and amounts of add-back if you applied different amounts).
   e) Extract statistics from each image and determine how the minimum, maximum, and standard deviations change with kernel size.
   f) How have sharp linear features/edges/other features in the original TIR image changed after applying the high-pass filters? Comment on the enhancement effects of the high-pass filter.

6. Now filter both the VNIR and TIR images with the `Laplacian`, `Sobel`, and `Roberts` edge detectors. Experiment with different kernel sizes (for the Laplacian filter) and the add-back factor (for all three). How do the results of these processes compare to one other? What worked best? What features do these convolutions emphasize? Remind yourself as to what each of these convolutions is doing (see lecture notes and Jensen). Given this knowledge, your knowledge of the geology/surface cover in your scene, and your observations of the convolution results, which process do you think is giving you the most useful/interesting information?

7. Examine your original VNIR and TIR images (and geologic maps and other ground control if available) and take note of lineaments/faults/geologic units/other surface cover features (roads, fields, etc.) with distinct orientations. Implement `directional filter`(s) (of varying size and add-back factor as you see fit) to enhance edges with those orientations (and high spatial frequency contrast-variations oriented 90° to those defining edges). What kernel size works best?

8. **In MATLAB, develop two simple programs for performing:**

   **a) a 2-D median filter operation, and**

   **b) an "unsharp mask" operation.**

   **For the median filter routine, you can use your median function program from Lab 1 (perhaps slightly modified) as the starting point.**

**For the unsharp mask routine (see lecture notes and item 6 above), you are allowed to use the `filter2()` function in MATLAB to compute the low-pass filter. The syntax is:**

```
Y = filter2( c , X ) ;
```

**where `X` is the original input image (matrix of brightness values), `c` is an `n` by `n` matrix kernel, and `Y` is the filtered output image. Note that you will have to define what the matrix kernel is! Type "`help filter2d`" in MATLAB for more information.**

**In //geotwo/ you are given two images. One is a VNIR image with a significant amount of noise (artificially applied). The other is a TIR image of the same area. Apply your median filter to the VNIR image. Then apply the unsharp mask to both the filtered VNIR image and to the TIR image. What did these operations do? Extract statistics (using either ENVI or MATLAB) to quantify your answers.**

## What To Turn In:

1. A very brief write-up of the processing you did to your imagery, with answers to the various questions posed in items 1-9 above. Don't forget to also include a discussion of your MATLAB-derived results! It must be clear and well written (punctuation, grammar, syntax, etc.). Limit your report to no more than 3 typed pages (normal margins, 12 point font, reasonable spacing), not including figures.

2. Representative images, plots, tables, etc. (as necessary) showing the results of your processing.

3. MATLAB code for the median filter and unsharp mask (item 9 above).

## How To Turn In:

For the report, produce a single PDF that includes both your report text and your figures. Do not turn in any loose image files of any format (especially ENVI). I also want the `.m` files for your MATLAB work. Be sure to put your name in comment at the top of all your `.m` files. Also be sure to adequately comment your code so that I know what your programs do and how they work. Turn in your files via the `DROPBOX` on `//yowa`. Be sure to properly package your files into a folder called `YOURNAME_lab5` before submitting them.