Check for updates

Smoothing surfaces and attributes

Downloaded 06/27/21 to 223.189.249.164. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/page/policies/terms
DOI:10.1190/tle33020128.1

Welcome to this new column. Every two months, a geoscientist will present a brief exploration of a geophysical topic. The idea is to take a tour bus around a subject and point out some of the sights, perhaps stopping briefly at an exemplary problem or instructive viewpoint. So far it's useful, but maybe not remarkable.

The remarkable thing, I hope, is that the tour will be open access. The tutors will use only open data sets that anyone can download. There will be no proprietary software. I will strongly encourage the use of Octave, R, or Python, all high-level (that is, easy-to-learn) programming languages for scientists, and the important parts of the code will be shared. I've tried to give a flavor of all this in today's tutorial, using Python. If you are new to Python, IPython is a great place to start—visit *ipython.org/install.*

There's more. Not only will the tutorials use open code and data, they will also be openly licensed. SEG has chosen to openly license these columns for free use and reproduction anywhere by anyone. This will also allow them to appear on *wiki.seg.org*—where anyone can edit and add to them—at the same time they appear in *TLE.* This is a bold and exciting move by the Society.

To kick off the series, I take a quick look at smoothing seismic horizons and attributes. Next time, Leo Uieda will write about uses and abuses of Euler deconvolution. If you have a "how to," "why do," or "what is" that you would like to share, please contact me at matt@agilegeoscience.com.

Several years ago, I wrote an article about smoothing (Hall, 2007). At the time, I was aware of but unversed in the ways of reproducible science and open-source software. So I offered to share my code, but it wasn't online. And it wasn't really code, it was a bunch of Landmark PowerCalculator scripts. My data set was proprietary. None of what I'd shown was practically reproducible by anyone. This time it will be different.

Filtering horizons

Seismic horizons, whether of the time, depth, or attribute variety, are just big tables of numbers. Think of them as arrays (a bit like a spreadsheet full of cells containing numbers) or even as images (in which the cells are pixels). Indeed, we can be even more general—all data can be thought of this way. Logs are 1D arrays, and seismic volumes are 3D arrays.

Like all subsurface data, these arrays or images contain signal and noise. Sometimes they are so noisy we want to do something about it. Broadly speaking, there are two approaches to noise removal: linear and nonlinear filters. Linear filters treat all pixels the same. Often they mix the noise with the signal. This is how stacking works—we average the traces across offset angles. The assumption is that the noise cancels itself because it is random and the signal is not. Nonlinear filters treat pixels differently. Most methods assume some spectral characteristic of the signal (usually that it is spatially smoother than the noise, which is the same as saying that it has a smaller wavelength in the wave-number domain) and use it to discriminate between signal and noise.

Convolution

Linear filters are usually implemented by convolution. Perhaps the simplest kind of convolution is a moving average. Imagine taking the moving average of a well log, with seven samples going into each mean. The seven-sample group is called a kernel. In the case of a mean, all the sample weights are equal—we sum the log samples in the neighborhood at each stop and divide by seven. Sometimes we might increase the weight of the central value at each stop and decrease the weight at the edges, remembering to ensure that the weights still sum to one.

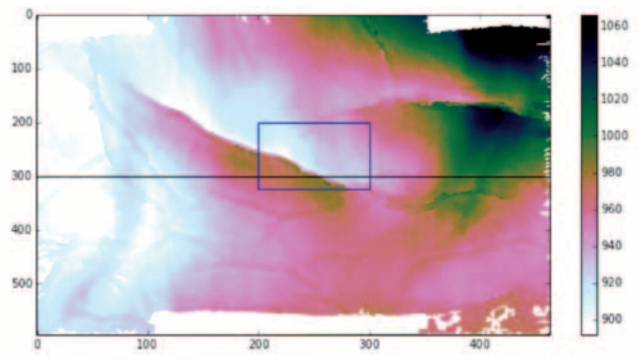

Figure 1. *The original horizon, Penobscot 3D, Nova Scotia, Canada.*

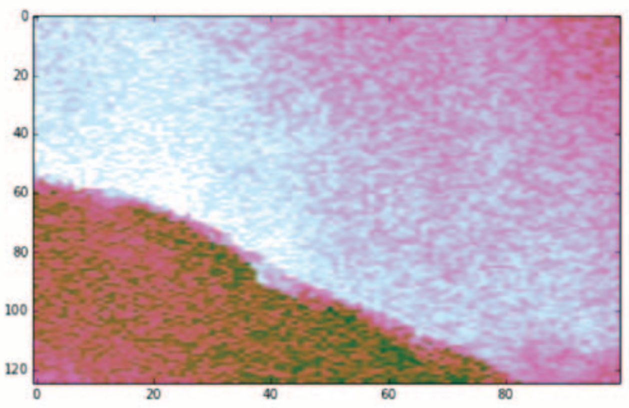

Figure 2. *Detail of the horizon, with added random noise.*

Downloaded 06/27/21 to 223.189.249.164. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/page/policies/terms
DOI:10.1190/tle33020128.1

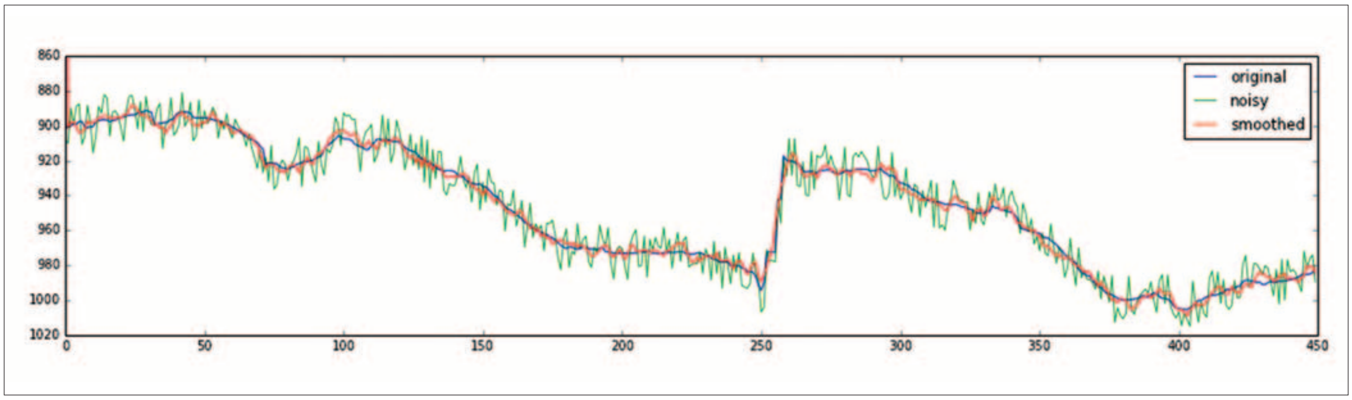

Figure 3. *Cross section of the original horizon, the noisy horizon, and the smoothed horizon.*

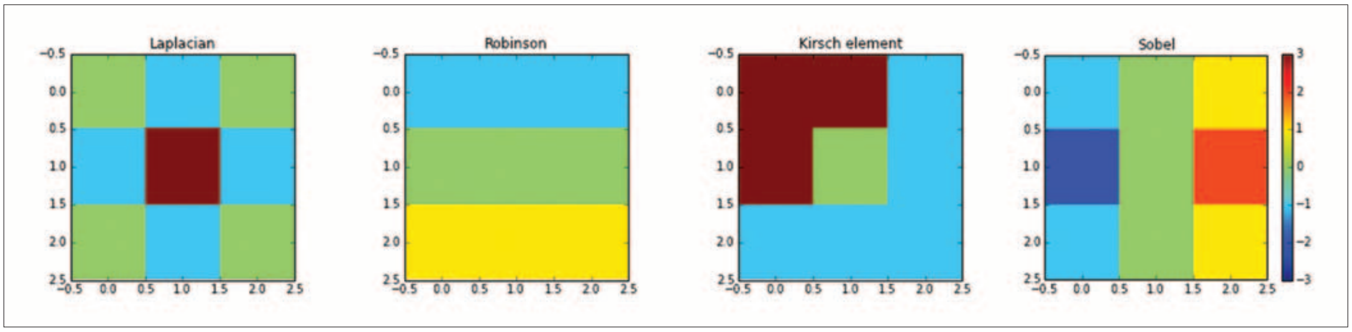

Figure 4. *Nonsmoothing kernels are just as easy to experiment with.*

A 2D kernel is easy to make in Python using the math library NumPy:

```
> import numpy
> kernel = numpy.array([[ 0.111111, 0.111111,
0.111111],
 [ 0.111111, 0.111111, 0.111111],
 [ 0.111111, 0.111111, 0.111111]])
```

Once I have a horizon array to operate on—shown in Figure 1 (without noise) and detailed in Figure 2 (with noise)—convolution can be done using another library, SciPy:

```
> import scipy.signal
> output = scipy.signal.convolve2d(horizon,
kernel)
```

A simple plotting command displays a cross section of the result (Figure 3). Then I can easily play with other kernels that might do other tasks, such as edge detection (Figure 4).

I will look more closely at nonlinear filters in a future tutorial. Because they can't be implemented by a simple convolution, they are harder to implement than linear filters. A great many have been implemented in various image-processing libraries for Python (such as PIL, Pillow, mahotas, and ScipPy), but we might also like to build our own.

Find a version of this article in the SEG Wiki at *wiki.seg.org*. The code to generate all of the figures is in an IPython Notebook at *github.com/seg.* TLE

—Matt Hall
Agile Geoscience
matt@agilegeoscience.com

An Encyclopedia for Applied Geophysics
SEGWiki

Reference

Hall, M., 2007, Smooth operator: Smoothing seismic horizons and attributes: The Leading Edge, **1**, 16–20, http://dx.doi.org/10.1190/1.2431821.

© The Author(s). Published by the Society of Exploration Geophysicists. All article content, except where otherwise noted (including republished material), is licensed under a Creative Commons Attribution 3.0 Unported License (CC BY-SA). See http://creativecommons.org/licenses/by/3.0/. *Distribution or reproduction of this work in whole or in part commercially or noncommercially requires full attribution of the original publication, including its digital object identifier (DOI). Derivatives of this work must carry the same license.*