Prediction error filters

Brad Artman

brad@sep.stanford.edu

ABSTRACT

For multi-dimensional data spaces the prediction error filter provides us a powerful tool for interpolation and a convenient substitute for the inverse covariance matrix needed for least squares inversion problems.

1. Basic filtering

For the 1D case, consider the matrix convolution of time signals. Given a data pulse $\mathbf{b} = (b_0, b_1, \ldots, b_n)$ and a filter $\mathbf{f} = (f_0, f_1, \ldots, f_n)$, the output transient convolution $\mathbf{c}$ is

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_{n+m} \end{bmatrix} = \begin{bmatrix} b_0 & 0 & \cdots & 0 \\ b_1 & b_0 & 0 & \\ b_2 & b_1 & b_0 & \\ \vdots & b_2 & b_1 & \ddots \\ b_n & \vdots & b_2 & \\ 0 & b_n & \vdots & \\ \vdots & & & b_{n-1} \\ 0 & 0 & \cdots & b_n \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_m \end{bmatrix} \tag{1}$$

The prediction error filter has the special definition that the leading coefficient will be constrained to 1, and a least squares formulation will be employed to find the rest of the coefficients of the filter such that when the data is convolved with the

filter the output **c** is approximately 0.

$$\begin{bmatrix} x_0 & & \text{zeros} & \\ x_1 & x_0 & & \\ x_2 & x_1 & x_0 & \\ x_3 & x_2 & \cdot & \ddots \\ \vdots & & \cdot & \\ x_n & & \cdot & \\ & x_n & & \\ & & \ddots & \\ \text{zeros} & & & x_n \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ \\ \\ \cdot \\ \cdot \\ \cdot \\ \\ \\ 0 \end{bmatrix} \tag{2}$$

If we multiply both sides by the complex conjugate of the data, we obtain a Toeplitz matrix we can solve to calculate the prediction error filter **a** using the the autocorrelation of the data, **r**.

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots \\ r_1 & r_0 & r_1 & \\ r_2 & r_1 & r_0 & \\ \vdots & & & \\ \cdot & & & \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3}$$

This provides intuition, as well as practical computation techniques. The inverse of the prediction error filter (PEF) has the same spectrum as the data. Also, in practice, PEF's tend to be quite short relative to the data and thus easily invertible.

This framework is easily extendable to multiple dimensions.

2. Properties

With the basics behind this simple definition of the the PEF, the most important property of these filters can be presented. Because we estimate the PEF on the autocorrelation of the data, we construct a filter specifically to absorb the color of the data and *the output of data filtered by a PEF estimated on it tends to whiteness.*

Conversely, convolving random noise with a PEF will synthesize a data set with the same spectrum as the data the filter was estimated on, and *hopefully* have similar structure.

Thus, these filters lend themselves readily to deconvolution, interpolation, noise separation, synthesis, and stability applications.

PEFs are very flexible, quick to calculate, and are implemented in the space domain rather than the Fourier domain. This lends itself to a malleability to handle problems of missing data, Fourier periodicity, and non-stationarity.

3. PEFs as dip-filters

In the framework of spatial statistics, data considered to be the superposition of various dips can be characterized well with these filters, which will also provide a geometric understanding to these beasts. If data can be characterized as a collection of dips, and we understand that the PEF is a data annihilator, then a filter of the

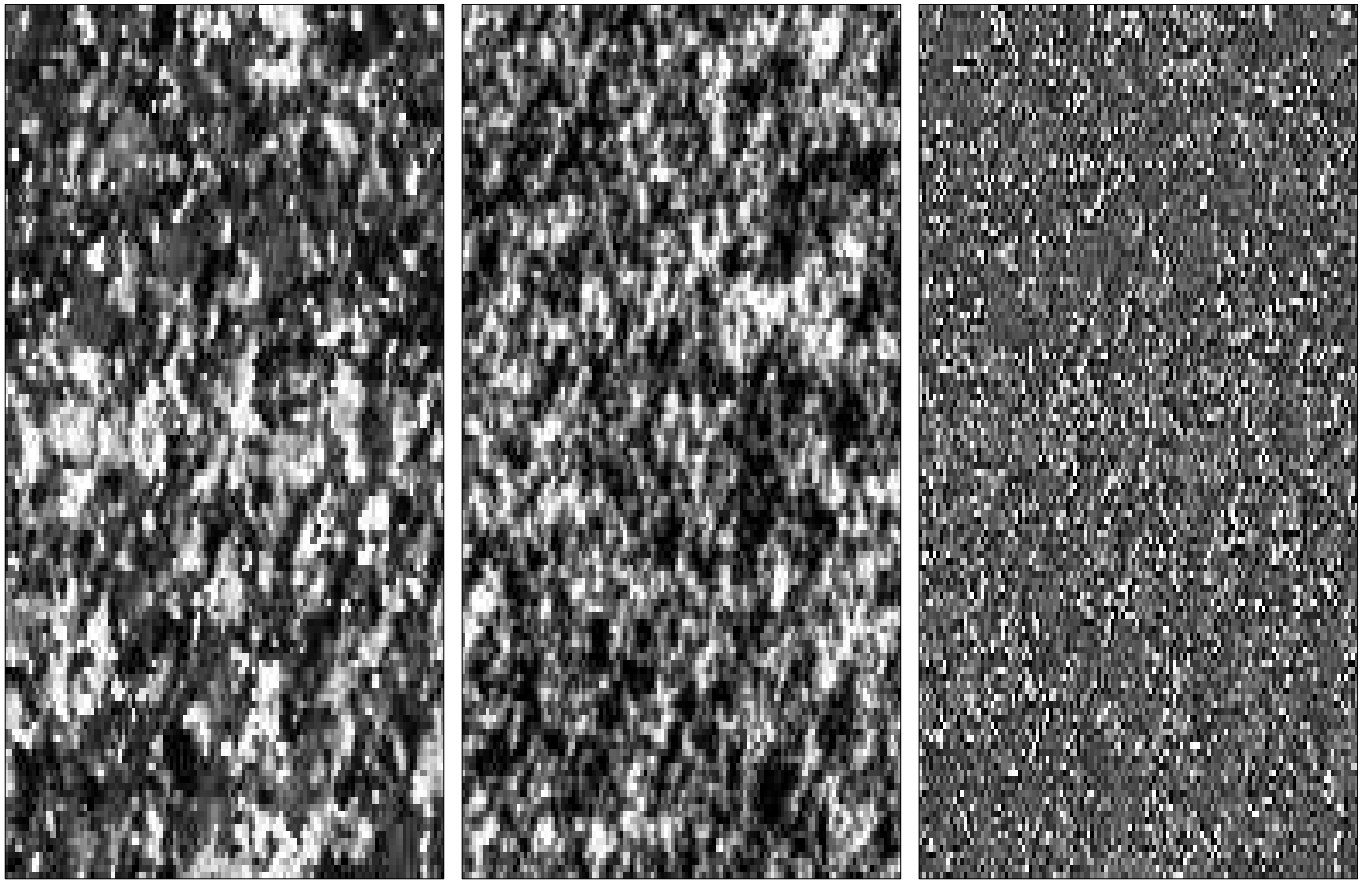

Figure 1: Data as scanned granite, synthesized from the PEF estimated on the data, residual. granite [NR]

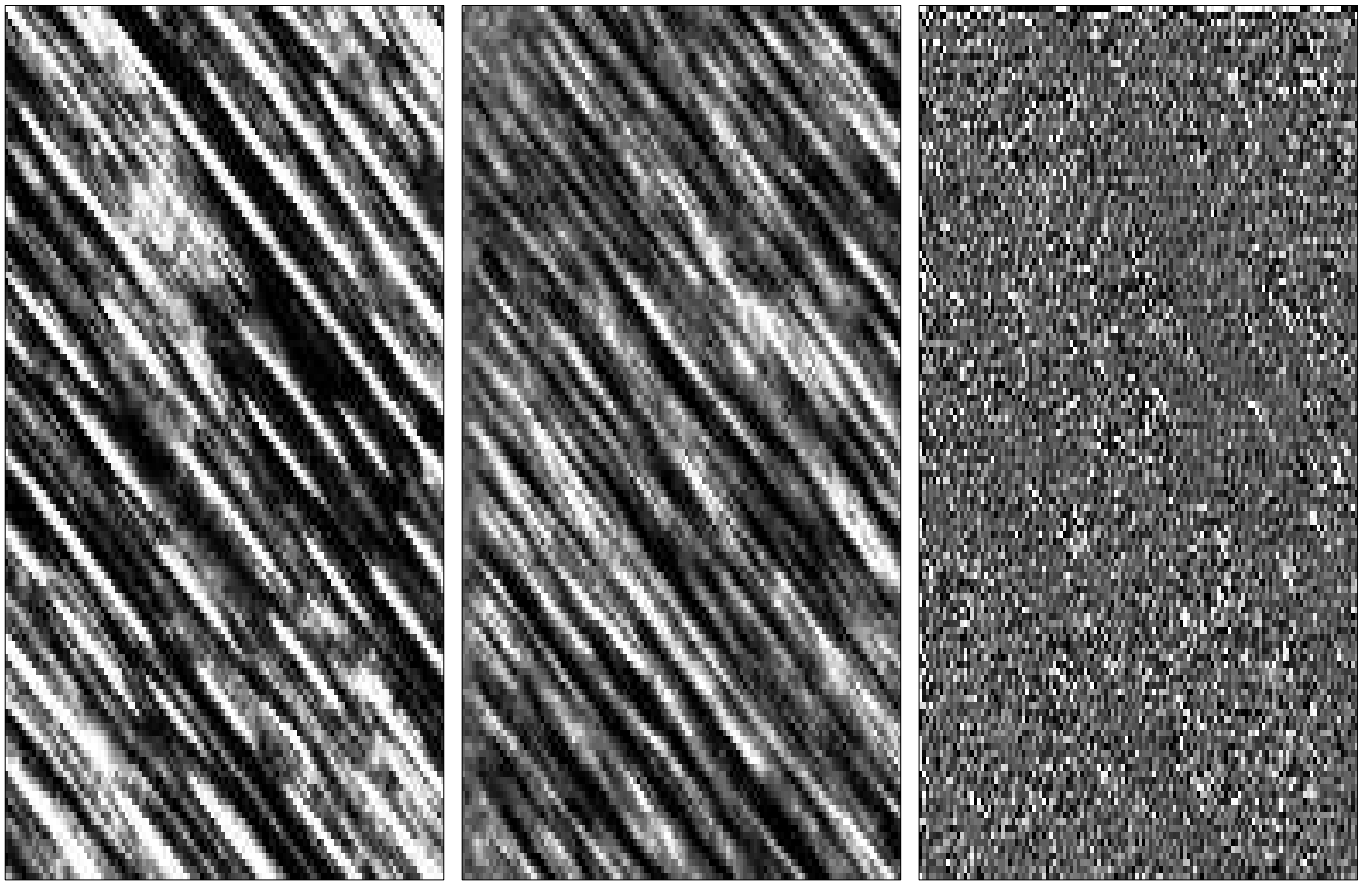

Figure 2: Data as wood grain, synthesized data, residual wood [NR]

shape

$$\begin{matrix} -1 & . \\ . & . \\ . & 1 \end{matrix} \tag{4}$$

would destroy an event or plane that aligned with the non-zero terms, and

$$\begin{matrix} . & 1 \\ -1 & . \end{matrix} \tag{5}$$

takes care of anything dipping in the opposite direction. By convolving the two, we can eliminate both dips simultaneously with something of the shape

$$\begin{matrix} . & -1 & . \\ 1 & . & . \\ . & . & 1 \\ . & -1 & . \end{matrix} \tag{6}$$

Thus, a PEF can generally described by a series of *nxm* coefficients following the leading 1 such as

$$\begin{matrix} h & c & \cdot \\ i & d & \cdot \\ j & e & 1 \\ k & f & a \\ l & g & b \end{matrix} \tag{7}$$

All the PEFs in these examples are 10x 10.

4. Interpolation

By training a PEF on data that are irregularly sampled, the PEF can be used as a regularization operator in an least-squares inversion problem to interpolate data. Minimizing the objective function

$$\min Q(\mathbf{m}) = ||\mathbf{Lm} - \mathbf{d}|| + \epsilon^2 ||\mathbf{Am}|| \tag{8}$$

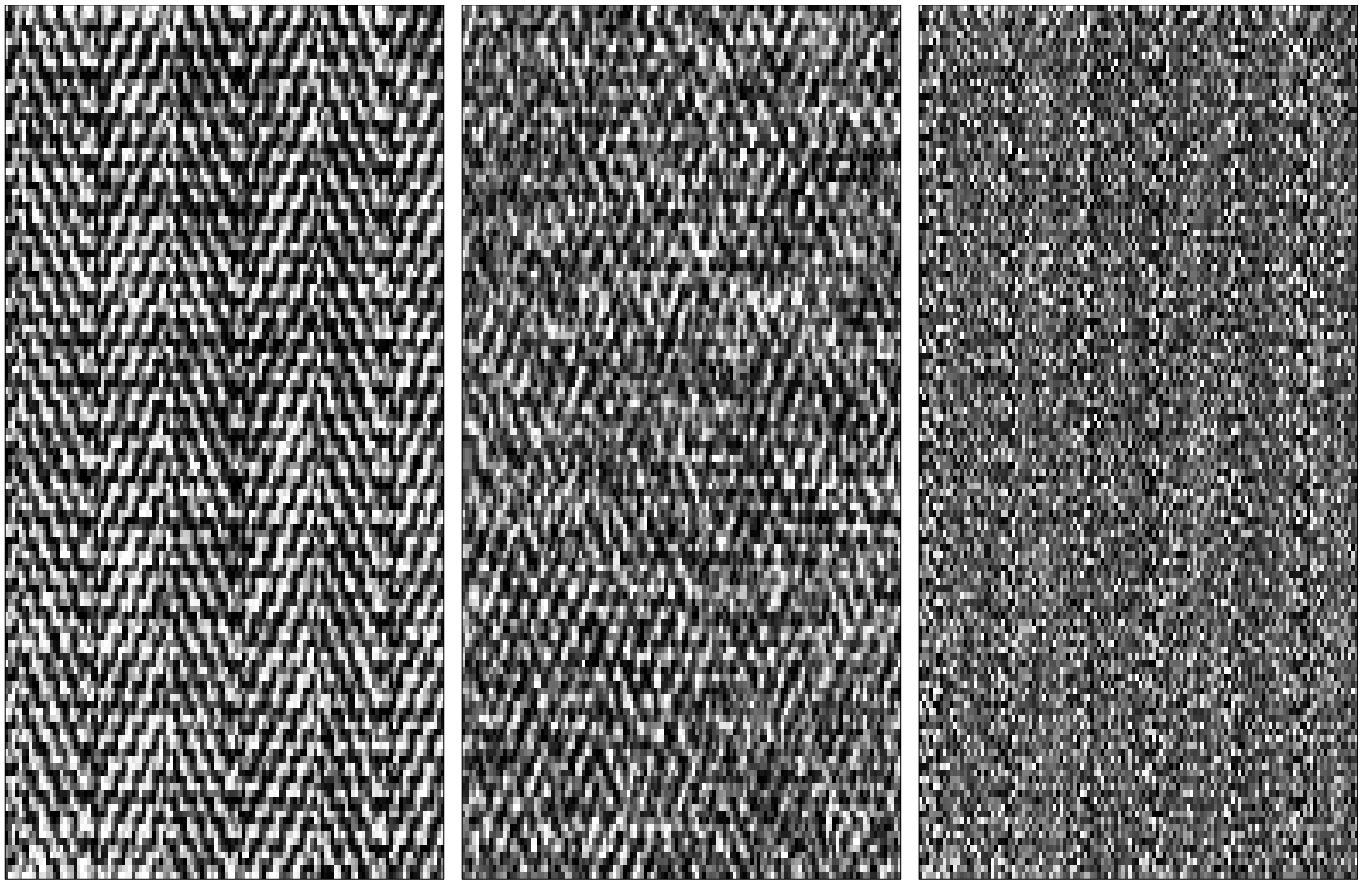

Figure 3: While the PEF did a good job synthesizing granite and wood, the sharp localization of the herringbone pattern leads to problems. Implementation over patches would give a perfect result. herr [NR]

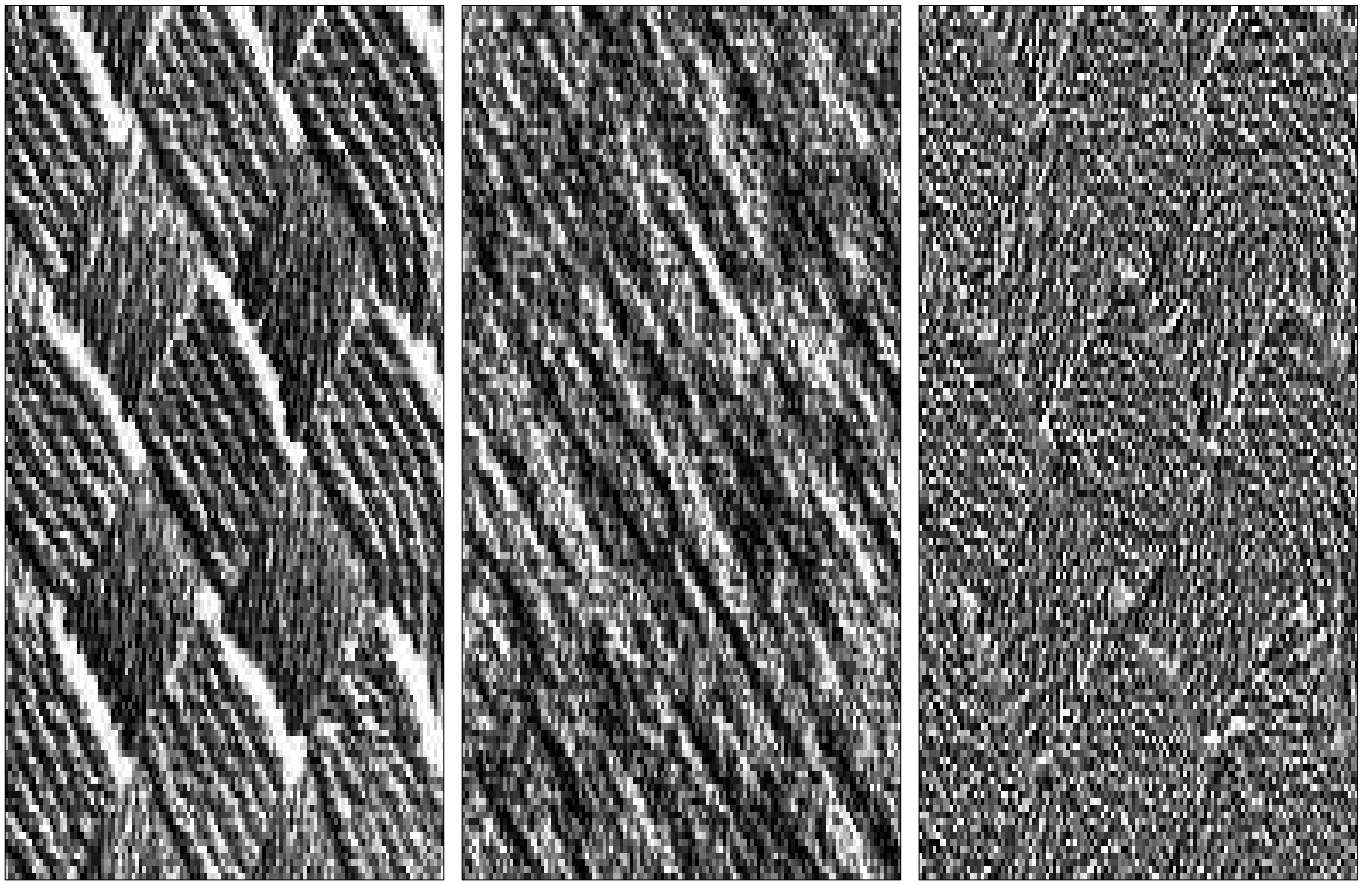

Figure 4: Weave once again fails due to non-stationary construction. basket [NR]

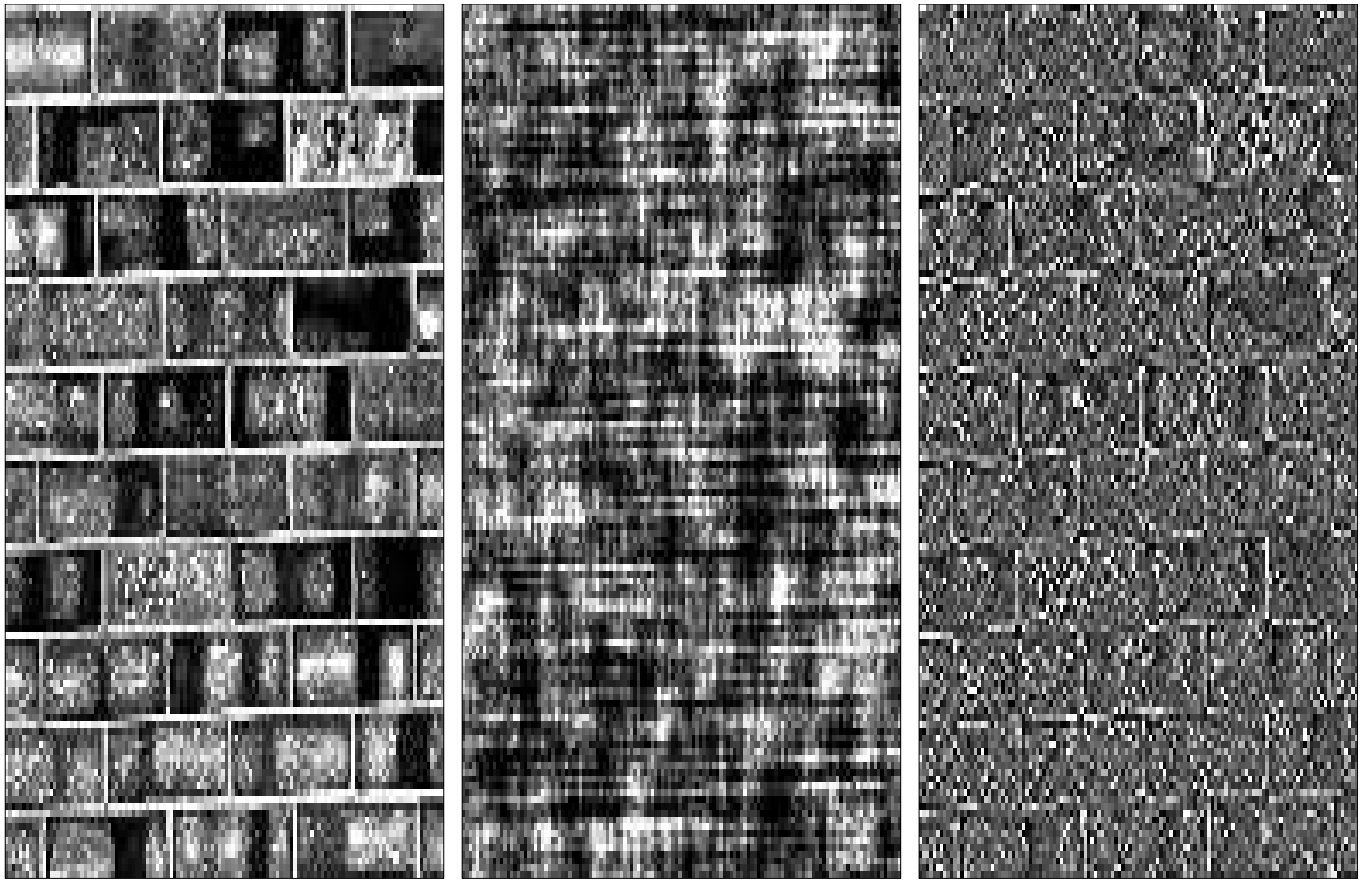

Figure 5: Edges have been generously spread around the result rather than localized. brick [NR]

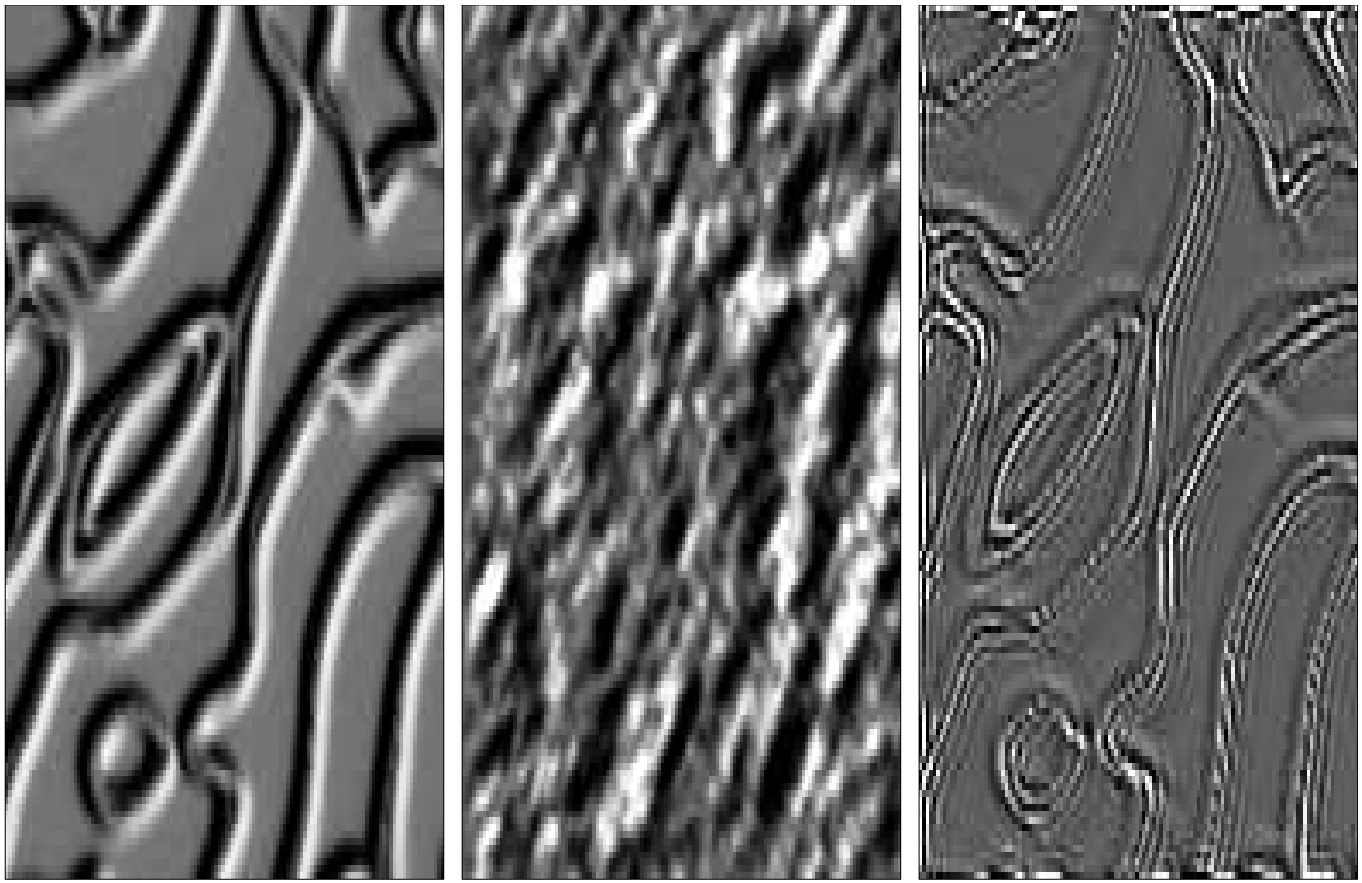

Figure 6: Riverine pattern very difficult as all dips are present in the data, and captured by the PEF, though they should be applied much more sparingly. ridges [NR]

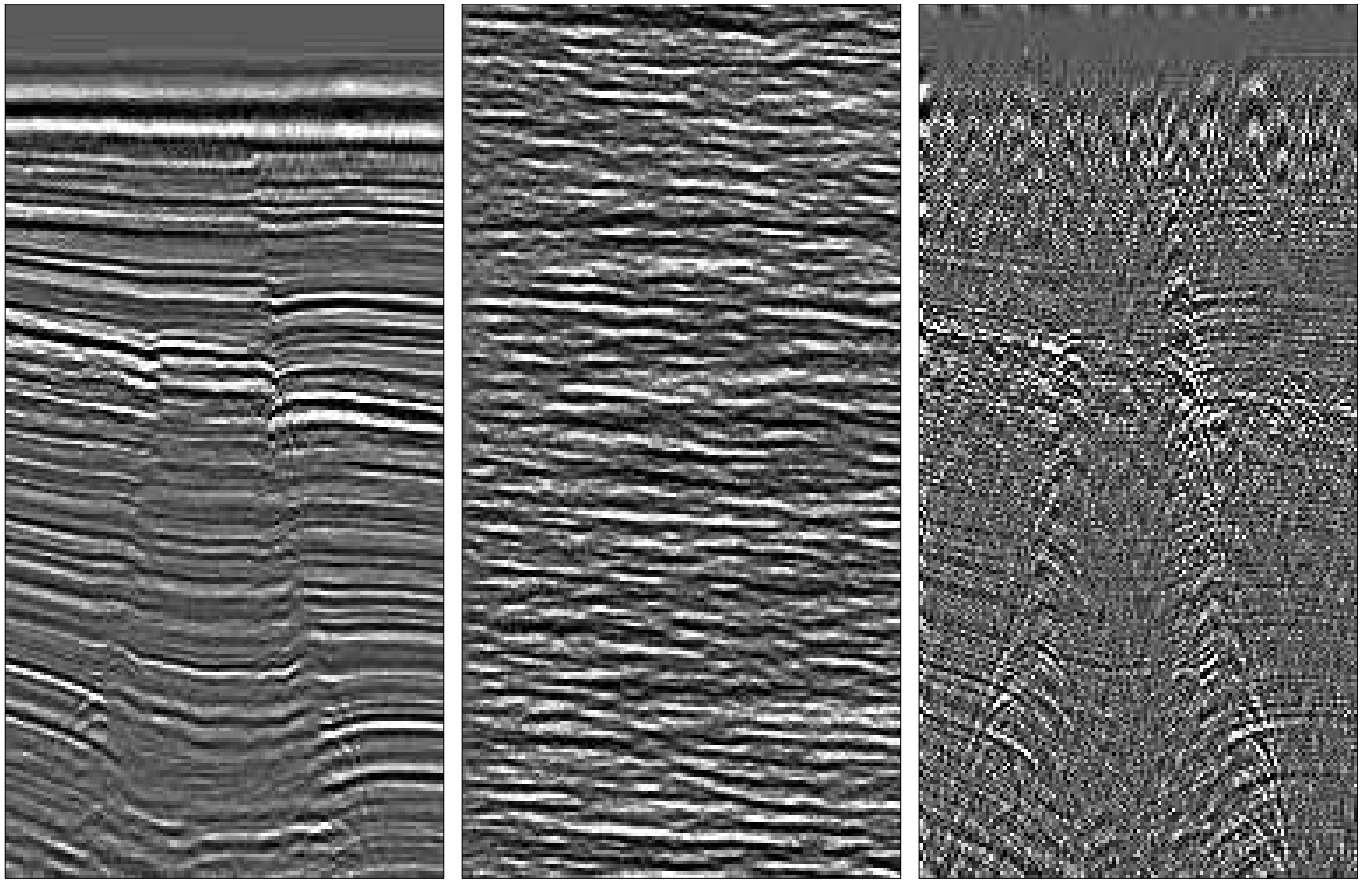

Figure 7: This time, the synthesized data does well predicting the repeating layers, while the residual highlights discontinuities. WGstack [NR]

where L is a simple masking operator (diagonal matrix with ones at known data locations), A is the PEF estimated on the data d, and the output model is m.

Bob Clapp at the Stanford Exploration Project has extended the regularization to incorporate an initial regularization model result that we will seed with random noise, n

$$\min Q(\mathbf{m}) = ||\mathbf{Lm} - \mathbf{d}|| + \epsilon^{\mathbf{2}} ||\mathbf{Am} - \sigma_{\mathbf{d}}\mathbf{n}|| \tag{9}$$

where σ_d is the variance of the data at measured locations to appropriately scale the random numbers.

When we recall the previous figures of the synthesized data, this result makes sense. Where data values exist, they are mapped directly to the model space, while elsewhere, we provide synthesized data that match the spectrum and hopefully structure of the data. This output is equivalent to results utilizing geostatistical realizations while operating in a least squares framework. Importantly, if many realizations are summed (beginning with different random number matrices n) the result is identicle to the solving the interpolation problem without the additional regularization term.

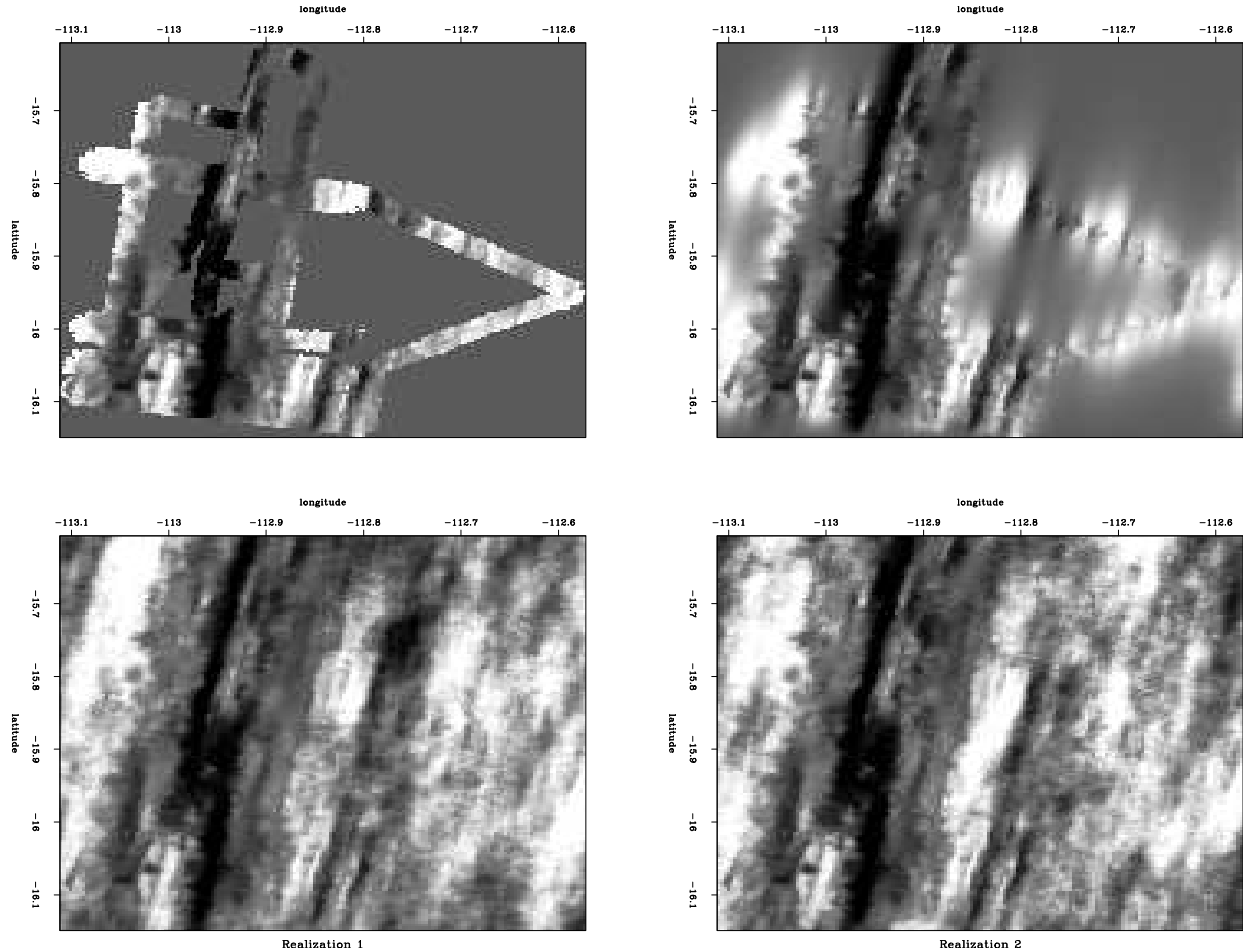

Figure 8: By seeding the regularization expression of the least squares inversion for interpolating the model space, the edges of the model will carry the texture of the data rather than tending to zero. bobsea [NR]

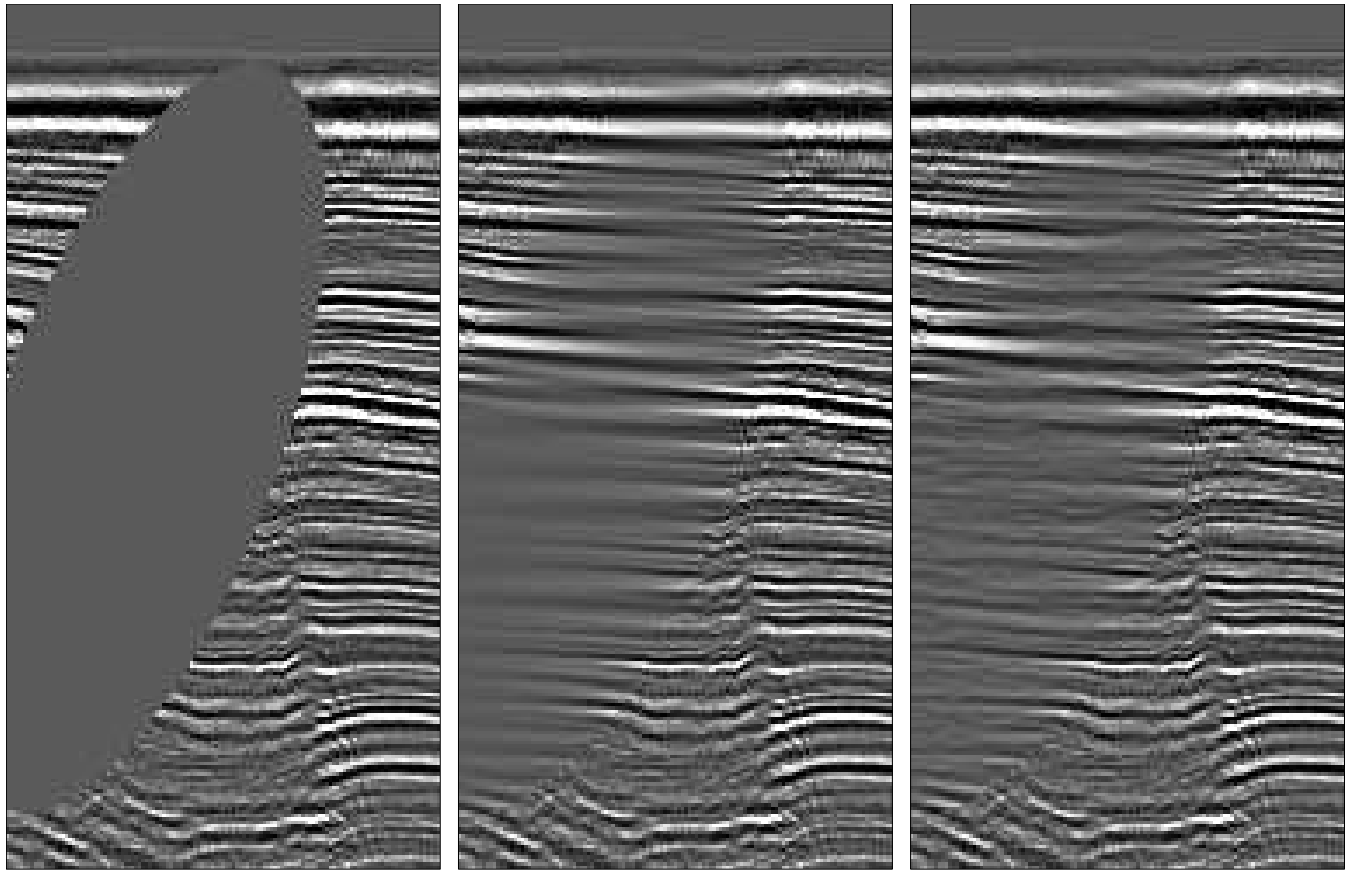

Figure 9: Cutting a huge hole in seismic data, interpolating with a PEF, interpolating after adding random noise to the regularization. WGstack-hole-fillr [NR,M]

5. References

Claerbout, J.F. , *Image estimation by example*
http://sepwww.stanford.edu/sep/prof/toc_html/gee/toc_html/index.html

Claerbout, J.F. , *Fundamentals of Geophysical Data Processing*
http://sepwww.stanford.edu/sep/prof/toc_html/fgdp/toc_html/index.html