F-xy eigen noise suppression

Stewart R. Trickett – Kernel Technologies Inc.

CSEG Geophysics 2002

Summary

F-xy eigen filtering performs random noise suppression on stacked 3-D seismic volumes using eigenimage analysis along constant-frequency slices. The method performs equally well on flat or dipping events and is independent of many x- and y-consistent effects. The method gives results comparable to f-xy prediction filtering but with far fewer calculations.

Introduction

Andrews and Patterson (1976) demonstrated how the singular value decomposition, or SVD, can be used for noise suppression and lossy data compression of digital images. The SVD is defined as (Golub and Van Loan, 1996)

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H.$$

We will consider $\mathbf{A}$ to be a square matrix of dimension *n*, but rectangular matrices are handled just as easily. $\mathbf{U}$ and $\mathbf{V}$ are *n*-by-*n* unitary matrices whose column vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are the left and right eigenvectors of $\mathbf{A}$, respectively. $\mathbf{\Sigma}$ is a real diagonal matrix whose diagonal elements σ_i, known as the singular values, are ordered such that

$$\sigma_1 \geq \sigma_2 \geq \Lambda \geq \sigma_n \geq 0.$$

It is easy to show

$$\mathbf{A} = \mathbf{I}_1 + \mathbf{I}_2 + \Lambda + \mathbf{I}_n, \quad \mathbf{I}_i = \sigma_i \mathbf{u}_i \mathbf{v}_i^H.$$

The matrix $\mathbf{I}_i$ is referred to as the i'th weighted eigenimage. For $k \leq n$ define $F_k(\mathbf{A})$, the k'th partial sum of $\mathbf{A}$, as

$$F_k(\mathbf{A}) = \mathbf{I}_1 + \mathbf{I}_2 + \Lambda + \mathbf{I}_k.$$

This is known as the truncated-SVD, or reduced-rank, approximation of $\mathbf{A}$. It is optimal in the sense that it solves for $\mathbf{M}$

$$\min_{\substack{\mathbf{M} \in C^{n \times n} \\ \text{rank}(\mathbf{M}) \leq k}} \|\mathbf{M} - \mathbf{A}\|$$

where $\|\cdot\|$ can be either the Frobenius or matrix 2-norm. We are not restricted to integer values of *k*. For instance, we can define

$$F_{2.5}(\mathbf{A}) = \mathbf{I}_1 + \mathbf{I}_2 + .5\,\mathbf{I}_3.$$

$F_k(\mathbf{A})$ becomes an increasingly better approximation to $\mathbf{A}$ as *k* increases, until finally $F_n(\mathbf{A}) = \mathbf{A}$. We generally, however, require only a few eigenimages to generate a reasonable image (Figure 1).

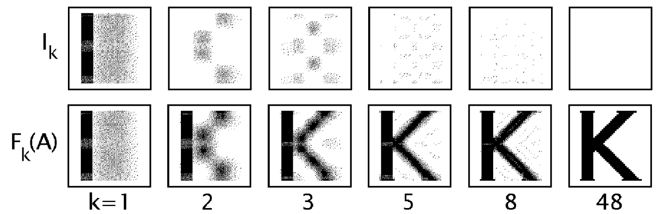

Figure 1: Eigenimages (top) and partial sums (bottom) of a 48x48 pixel image of the letter K. Only a few of the eigenimages are needed to generate a reasonable image. Summing all 48 eigenimages produces the original.

Many noise suppression schemes transform the data into a domain where signal and noise map onto separate regions. Eigen filtering is similar – it assumes that coherent energy maps onto the first few eigenimages, and that incoherent energy maps onto the remainder (or at least is more evenly distributed.) Figure 2 demonstrates this for a noisy letter T.

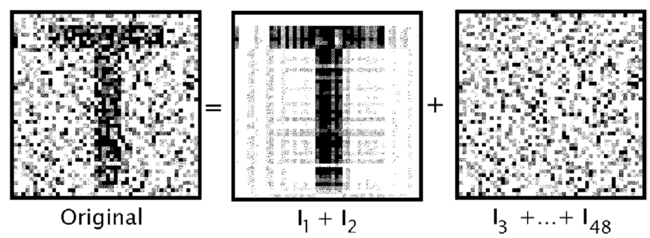

Figure 2: A 48x48 pixel image of a noisy letter T decomposed into the sum of its first 2 weighted eigenimages, containing most of the signal, and the sum of the remaining 46 eigenimages, containing most of the noise.

Ulrych, Freire, and Siston (1988) investigated a variety of seismic applications for eigenimage analysis, including noise suppression, dip filtering, separation of wavefields in vertical seismic profiles, and residual statics corrections. Although their noise suppression technique was not described in detail, it is (apparently) in the t-x domain and probably does not give satisfactory results for dipping reflectors.

Methodology

Eigenimage analysis can be adapted to suppress noise in stacked 3-D volumes of seismic traces. Start with an *n*-by-*n* grid of stacked trace (typical dimensions might be 20 by 20.) The method, which I call *f-xy eigen filtering*, is as follows:

Take the DFT of each trace.

For each frequency…

Form the *n*-by-*n* complex-valued matrix $\mathbf{A}$ from the DFT value of each trace.

Calculate $F_k(\mathbf{A})$ for some small value of *k*.

Replace the trace DFT values with the $F_k(\mathbf{A})$ values.

Take the inverse DFT of each trace.

The amount of attenuated noise can be increased by increasing the grid size *n* and most importantly decreasing *k*, the number of eigenimages summed in (Figure 3.) By doing so, however, we also increase the chance of distorting the coherent signal. In practise typical values for *k* are 1 (harsh), 2 (strong), and 3 (moderate).

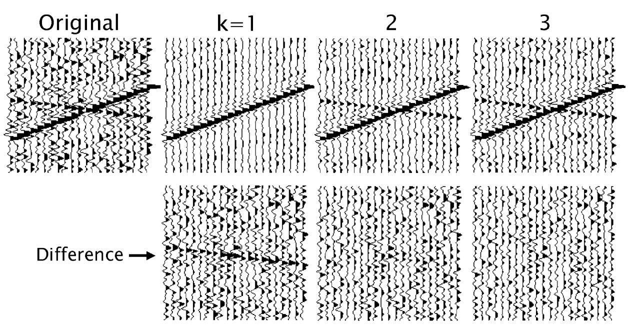

Figure 3: Inline slice through a noisy artificial volume showing the effects of eigenimage filtering with different values of *k*. Filtering becomes harsher as *k* decreases. Although *k* = 1 has removed most of the noise, the difference plot (the removed energy) shows it has also removed the weaker event. The result for *k* = 2 is more acceptable.

Properties

The SVD is well understood, allowing us to establish a number of results about eigen filtering (assume integer *k*.) They justify us working in the f-xy domain since all but the last property do not generally hold when working on, for example, constant-time slices.

Exactness Property: If a noiseless seismic section contains no more than k dips then $F_k(\mathbf{A}) = \mathbf{A}$.

In other words, eigen filtering does nothing to noiseless seismic data with a restricted number of dips (Figure 4.) The exactness property is the critical result, allowing us to consider eigen filtering even for structured data. A similar property was shown by Canales (1984) for f-x prediction filtering.

Filtering Property: If a noiseless seismic section contains no more than k dips, and then has x- and y-consistent filters applied, then $F_k(\mathbf{A}) = \mathbf{A}$.

Statics Property: $F_k(\mathbf{A})$ *is independent of x- and y-consistent statics*.

This means we can apply x- and y-consistent statics, eigen filter the data, correct the statics, and get the same thing as if we had filtered the data without statics. The previous two properties suggest that eigen filtering may be useful for removing noise from prestack data before deconvolution or statics.

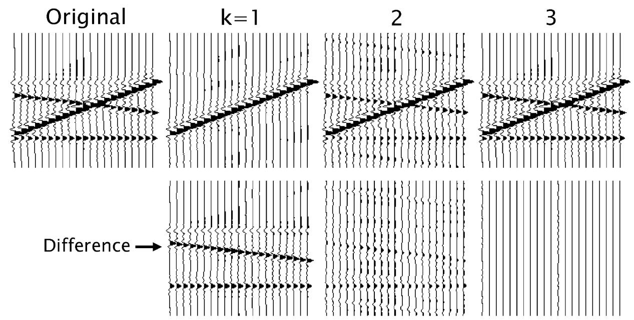

Figure 4: The exactness property. Shown is an inline slice through an artificial volume with four events but only three distinct dips. Eigenimage filtering with *k*=3 does nothing to the original data since the number of eigenimages equals the number of distinct dips.

Ordering Property: $F_k(\mathbf{A})$ *is independent of the x and y ordering of the matrix*.

This means we can reorder the matrix in an x- and y-consistent manner, eigen filter the data, undo the reordering, and get the same thing as if we had filtered the matrix directly. It suggests that eigen filtering behaves well at the grid boundaries since from the filter's point of view there are no boundaries.

Projection Property: $F_k(F_k(\mathbf{A})) = F_k(\mathbf{A})$.

In other words, it is no good applying eigen filtering twice in a row since the second pass does nothing. Mathematically this is because $F_k(\mathbf{A})$ is an orthogonal projection onto the set of *n*-by-*n* matrices of rank *k*.

Fast Approximations

Calculating the SVD is expensive (run time is comparable to f-xy prediction filtering using typical parameters for both), and most of the generated information is ignored. There is, however, a low-cost alternative. Lanczos bidiagonalization (Golub and Van Loan, 1996) consists of decomposing a matrix $\mathbf{A}$ into the form

$$\mathbf{A} = \mathbf{P}\mathbf{B}\mathbf{Q}^H$$

where $\mathbf{P}$ and $\mathbf{Q}$ are unitary matrices and $\mathbf{B}$ is a real bidiagonal matrix of the form

$$\mathbf{B} = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \Lambda & \Lambda & \\ & & & \alpha_{n-1} & \beta_{n-1} \\ & & & & \alpha_n \end{pmatrix}.$$

We need not carry this out to completion. By halting after the calculation of the first k columns of $\mathbf{P}$ and $\mathbf{Q}$ (call them $\mathbf{P}_k$ and $\mathbf{Q}_k$) and the first *k* rows and columns of $\mathbf{B}$ (call it $\mathbf{B}_k$), we can define

$$L_k(\mathbf{A}) = \mathbf{P}_k \mathbf{B}_k \mathbf{Q}_k^H .$$

Simon and Zha (2000) show that $L_k(\mathbf{A})$ is an approximation to $F_k(\mathbf{A})$, and for small *k* can be calculated at a fraction of the cost. Since it is an approximation, some deterioration in results is expected. On real data the difference between the original and Lanczos methods is usually, but not always, minor.

To address the occassional inaccuracies of Lanczos filtering I developed a hybrid method. The first step calculates the Lanczos approximation out to *k* + *r* rather than *k* steps, where *r* is perhaps 2 or 3, giving

$$L_{k+r}(\mathbf{A}) = \mathbf{P}_{k+r}\mathbf{B}_{k+r}\mathbf{Q}_{k+r}^{H}.$$

This is an approximation to $F_{k+r}(\mathbf{A})$. For an approximation to $F_k(\mathbf{A})$ apply $F_k(\cdot)$ to the central matrix only, giving

$$D_{k,r}(\mathbf{A}) = \mathbf{P}_{k+r}F_k(\mathbf{B}_{k+r})\mathbf{Q}_{k+r}^{H}.$$

I call this the "double-truncated SVD". Calculating $F_k(\mathbf{B}_{k+r})$ requires an SVD. It is still, however, much faster than calculating $F_k(\mathbf{A})$ because $\mathbf{B}_{k+r}$ has much smaller dimension than $\mathbf{A}$, is real rather complex, and is bidiagonal rather than dense. These effects compound so that the SVD takes only a small part of the total run time. With *r* = 2, double-truncated SVD runs about five times faster than the original method but gives essentially the same results.

Data Results and Discussion

Figure 5 compares eigen filtering and its natural rival f-xy prediction filtering (Chase, 1992), on a flat, moderately noisy 3-D section. For the selected parameters there is little to choose between the two methods. The difference plots (that is, the removed noise) show that neither has removed much coherent energy.

It is difficult to compare different noise suppression methods since so much depends on implementation and parameter selection, but generally speaking eigen filtering appears milder than f-xy prediction filtering for a given amount of signal preservation. Its fast speed and excellent behaviour at the grid boundaries, however, may make it the preferred choice for certain situations.

For very strong noise suppression, eigen filtering can be applied after f-xy prediction or projection filtering (the other way around is not as successful). One reason these might work well together is that they use completely different noise-suppression mechanisms – that is, separation versus prediction. The extra cost over fx-prediction alone is minor since eigen filtering is much faster.

Concerning future work, the speed of eigen filtering, combined with its ability to absorb x- and y-consistent multipliers, suggests it may be useful for noise suppression of prestack 2-D data prior to deconvolution or statics. Eigenimage decomposition can also be used for data compression (Andrews and Patterson, 1976.) If a stacked 3-D volume has had eigen noise suppression performed, eigen compression in the f-xy domain can be nearly lossless due to the projection property.

Acknowledgements

I would like to thank Noranda Inc. and Aur Resources Inc. for permission to show their seismic data. Thanks also to Kelman Technologies Inc. for supporting this research.

References

Andrews, H. C., and Patterson, C. L., 1976, Outer product expansions and their uses in digital image processing: IEEE Transactions on Computers, **25**, no. 2, 140-148.

Canales, L. L., 1984, Random noise reduction: 54th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session: S10.1.

Chase, M. K., 1992, Random noise reduction by 3-D spatial prediction filtering: 62nd Annual Internat. Mtg., Soc. Expl. Geophys., Expaneded Abstracts, 1152-1153.

Golub, G. H., and Van Loan, C. F., 1996, Matrix Computations, 3rd ed.:, John Hopkins University Press.

Simon, H. D., and Zha, H., 2000, Low-rank matrix approximation using the Lanczos bidiagonalization process with applications: SIAM J. Scientific Computing, **21**, no. 6, 2257-2274

Ulrych, T. J., Freire, S., and Siston, P., 1988, Eigenimage processing of seismic sections: 58th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session: S22.4.

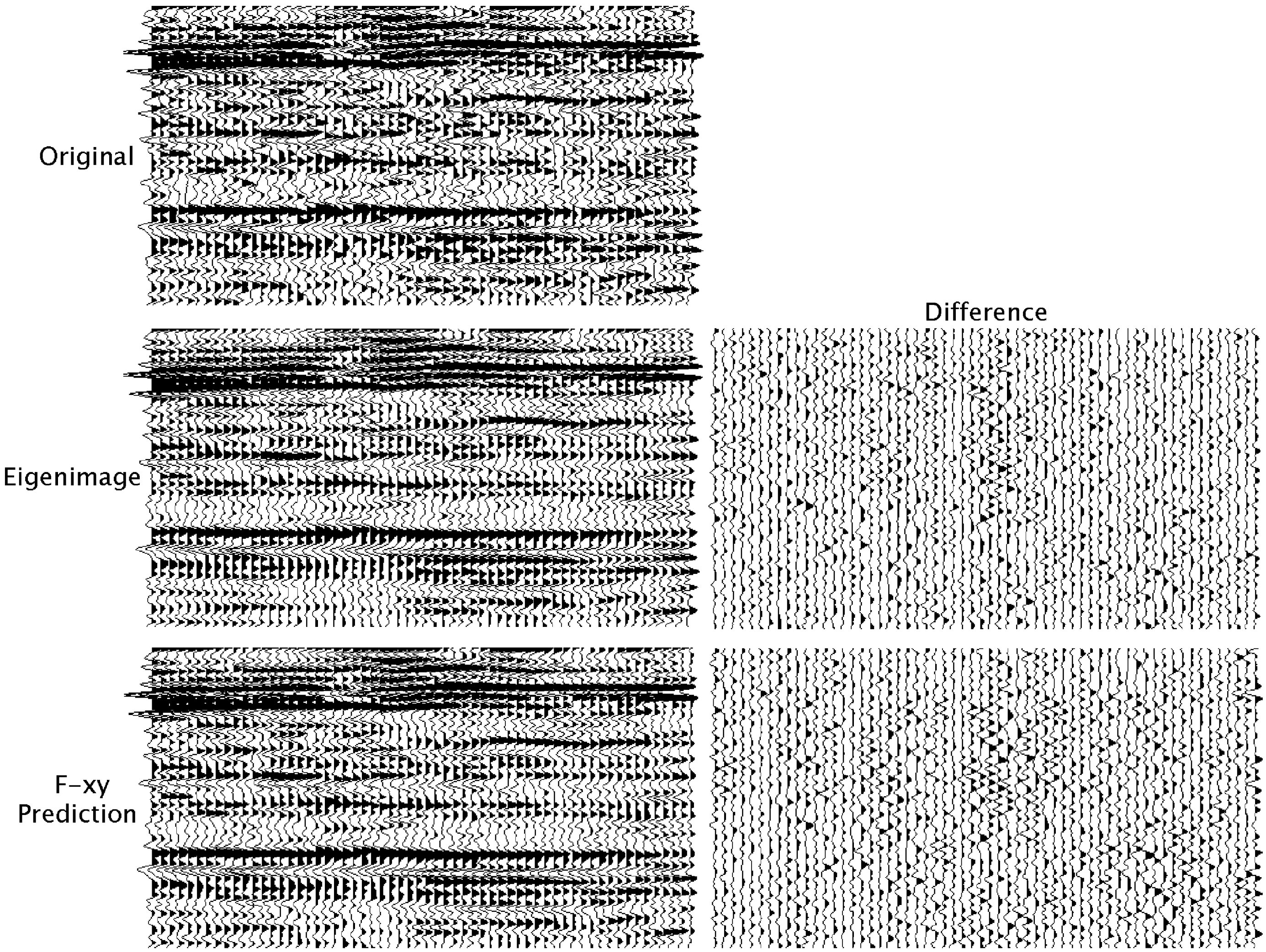

Figure 5: Real data example along an in-line slice comparing eigenimage filtering with k = 2, and f-xy prediction filtering with a 5x5 operator. There is little to choose between them using these parameters. Neither has remove much coherent signal.