# Rank-Reduction Filters for 3 and 4 Spatial Dimensions

*Alex Falkovskiy*

*Rocky Mountain Seismic Software, Calgary, AB, Canada*

## Summary

Discussed different implementations of Singular Spectrum Analysis (SSA) and rank-reduction technique for random noise attenuation on seismic data. Compared application of conventional FX filter, Rank Reduction 3D and Rank Reduction 4D filters. Calculated time complexity for windowed implementation of SSA algorithm on seismic data.

## Introduction

Attenuation of various noise types has always been a challenging problem in seismic exploration. Over recent ten to fifteen years, a Singular Spectrum Analysis (SSA) method has been successfully implemented to seismic data for noise reduction (Trickett, 2003, 2009; Sacchi, 2009). These applications of SSA on seismic data evolved from earlier works on filtering methods outside seismic known as Cadzow filter (Cadzow, 1988), or the Caterpillar method (Golyandina et. al., 2001, 2005).

Modification of the standard SSA method for application to seismic data proposed by Tricked comprised application of SSA on complex Hankel matrices composed from Fourier transformed traces, and it was applied separately on each frequency slice.

In this paper we discuss technique for creating nested Hankel matrices for two, three and four spatial dimensions, and assess time complexity for these method modifications. Though it involves more computations, fourth dimension (shot lines) is inevitable in 3D processing.

## Theory and Method

Rank-reduction filters are applied on frequency slices of input seismic data after Fourier transform is applied to each trace. For each frequency slice a complex, input Hankel matrix A is created (as described below) and decomposed into a sum of eigenimages using SVD:

$$A = I_1 + I_2 + .. I_n$$

Each eigenimage $I_j$ in this sum is a matrix of rank 1, and the strongest components of the signal are in first eigenimages. At such representation, rank reduction filtering means leaving in the sum only the first few $r$ eigenimages $A_r = I_1 + \cdots + I_r$ that are supposed to represent a signal, and zeroing all other eigenimages with higher numbers, that are supposed to be noise.

There are different methods for construction of or A. For a 4D case, the input matrix A is a complex multi-level block Hankel matrix created from a several sets of nested Hankel matrices.

**1. Rank-reduction 2D (applied shot gather or 2D stack)**

For *n* consecutive traces in a shot gather or 2D stack, Hankel or Cadzow matrix is created as

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ a_2 & a_3 & \ldots & a_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_m & a_{m+1} & \cdots & a_n \end{bmatrix},$$

where $a_1, a_2, \ldots a_n$ are Fourier coefficients of these traces for one frequency slice, and $\boldsymbol{m} \approx \boldsymbol{n/2}$ .

**2. Rank-reduction 3D (applied on one shot line or 3D stack)**

For *p* shots on one shot line, Hankel block-matrix is created as a nested matrix

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_m \\ A_2 & A_3 & \ldots & A_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ A_m & A_{m+1} & \cdots & A_p \end{bmatrix},$$

where $A_i = \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{im} \\ a_{i2} & a_{i3} & \ldots & a_{i,m+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{im} & a_{i,m+1} & \cdots & a_{in} \end{bmatrix}$ is a Hankel matrix for *i*-th shot, $a_{i1}, a_{i2}, \ldots a_{in}$ are Fourier coefficients of these traces for one frequency slice, and $\boldsymbol{m} \approx \boldsymbol{n/2}$ .

**3. Rank-reduction 4D (applied on 3D prestack data: several shot lines)**

For *s* shot lines, p shots on one shot line, and *n* receivers on each for each shot line, a combined block Hankel matrix is created as a two-level nested matrix

$$\begin{bmatrix} B_1 & B_2 & \cdots & B_m \\ B_2 & B_3 & \ldots & B_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ B_m & B_{m+1} & \cdots & B_s \end{bmatrix},$$

where $B_j = \begin{bmatrix} A_{1,j} & A_{2,j} & \cdots & A_{m,j} \\ A_{2.j} & A_{3,j} & \ldots & A_{m+1,j} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,j} & A_{m+1,j} & \cdots & A_{p,j} \end{bmatrix}$, $A_{ij} = \begin{bmatrix} a^j_{i,1} & a^j_{i,2} & \cdots & a^j_{i,m} \\ a^j_{i,2} & a^j_{i,3} & \ldots & a^j_{i,m+1} \\ \vdots & \vdots & \ddots & \vdots \\ a^j_{i,m} & a^j_{i,m+1} & \cdots & a^j_{i,n} \end{bmatrix}$,

where $B_j$ is a block Hankel matrix for *j*-th shot line, where $A_{ij}$ is a block Hankel matrix for *j*-th shot line and *i*-th shot, $a^j_{i,1}, a^j_{i,2}, \ldots a^j_{i,n}$ are Fourier coefficients of these traces for one frequency slice, and $m \approx n/2$ .

## Time Complexity

To estimate time complexity of these algorithms, we assume that time complexity of exact SVD of m x n matrix is $O(\min\{mn^2, m^2n\})$ (Golub and Van Loan, 1989). Since all these matrices are build in a way to be almost square, $\approx n$ , and in this case the complexity of exact SVD is $O\ (n^3)$. Another assumption is that practical implementations of all these methods apply these algorithms in windows of limited number of traces (1-20). Note that all following calculations are performed for one frequency slice.

To estimate time complexity (computational cost) of Rank Reduction 2D, let us assume that we have *n* traces, and the algorithm is applied not on all traces simultaneously, but in windows of $l$ traces each. (For simplicity we assume that these windows do not overlap). Then, the total number of windows is $\frac{n}{l}$ , and each Hankel matrix for SVD application has dimensions of $\frac{l}{2}$ x $\frac{l}{2}$ , and the time complexity of exact SVD on such matrix is $O\ ((l/2)^3)$. The complexity of this windowed version of this algorithm is $\frac{n}{l} * O\ ((l/2)^3) = O\ (n\, l^2)$. Note that time complexity for windowed version of algorithm is much less than that when this method is applied on the whole set of *n* traces ( compare to $O\ (n^3)$), since $m \ll n$ .

For estimation of computational complexity of Rank Reduction 3D, for simplicity, we assume that the number of receivers $n$ is the same as the number of shots, and the length of window for application of this algorithm is $l$. Than the algorithm will be applied in $\frac{n}{l} \times \frac{n}{l}$ windows with computational cost in each of $O\left(\left(\frac{l}{2} \times \frac{l}{2}\right)^3\right) = O(l^6)$. Total computational cost for Rank Reduction 3D is $\frac{n^2}{l^2}\ O(l^6) = O(n^2 l^4)$.

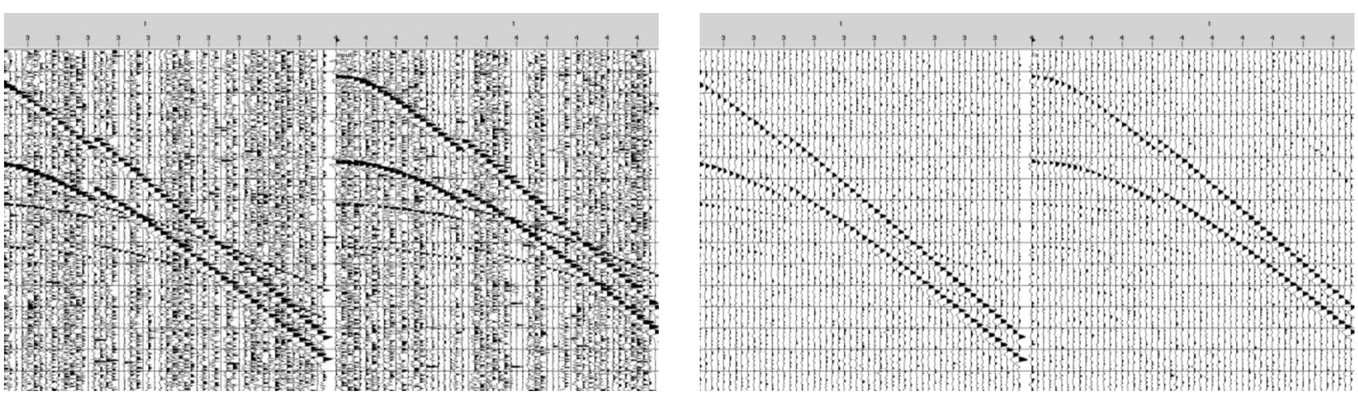
Similarly, the computational complexity for Rank Reduction 4D is $\frac{n^3}{l^3}\ O(l^9) = O(n^3 l^6)$.

Despite the fact that the values of $l^4$ or $l^6$ in these equations may be large, it is a constant coefficient in a windowed version of this algorithm, and for a constant window size, so computational complexity is at most $O(n^3)$. At reasonably small window sizes make application of Rank Reduction 4D is feasible and comparable with migration run time.

## Examples

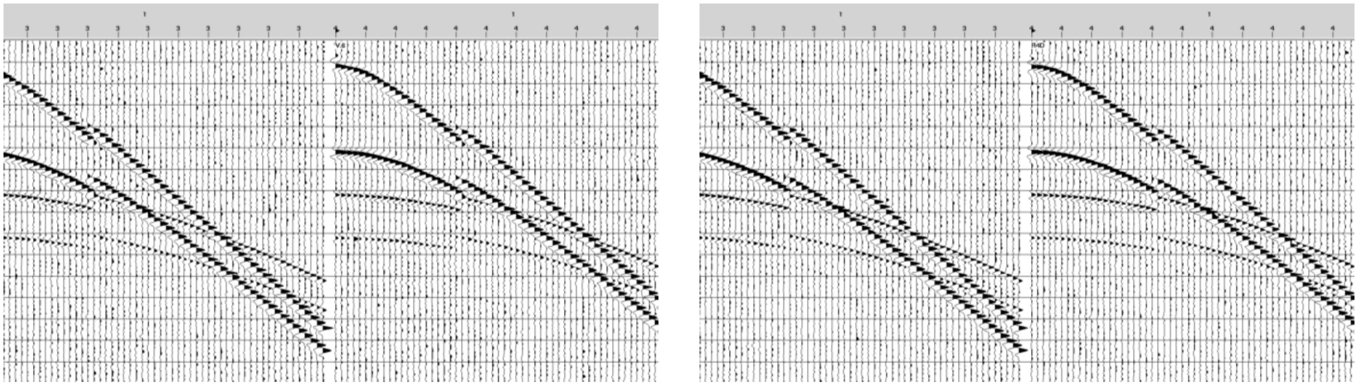
The following are some examples of application of these algorithm on synthetic data.

Figure 1 shows input data: synthetic shot gathers with added random noise. Though noise is removed, the signal amplitude is weaker, and lower weaker event is almost lost.  Figure 2 and 3 compare application of Rank Reduction 3D and Rank Reduction 4D filters. Both filters show better results than traditional FX filter in signal amplitude preservation, but 4D shows better result in preservation of complex feature (blue oval in Figure 3).



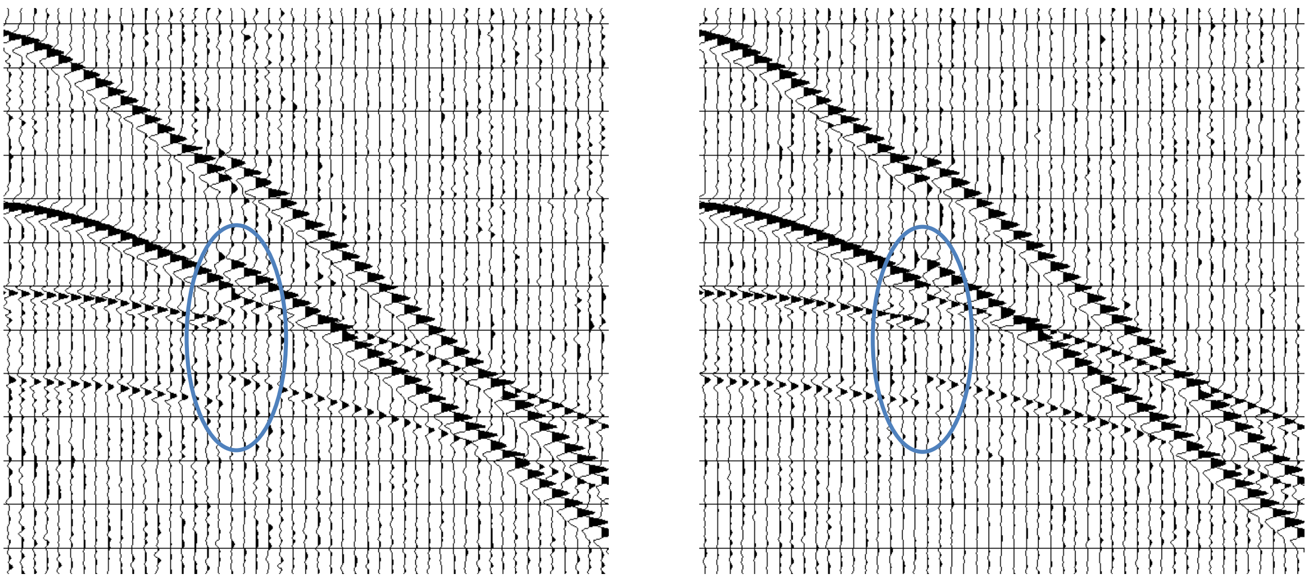
a. Input data  b. FX filter

Figure 1. Input synthetic shot gathers and application of FX filter.



a. Rank Reduction 3D b. Rank Reduction 4D

Figure 2. Application of Rank Reduction 3D and Rank Reduction 4D filters



a. Rank Reduction 3D b. Rank Reduction 4D

Figure 3. Application of Rank Reduction 3D and Rank Reduction 4D filters (enlarged)

## Conclusions

Presented various modifications of rank-reduction filters for 2D, 3D and 4D seismic data. Discussed equation for time complexity for windowed version of this algorithm. Rank reduction SSA filters showed superior results on noise attenuation compared to FX filter, including amplitude preservation. Rank Reduction 4D filter showed better results on complex structure, than 3D version on synthetic shot gathers. Rank Reduction 4D has longer run time but shows much better quality including amplitude preservation and performance in the areas with discontinuities.

**References**

Cadzow, J., 1988, Signal Enhancement – A Composite Property Mapping Algorithm: IEEE Transactions on Acoustics, Speech and Signal Processing, 36, 49-62.

Canales, L.L., 1984, Random Noise Reduction: SEG, Extended Abstracts, 525-527.

Falkovskiy, A., Floreani, E., Schlosser, G. 2011, Application of FX Singular Spectrum Analysis on Structural Data, CSPG CSEG CWLS Convention.

Golub, G.H. and Van Loan, C.F. 1989, Matrix Computations. Johns Hopkins University Press, Baltimore.

Golyandina, N., Nekrutkin, V., and Zhigljavsky, A., 2001, Analysis of Time Series Structure: SSA and Related Techniques: CRC Press

Golyandina, N., Usevich , K., and I.Florinsky, 2007, Filtering of Digital Terrain Models by two dimensional Singular Spectrum: International Journal of Ecology & Development, Vol. 8, No. F07, 81-94.

Sacchi, M. D., 2009, FX Singular spectrum analysis: CSPG CSEG CWLS Convention, Abstracts, 392-395.

Trickett, S. R., 2003, F-xy Eigenimage Noise Suppression, Geophysics, 68, 751-759.

Trickett, S., 2009, Prestack Rank-Reduction-Based Noise Suppression: CSEG Recorder, November,24-31.