CHAPTER 2

NOISE SUPPRESSION THEORY

This chapter reviews the principles of three main noise suppression methods; f-x prediction, 2-D median and SVD filters. These were applied with real seismic and GPR data

2.1 F-x prediction filtering

The f-x prediction filter was proposed by Canales in 1984 based on signal predictability. The prediction used least-squares method or the Wiener filter and apply in f-x domain. The f-x prediction filter is effective in attenuating random noise (Gulunay, 1986). The principles function of f-x prediction filter is similar to predictive deconvolution. The predictive deconvolution attenuates multiples with the predictive lag perform in time *(t)* domain but the predictive lag of f-x prediction filter is operated prediction in distance *(x)* domain.

F-x prediction filter works by transformation using Fourier Transform in the time *(t)* domain to compute the complex frequency series for each trace. Each sample in complex frequency series has both real and imaginary components. The autocorrelation was used to design predictive lag using least-squares method. The convolution between predictive lag with complex numbers of each frequency was filtering result in frequency domain, noises were removed and the data across space were smoothed (Canales, 1984). After that, the filtering result will be by transformation using Inverse Fourier Transform as presented in flowchart Figure 2-1.

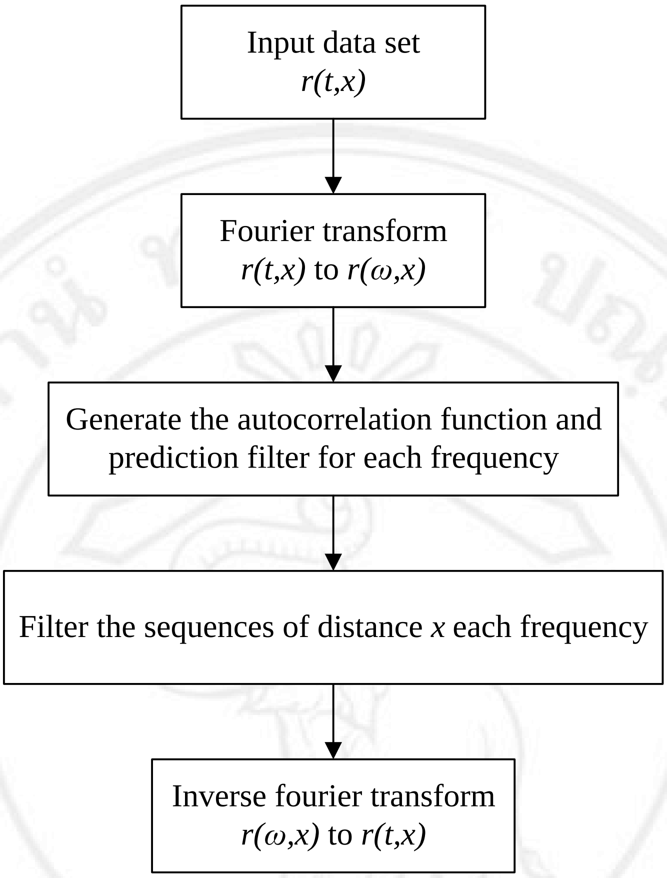

Figure 2-1. Process flowchart for the f-x prediction filter (modified from Harrison, 1990).

2.2 2-D median filtering

The operation of 2-D median filter can be used in data processing to reduce random noise, effective to remove glitches on data and unusual amplitudes as well as enhance discontinuities in the data (Stewart, 1985).

A 2-D median filter design is a window of time-space in matrix of dimensions $M \times N$, where M is samples of trace, N is numbers of trace. Given $C_{M \times N}$ is the window filter size, normally $C_{M \times N}$ is odd.

The 2-D median filter operates by selecting the middle value of a sequence of numbers sorted from smallest to largest in designed window. The sorting can be either across adjacent traces (left hand side to right hand side) or along a given trace (top to bottom). The median was replaced in the middle position of the designed window. The window was slid from first trace until last trace by increment of one trace per one slide in *(x)* direction. When the processing was finished in one *(x)* direction, the window was moved into the first trace by increment of one sample and slide in *(x)* direction again and then the step of window sliding was repeated until last sample of trace.

For example, the Figure 2-2 shows setup red window size of 3 x 3 which the central value is 150. The 9 points in the window are sorted from 115 to 150 with median value 124. The value 150 is than replaced by value 124 and then window shift is shifted in direction sequence with the same window size (Huang et al., 1979).

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

⇩

Neighborhood values:

115,119,120,123,124,125,126,127,150

Median value: 124

⇩

123	125	126	130	140
122	124	126	127	135
118	120	124	125	134
119	115	119	123	133
111	116	110	120	130

Figure 2-2. The calculating the median value of an amplitude value neighborhood (modified from Fisher et al., 2013).

2.3 Singular value decomposition filtering

SVD filter can be applied on seismic and GPR data for suppression both of coherent and random noises. Noises in data sets (i.e. shot gather, CDP gather, NMO corrected CDP gather and stack section) are extracted using an eigenvalue decomposition of the data covariance matrix (Bekara and Van der Benn, 2007).

A data section in a form of matrix *X* has a dimension of *M*×*N*. A row *M* is a trace function of time, whereas a column *N* represents amplitude variations of the distance. The eigenvector u_i of eigenimage describes a wavelet in function of time. The eigenvector v_i of eigenimage describes a wavelet in function of the distance.

$$X = \sum_{i=1}^{r} \sigma_i u_i v_i^T \quad (2\text{-}1)$$

where ^{T} indicates transpose, *r* is the range of *X*, u_i is the i^{th} eigenvector of XX^T, v_i is the i^{th} eigenvector of X^TX. The eigenvalue matrix is a diagonal matrix containing the singular values $diag[\sigma_1, \sigma_2, \ldots, \sigma_r]$. σ_i is the i^{th} singular value of *X*. The term of $u_i v_i^T$ matrices in equation 2-1 was called “eigensection” or “eigenimage” (Andrews and Patterson, 1976).

The matrix *X* of the equation (2-1) depends on the summation since *i* = 1 to *r* range and the range at *r* equals to the number of input data traces. The singular values σ_i were sorted from largest to smallest $(\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r)$ and the summation since *i* = 1 to *r* range of matrix *X* can be written as:

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T \quad (2\text{-}2)$$

This is the form of the full SVD which is used in signal separation of matrix *X*. The filter processing was performed by the choice some part of summation in range of

i = 1 to r in equation (2-2). The determination of the variables range from p to q operated on ranges of $1 < p \leq q < r$.

Ulrych et al., (1988) defined band-pass X_{BP}, low-pass X_{LP} and high-pass X_{HP} eigenimage in terms of the ranges of singular values for signals separation. The SVD band pass filter can be recreated by rejecting highly correlated or highly uncorrelated of traces and is given by

$$X_{BP} = \sum_{i=p}^{q} \sigma_i u_i v_i^T, \qquad 1 < p \leq q < r \qquad (2\text{-}3)$$

The summation for X_{LP} is from p = 1 to q when $q < r$ and for X_{HP} from p to q when q = r. The p and q value were selected from slope of singular value spectrum.

For the example, in Figures 2-3 are illustrated the reformation of a flat event immersed in noise. The singular value spectrum in Figure 2-4 was plotted between relative magnitude and each singular value. The singular value spectrum was an abrupt change and can be divided the slope into 2 parts as follows by p = q = 1 and p=2 to q=32. The high relative magnitude at the 1st singular values was represented the flat event signal. The low relative magnitude was related to the random noise. The SVD filter parameter was p = q = 1. The noise was suppressed by existence of a term in equation (2-2). This term is $\sigma_1 u_1 v_1^T$ with band pass filter of $X_{BP} = \sigma_1 u_1 v_1^T$ in the data. Therefore, p and q were selected in low number of singular values which the random noise was suppressed.

In certain cases, an abrupt change of singular value spectrum is easily recognized. In other cases, the change of singular value spectrum is more gradual and care must be exercised in the choice of the appropriate singular value (Freire and Ulrych 1988).

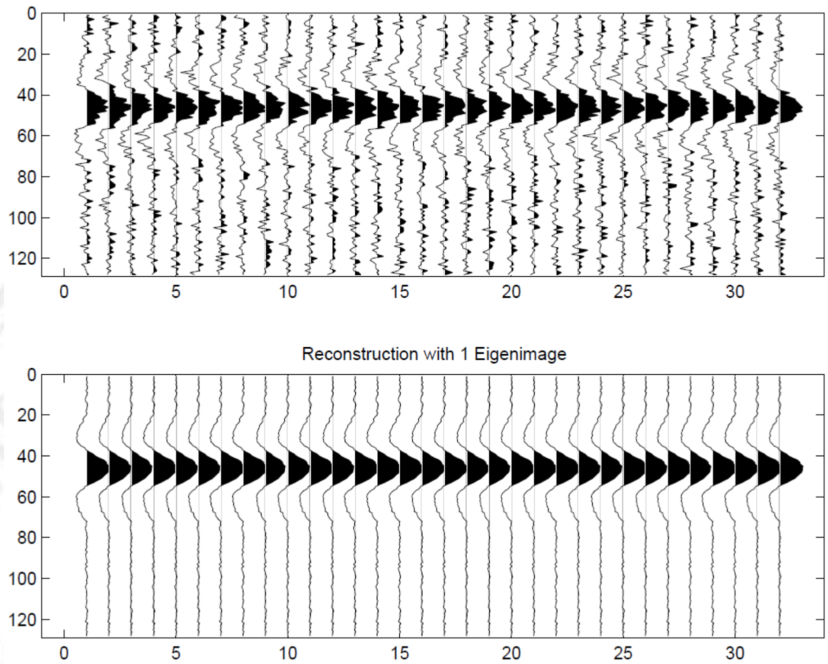

Figure 2-3. A flat event is contaminated with noise and the reformation by $p = q = 1$ (from Signal Analysis and Imaging Group Department of Physics, University of Alberta, Available: http://saig.physics.ualberta.ca/s/sites/default/files/chapter6.pdf. 2013, March 16).

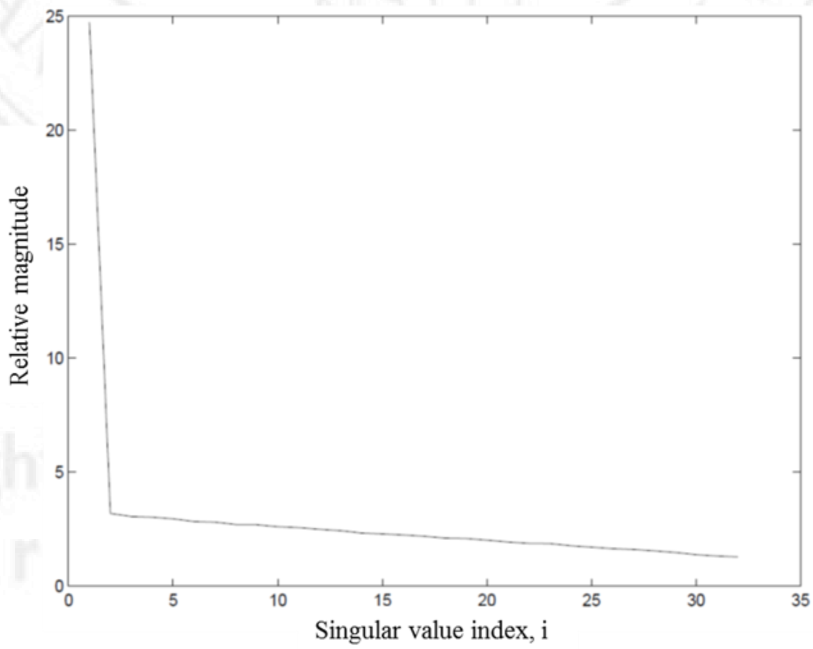

Figure 2-4. The singular value spectrum graph was an abrupt change within the singular values for the data in Figure 2-3 (from Signal Analysis and Imaging Group Department of Physics, University of Alberta, Available http://saig.physics.ualberta.ca/s/sites/default/files/chapter6.pdf. 2013, March 16).

2.4 Source code development

The Mathematica® Software was used to process the SVD filter with each forms of seismic and GPR data. Each forms of seismic and GPR data was converted into ASCII format for noise suppression and imported data to Mathematica® Software. The singular values were extracted and sorted from largest to smallest. The relative magnitude of each singular value was normalized. The singular value spectrum was plotted between relative magnitude with each singular value. The *p* and *q* value were selected from singular value spectrum slope characteristic. After data were filtered, the processing outputs were converted into SEG-Y and put back to VISTA® Software version 5.5. The SVD filter was developed by the flow chart in Figure 2-5.

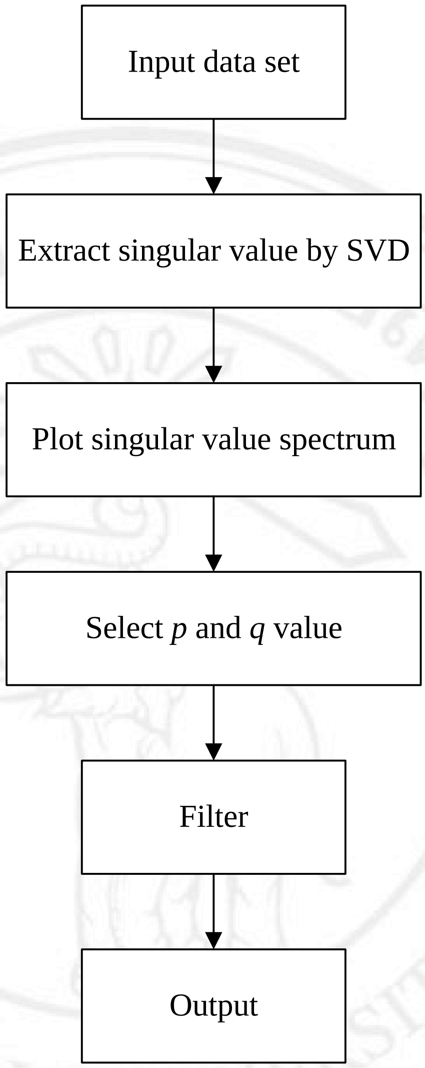

Figure 2-5. Process flowchart for the SVD filter.