Stanford Exploration Project, Report 93, November 11, 1997, pages 163–179

The pyramid transform and its application to signal/noise separation

Yalei Sun and Shuki Ronen[1]

keywords: *deconvolution, noise*

ABSTRACT

The "pyramid transform" is spatial-resampling of data in the frequency-space domain with frequency-dependent grids (`FDG`), in which the spatial-sampling interval is inversely proportional to the frequency. A cube in the F-X-Y domain is thus transformed to a pyramid. Wavefields do not contain high wavenumbers in low frequencies. Therefore, the pyramid transform is invertible when applied to wavefields. After pyramid transformation, the size of data has shunk dramatically. This feature can save many resources. However, the main benefit is that the low frequencies are not over-parameterized which makes frequency-dependent grids more suitable for inversion and interpolation. For example, spatial prediction filters become independent of the temporal frequency in the pyramid domain. This feature has a great potential in signal/noise separation and trace interpolation. In this paper, we investigate the possibility of applying the pyramid transform to noise suppression. Our results show the prediction filter estimated in the pyramid domain can remove the low temporal frequency random noise, which can not be handled by the prediction filter estimated from the common frequency-independent grids (`FIG`).

INTRODUCTION

Seismic data is recorded with constant temporal and spatial intervals (e.g., 4ms and 25m). Since seismic velocity does not change with time, many operators become time invariant and can be applied more economically in the temporal frequency domain (sometimes with the aid of the log-stretch transform). Spatial prediction for signal/noise separation (Canales, 1984) and spatial prediction for interpolation (Spitz, 1991) are two examples in the frequency-space (F-X-Y) domain. In the above two applications, the operators are conducted respectively from one frequency to another.

The subset of the data which contain all X-Y samples for one frequency is called a

[1]**email:** yalei@sep.stanford.edu, shuki@gatwick.Geco-Prakla.slb.com

`frequency slice`. All the frequency slices from low to high frequency have the same size in the frequency-space domain. According to Shannon's sampling theorem, we can use longer spatial interval to resample the low frequency data, which also means that the low frequency components are usually oversampled.

We present an new way to represent a seismic dataset, which is called `pyramid transform`. We take the conventional dataset in the F-X-Y domain as input and resample it in accordance with sampling theorem. The output is still in the F-X-Y domain. But it is no longer in the form of a cube. The shape of the output is a pyramid. More specifically, the lower the frequency, the smaller the number of data samples. But the pyramid is equivalent to the cube in terms of information amount. In other words, the `pyramid transform` is invertible. If we conduct the inverse transform, we can get back to the input dataset.

The dataset in the `pyramid domain` has some interesting features. For example, the spatial prediction filter is independent of temporal frequency. This means that we only need to estimate **one** prediction filter from many different frequencies. This feature will make those operators more efficient.

In this paper, we first discuss the `pyramid transform` and show how to make it invertible. Then we prove the advantages of the pyramid domain theoretically. Finally we apply this new scheme to signal/noise separation. Our result shows that the spatial prediction filter estimated in the pyramid domain can not only remove high temporal frequency noise but also eliminate low frequency noise, whereas the prediction filter obtained in the conventional way cannot handle low frequency noise.

PYRAMID TRANSFORM

Assume v is medium velocity, δx is trace interval and θ is dip angle, the maximum threshold frequency that is not aliased is as follows (Yilmaz, 1987):

$$f_{\max} = \frac{v}{4\delta x \sin\theta} \tag{1}$$

We can rewrite the above formula in another form:

$$\sin\theta = \frac{v}{4f\delta x} \tag{2}$$

Equation (2) is the relation between the dip angle and the threshold frequency. If f decreases, θ will increase. Since $0^\circ \le \theta \le 90^\circ$, there is a critical frequency $\bar{f}$ at which

$$\sin\theta = \frac{v}{4\bar{f}\delta x} = 1 \tag{3}$$

So we conclude that

- For $f < \bar{f}$, the frequency slice contains information up to 90°. In other words, the data is oversampled.

- For $f > \bar{f}$, the frequency slice only contains information $\theta \leq \theta_{\max}(< 90°)$. Dip over $\theta_{\max}$ will be aliased. And $\theta_{\max}$ is defined by

$$\theta_{\max} = sin^{-1}(\frac{v}{4\bar{f}\delta x})$$

The existence of $\bar{f}$ provides the feasibility of resampling the data in the `f-x` domain without losing any information. Resampling means that δx is no longer constant but a function of frequency, i.e., Δx_{f}. When $f < \bar{f}$ and $\theta = 90°$, equation (2) can be transferred into

$$\Delta x_{\mathrm{f}} = \frac{v}{4f} \tag{4}$$

Equation (4) shows that we can subsample the frequency slices using larger trace interval. According to sampling theorem, the resampled data does not lose any information theoretically. We call this resampling scheme "pyramid transform" and the output a special kind of F-X domain, i.e., `pyramid domain`.

Figure 1 shows the RHS of equation (3) versus frequency. The solid curve represents the frequency-independent grids. The dash line represents the frequency-dependent grids. From this figure, we can see that all the frequency lower than $\bar{f} = 94Hz$ have oversampled the data. Therefore, the next question is what is the disadvantage of oversampling the data?

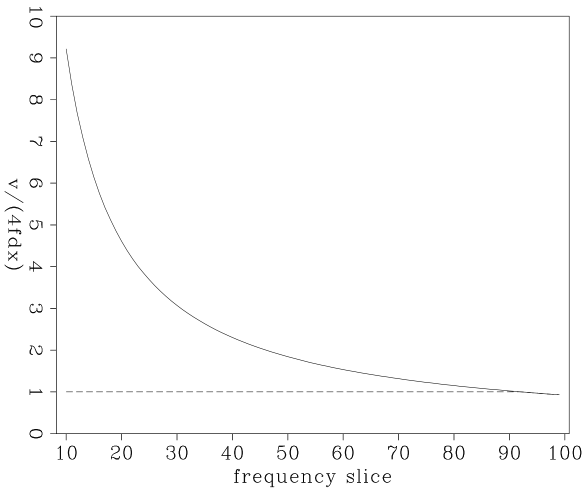

Figure 1: $\frac{v}{4f\delta x}$ versus `f`. The solid curve represents the frequency-independent grids. The dash line represents the frequency-dependent grids. yalei1-sin-value [NR]

Unless carefully constrained, an over-parameterized model may contain illegal components that will fit noisy data better than a better model which does not include high wavenumbers for low frequencies.

Spatial prediction filters become the same for all frequencies (Nichols, 1996). Since the prediction filter is independent of frequency, it will become more stable and more insensitive to the noise. Also, we do not need to estimate independent prediction filters for each frequency slice separately.

In any case, oversampling will not bring any profit to noise suppression. So we will use δx, not Δx_f, when $f > \bar{f}$. Figure 2 shows that when $f > \bar{f}$, δx is actually the base of another pyramid. So we can regard the input data ($f > \bar{f}$) as frequency-dependent grids directly.

In other words, we can get a series of pyramids corresponding to the frequency which is higher than $\bar{f}$. From each pyramid, we can estimate a prediction filter. The difference between the first prediction filter ($f \leq \bar{f}$) and the other ($f > \bar{f}$) , is that we can apply the first prediction filter to all the frequencies which are lower than $\bar{f}$. Each of the remaining filters can only be applied to one frequency slice, which is also the base of that pyramid.

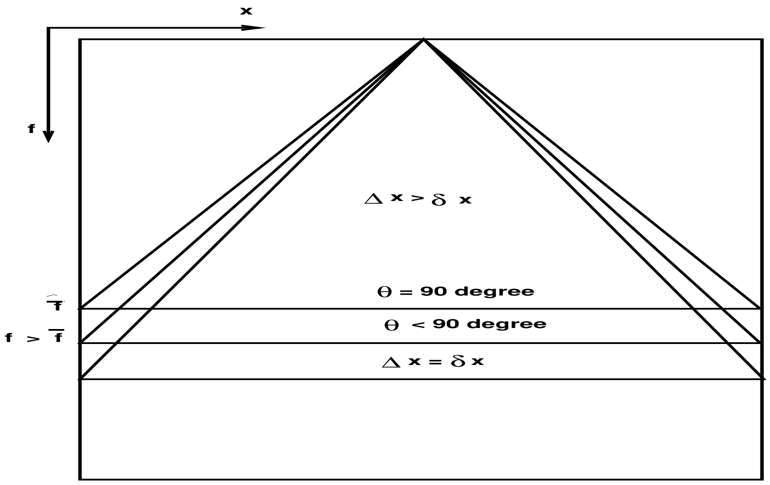

Figure 2: The size of the pyramid(in 2-D, it is actually a triangle) is changing when ($f > \bar{f}$). yalei1-pyramid [NR]

The basic and most important requirement for pyramid transform is that this transform is invertible. In other words, "forward" transform can turn a rectangular `f-x` domain to a pyramid domain and "inverse" transform can turn a pyramid domain back to a rectangular `f-x` domain, as shown in Figure 3.

How do we guarantee this transform invertible? The key is to find a good complex-valued interpolation scheme. Here, we use Dave Hale's 8-point sinc interpolation program in SU. Some more accurate and complicated interpolation scheme can be found in Wade's paper (1988). Figure 4 is one example using 8-point sinc interpolation. The result shows that our scheme is very accurate. There is little difference between input and output.

ESTIMATION OF SPATIAL PREDICTION FILTERS

In this section, we discuss the theory of estimating the prediction filter for plane wave. We use one and two plane waves as examples. First, we prove that plane wave is predictable. Then we show how to estimate the spatial prediction filter for both frequency-independent grids (**FIG**) and frequency-dependent grids (**FDG**). The

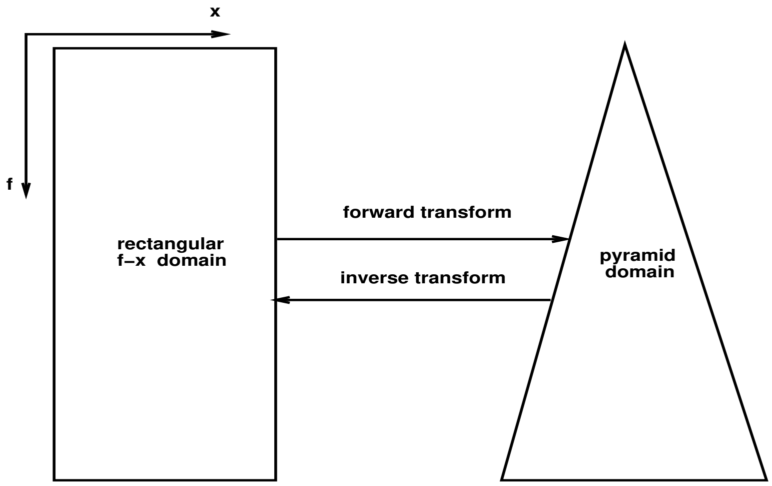

Figure 3: Schematic demonstration of forward/inverse pyramid transform. $\boxed{\text{yalei1-transform}}$ [NR]

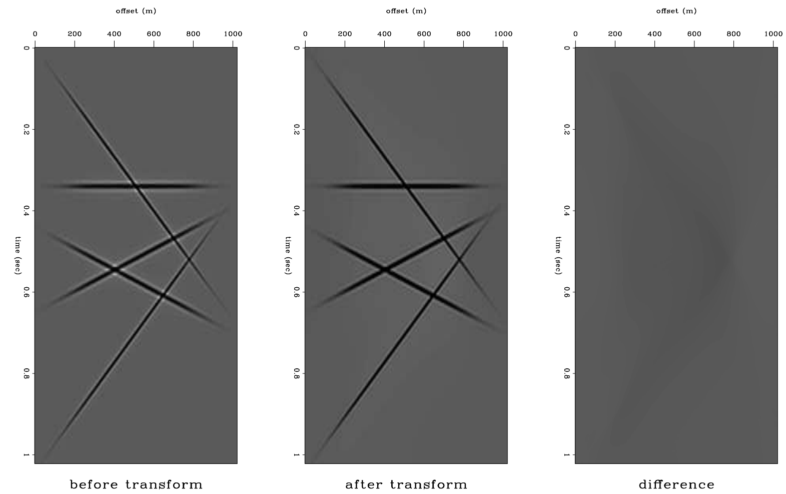

Figure 4: The left is the input dataset, the middle is the result after forward/inverse transform, and the right one is the difference.(Displayed at the same scale) $\boxed{\text{yalei1-forward-inverse-transform}}$ [ER]

difference is also the advantage of **FDG** over **FIG**. In the appendix, we extend our theory to N plane waves.

Wavefield composed of one plane wave

As shown in Figure 5, this wavefield in the `t-x` domain can be expressed by

$$w(t, x) = \delta(t - t_1 - p_1 x) \tag{5}$$

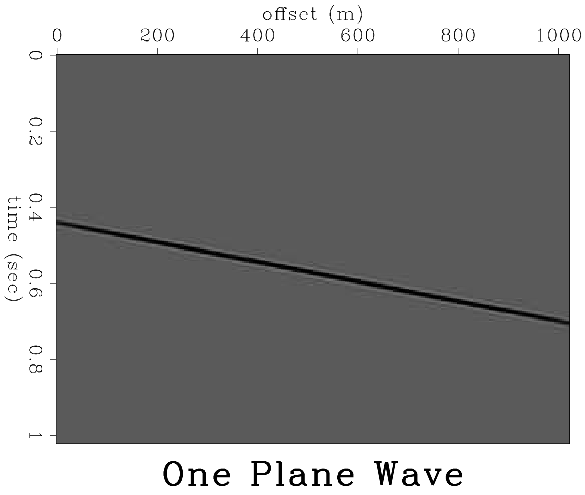

Figure 5: A synthetic wavefield composed of a single plane wave. yalei1-one-plane-wave [ER]

After Fourier transform along the time axis, we get the expression in the `f-x` domain:

$$W(f, x) = e^{i2\pi f(t_1 + p_1 x)} \tag{6}$$

It is easy to show that one trace can be predicted from an adjacent trace:

$$W(f, x - \Delta x) = e^{i2\pi f(t_1 + p_1 x - p_1 \Delta x)} = W(f, x)e^{-ip_1 2\pi f \Delta x} \tag{7}$$

Or in another form,

$$W(f, x) = e^{ip_1 2\pi f \Delta x} W(f, x - \Delta x) \tag{8}$$

This means that, each trace can be predicted with the propagator $P_1 = e^{ip_1 2\pi f \Delta x}$. Up to now, we have not mentioned **FIG** and **FDG**. Therefore, all above conclusions are effective for both cases. But when we introduce the details of the two schemes, we will reach different conclusions:

- Frequency-independent grids

  $\Delta x = \delta x$ is constant. The propagator P_1 is the function of frequency f. It will change from one frequency slice to another.

- Frequency-dependent grids

  From (3), we can choose
  $$\Delta x = \frac{v}{4f} \tag{9}$$
  Therefore, the propagator P_1 becomes
  $$P_1 = e^{i\frac{\pi}{2}p_1 v} \tag{10}$$
  So the propagator P_1 is frequency-invariant.

Wavefield composed of two plane waves

If we have two plane waves, the wavefield (Figure 6) can be expressed by
$$w(t,x) = a_1\delta(t - t_1 - p_1 x) + a_2\delta(t - t_2 - p_2 x) \tag{11}$$

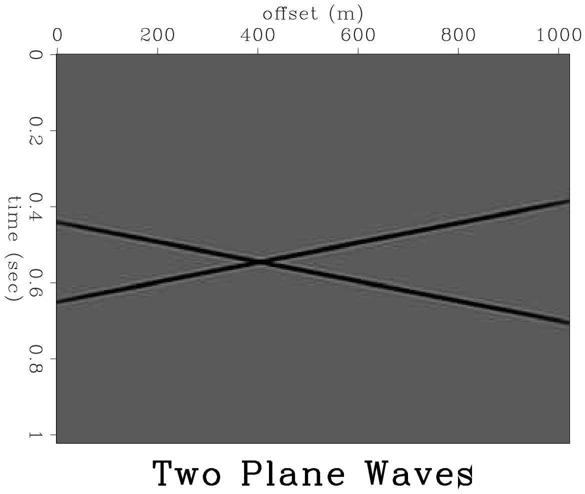

Figure 6: A synthetic wavefield composed of two plane waves. yalei1-two-plane-wave [ER]

In the f-x domain, it has another expression
$$W(f,x) = a_1 e^{i2\pi f(t_1+p_1 x)} + a_2 e^{i2\pi f(t_2+p_2 x)} \tag{12}$$
Or
$$W(f,x) = E_1 + E_2 \tag{13}$$
where $E_1 = a_1 e^{i2\pi f(t_1+p_1 x)}$ and $E_2 = a_2 e^{i2\pi f(t_2+p_2 x)}$. The adjacent trace can be expressed by
$$W(f, x - \Delta x) = P_1^{-1}E_1 + P_2^{-1}E_2 \tag{14}$$
Now we cannot predict one trace from another trace, since there are two unknown propagators. Consequently, we need the information from another trace.
$$W(f, x - 2\Delta x) = P_1^{-2}E_1 + P_2^{-2}E_2 \tag{15}$$

Equation (14) and (15) can be solved for the propagators. But in reality, we do not know p_1 and p_2. So we cannot get the explicit form of the propagators. We must use another form of parameter to link the the different traces together, which is called a spatial prediction filter.

If we have trace $x - \Delta x$ and trace $x - 2\Delta x$, our aim is to predict trace x. Assume there is a relation between the three traces:

$$W(f,x) = C_1 W(f, x - \Delta x) + C_2 W(f, x - 2\Delta x) \tag{16}$$

Substituting Ws with equations (13), (14) and (15), we get

$$E_1 + E_2 = C_1(P_1^{-1}E_1 + P_2^{-1}E_2) + C_2(P_1^{-2}E_1 + P_2^{-2}E_2) \tag{17}$$

E_i can be any value, as long as it is a plane wave. Therefore, equation (17) holds for all of E_1 and E_2. In particular, we have

$$1 = C_1 P_1^{-1} + C_2 P_1^{-2}, \quad (E_1 = 1, \quad E_2 = 0) \tag{18}$$

$$1 = C_1 P_2^{-1} + C_2 P_2^{-2}, \quad (E_1 = 0, \quad E_2 = 1) \tag{19}$$

Solving this linear equation, we find the relation between C_i and P_i $(i = 1, 2)$. That is

$$C_1 = P_1 + P_2, \qquad C_2 = -P_1 P_2 \tag{20}$$

Like the one plane wave case, all above deductions have nothing to do with the gridding scheme. All the results are effective for both cases. The differences between the two gridding schemes are

- Frequency-independent grids

  As proved before, since the propagator P_i depends on frequency, the coefficients of the prediction filter C_i will also depend on frequency.

- Frequency-dependent grids

  Since the propagator P_i is independent of frequency, the coefficients of the prediction filter C_i are also independent of frequency.

We can extend our deduction to N plane waves (see the Appendix) and we can get similar results: for **FIG**, coefficient C_i is frequency-dependent; for **FDG**, C_i is frequency-independent. This is the basis of estimating the spatial prediction filter.

Estimating the spatial prediction filter

We use subscript k to represent the location of the trace, i.e., $W_k = W(f, k\Delta x)$. If W_k is predictable from $(W_{k-1}, W_{k-2}, ..., W_{k-N})$, we have

$$W_k = \sum_{j=1}^{N} C_j W_{k-j}, \quad (k = 2, 3, ..., N_2) \tag{21}$$

where N_2 is the length of the spatial axis.

Determination of the coefficients C_{j} can be treated as an optimization problem. Here we choose conjugate gradient code `cgplus` (Claerbout, 1994) as the solver. Originally, `cgplus` was used for real-valued optimization. We extend it to complex-valued optimization. Between the two gridding schemes, the most important difference is that

- Frequency-independent grids
  - Prediction filter is estimated in the rectangular `f-x` domain.
  - For each frequency slice, we need to solve an optimization problem separately. And the coefficients change from one slice to another.
- Frequency-dependent grids
  - Prediction filter is estimated in the pyramid domain.
  - We incorporate all the equations together and only solve one optimization problem. But the coefficients can be applied to all the frequencies within the pyramid.

Hopefully, this difference will save computing resources in 3-D. Also, we can expect that the result from **FDG** will more stable.

APPLICATION AND DISCUSSION

In this section, we compare our new sampling scheme in the application of signal/noise separation with F-X noise suppression. We expect that our scheme will produce better results. Figure 7 shows the relation and difference between the two implementations:

The input dataset is composed of one horizontal event, four dipping events. This dataset has no random noise. Therefore, it is fully predictable. We use this data to test whether subsampling will influence the predictability of each frequency slice. Figure 8 shows the data in both `t-x` and `f-x` domain.

F-X noise suppression algorithm estimates the spatial prediction filter for each frequency slice and then applies it to the input data. Figure 9 shows that there is little difference between input and output. This means that if there is no noise or aliased energy in the dataset, the prediction filter will produce a dataset which is the same as the input.

Figure 10 is the corresponding result from our scheme. It is clear that subsampling does not make the frequency slice more unpredictable. All the five events have been retained after convolving the prediction filter with the input dataset. Figure 11 is the

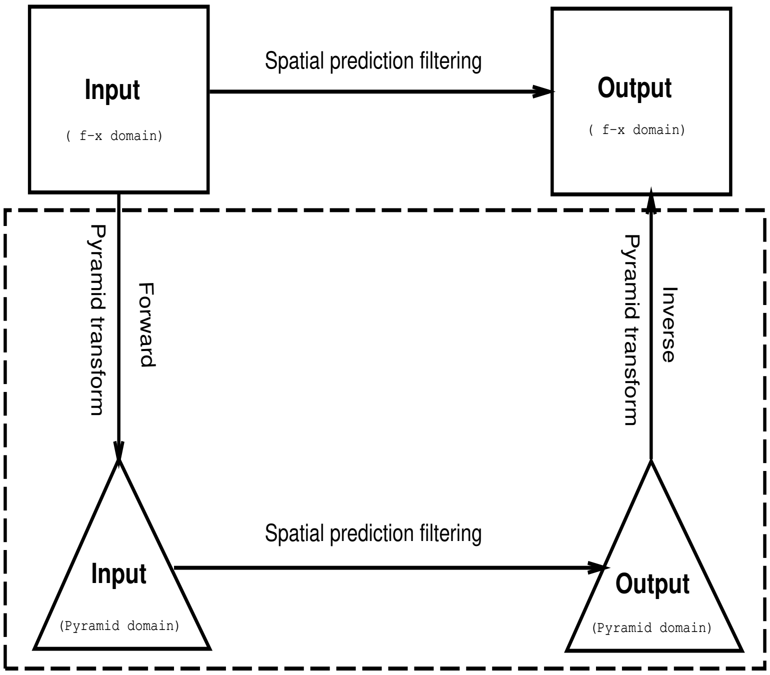

Figure 7: The relation between F-X noise suppression and our scheme. We estimate the spatial prediction filter in the pyramid domain. The dashed-line box includes all the operations in the pyramid domain. yalei1-flow-chart [NR]

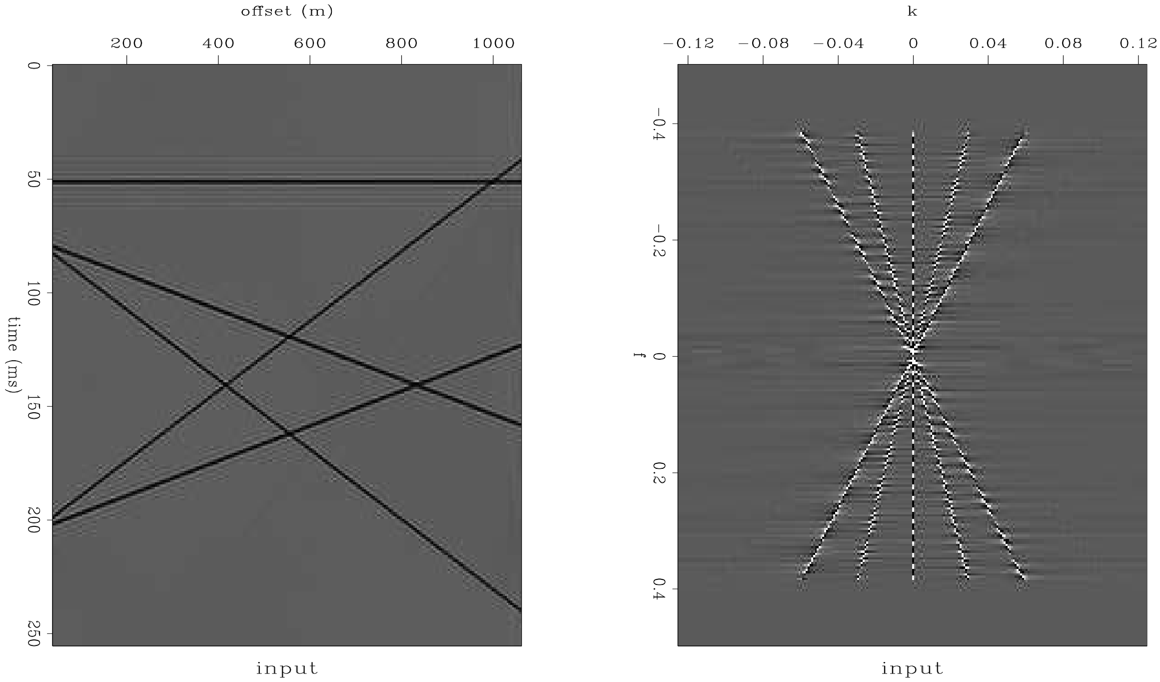

Figure 8: Input data in the `t-x` domain and `f-k` domain. yalei1-input1 [ER]

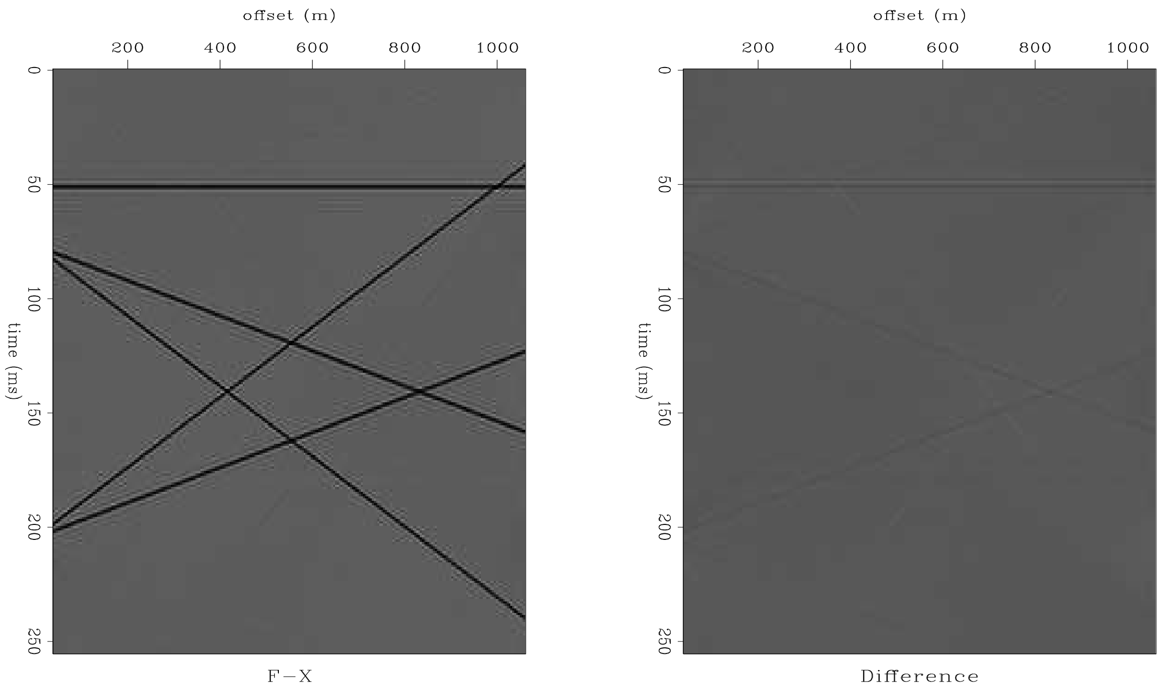

Figure 9: Result from F-X noise suppression algorithm in the `t-x` domain. The left is the output and the right is the difference between input and output. yalei1-FIG1 [ER]

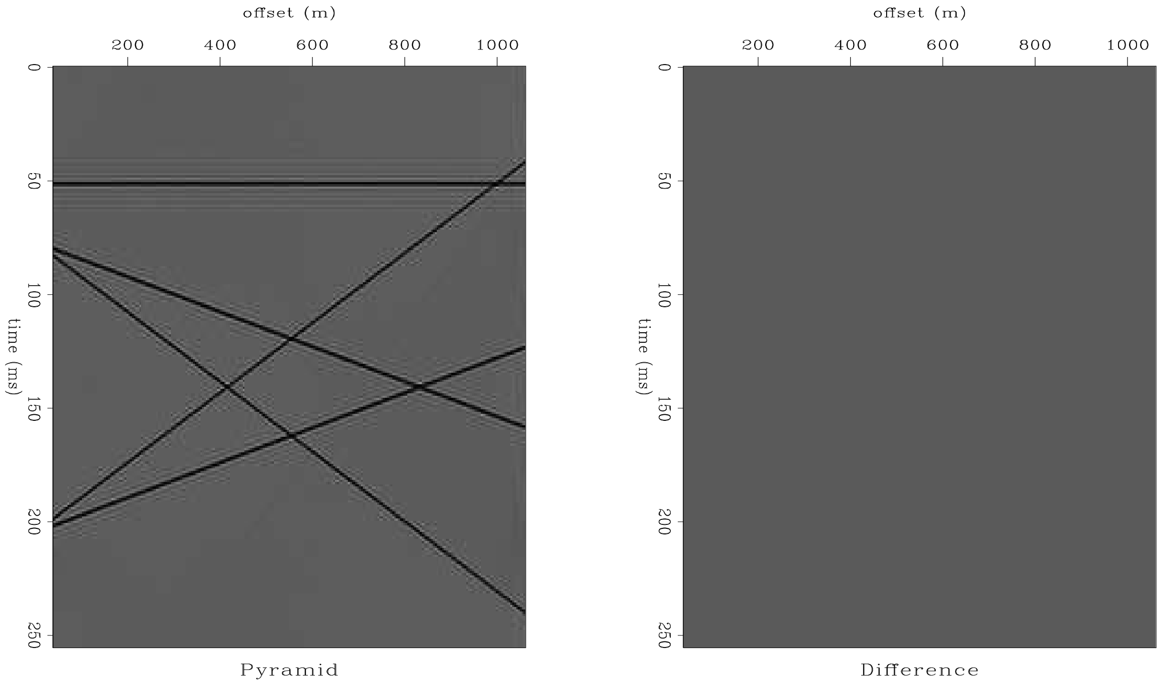

Figure 10: Result from Pyramid Transform scheme in the `t-x` domain. The left is the output and the right is the difference between input and output. yalei1-FDG1 [ER]

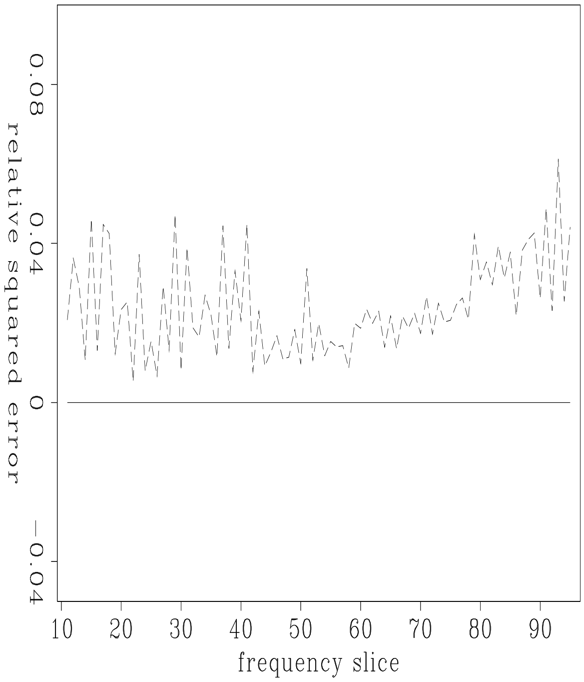

Figure 11: Squared error for both F-X noise suppression and Pyramid scheme. Notice that the subsampled dataset is even more predictable than the oversampled dataset. $\boxed{\text{yalei1-error1}}$ [ER]

relative residual error for each frequency. It is interesting to notice that subsampled dataset is even more predictable than the oversampled dataset.

The second dataset is the same as the first one, except that we introduce random noise into it, as shown in Figure 12.

Unfortunately, we have not got very good results from noisy data. As shown in Figure 13, F-X noise suppression cannot remove low frequency noise. Therefore, the difference is very small. Although Pyramid scheme can remove some parts of low frequency noise, it can also hurt the signal at the same time. We will keep working on it and try to get a better result.

CONCLUSION AND FUTURE WORK

Pyramid transform is a new sampling scheme. It is based on Shannon's sampling theorem. Theoretically pyramid transform will not introduce error to the dataset. After the pyramid transform, we get a frequency-dependent grid. This new grid has a feature that $f\Delta x = const$. This feature makes the spatial prediction filter estimated in the pyramid domain frequency-independent. We apply this new sampling scheme to signal/noise separation. Our result shows that subsampling will not make the data more unpredictable. The subsampled data is even more predictable than the oversampled one. Pyramid scheme can eliminate low frequency noise, but it also hurts the signal as well. We will continue this project and try to get better results.

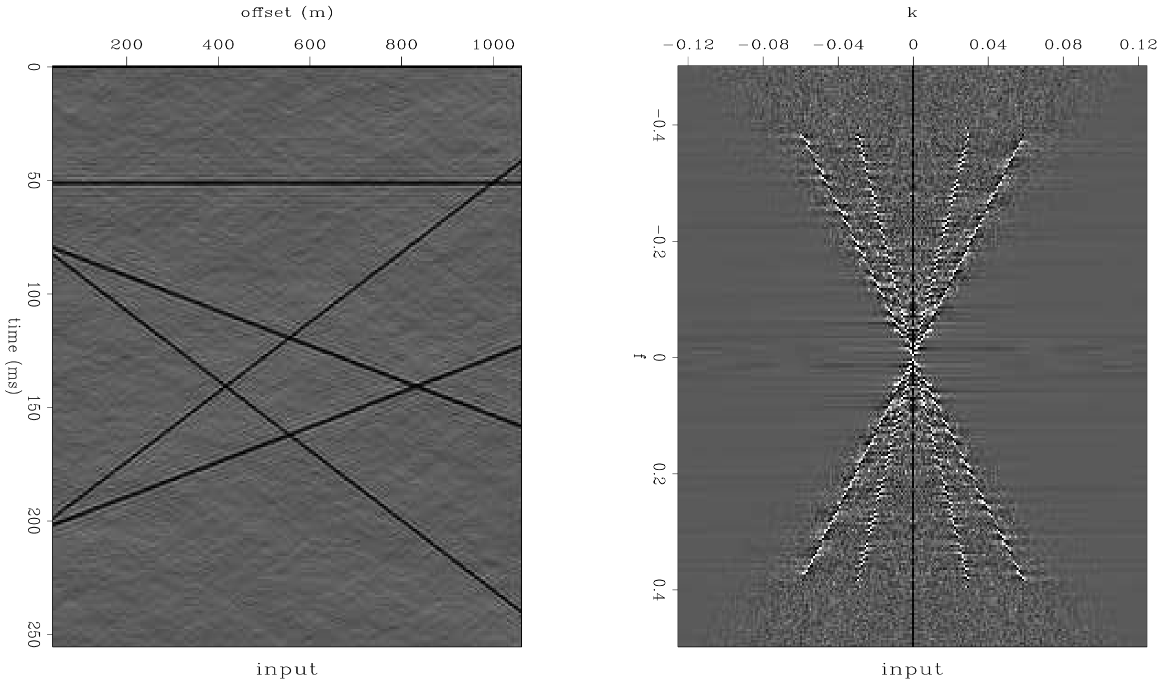

Figure 12: Input data in the `t-x` domain and `f-k` domain. There is no random noise in the low frequency, high wavenumber part. yalei1-input2 [ER]

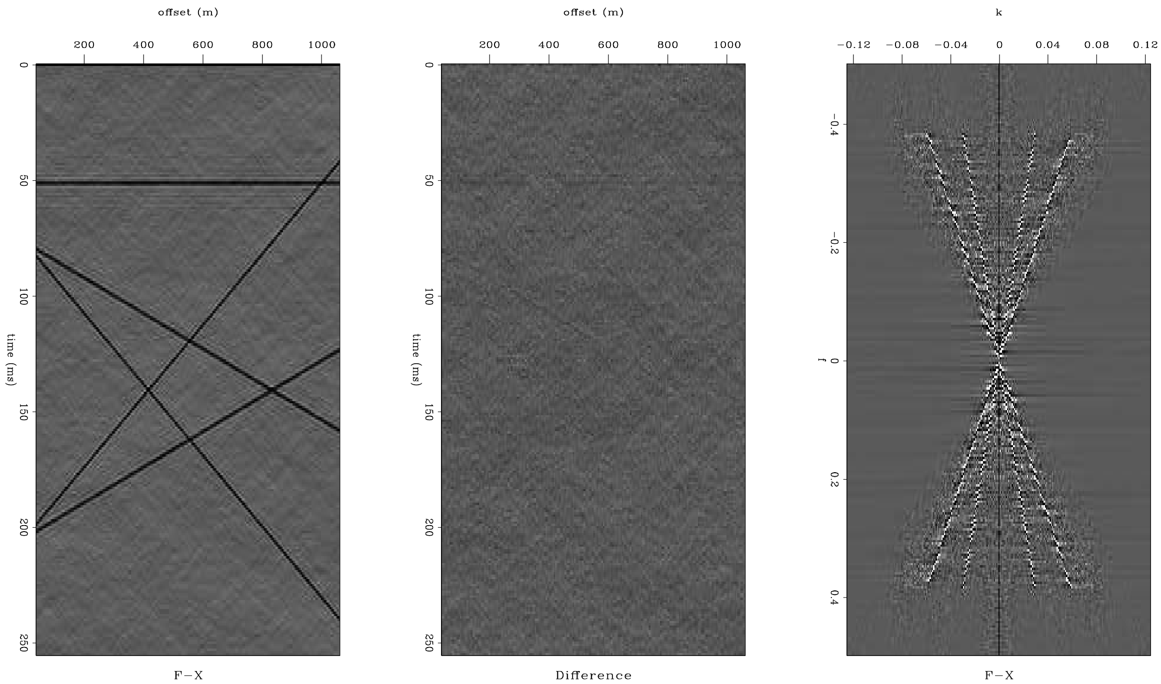

Figure 13: Result from F-X scheme. The left is the output, the middle is the difference between input and output, and the right is the output in `f-k` domain. Only some high frequency noise is eliminated from the input. Therefore the difference is very small. yalei1-FIG2 [ER]

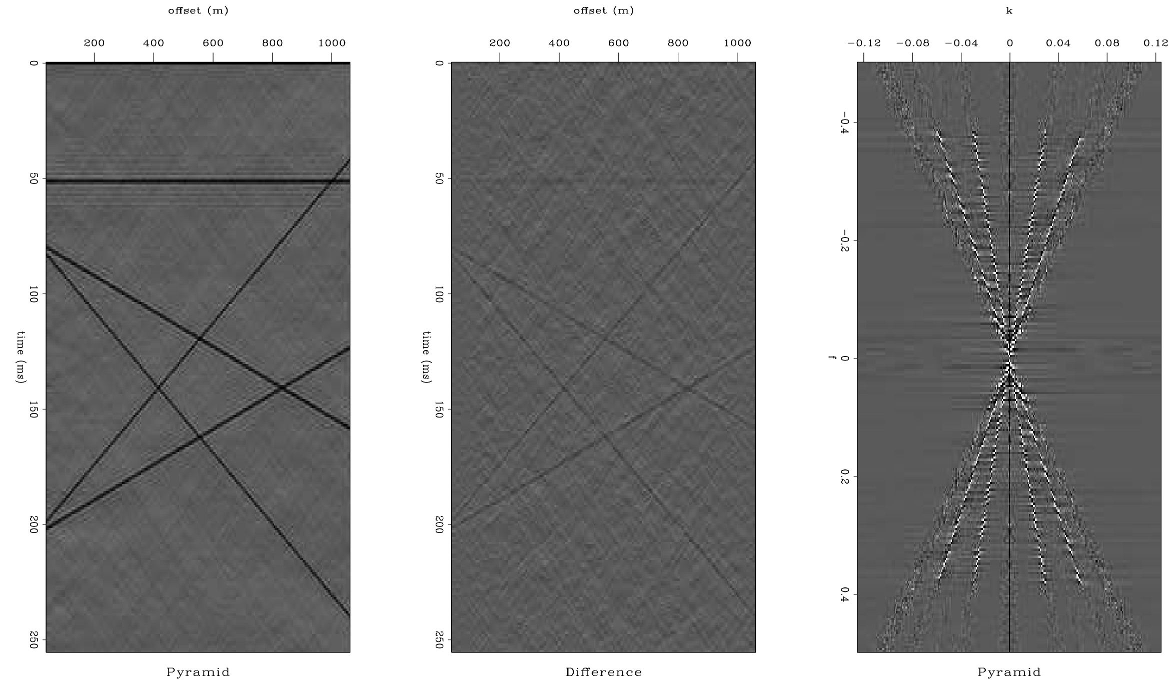

Figure 14: Result from Pyramid scheme. The left is the output, the middle is the difference between input and output, and the right is the difference between input and output, and the right is the output in `f-k` domain. Pyramid scheme can remove both high and low frequency noise. But it also remove some signal components. yalei1-FDG2 [ER]

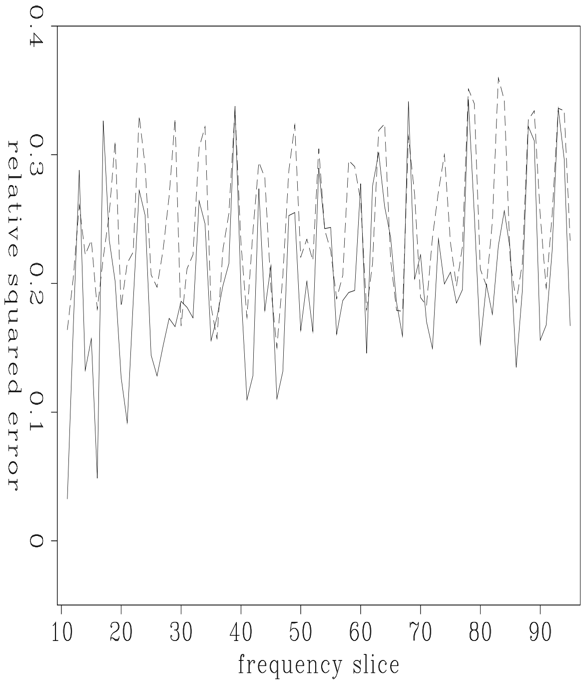

Figure 15: Squared error for both schemes. yalei1-error2 [ER]

REFERENCES

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

Claerbout, J. F., 1994, Applications of Three-Dimensional Filtering: Stanford Exploration Project.

Nichols, D., 1996, Personal communication:.

Spitz, S., 1991, Seismic trace interpolation in the f-x domain: Geophysics, **56**, no. 6, 785–794.

Wade, C. J., 1988, An efficient fourier interpolation operator: 58th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session:SP1.6.

Yilmaz, O., 1987, Seismic data processing: Soc. Expl. Geophys., Tulsa, OK.

APPENDIX A

Wavefield composed of N plane waves

Here, we extend prediction filter theory to N plane waves. Assume that the data have N plane waves with different dipping angles, the dataset can then be expressed by

$$w(t,x) = \sum_{\mathrm{i}=1}^{N} a_\mathrm{i}\delta(t - t_\mathrm{i} - p_\mathrm{i}x) \tag{A-1}$$

Fourier transform along the time axis will give us

$$W(f,x) = \sum_{\mathrm{i}=1}^{N} a_\mathrm{i} e^{i2\pi f(t_\mathrm{i}+p_\mathrm{i}x)} = \sum_{\mathrm{i}=1}^{N} E_\mathrm{i} \tag{A-2}$$

where $E_\mathrm{i} = a_\mathrm{i} e^{i2\pi f(t_\mathrm{i}+p_\mathrm{i}x)}$.

Trace $(x - j\Delta x)$ is represented by

$$W(f, x - j\Delta x) = \sum_{\mathrm{i}=1}^{N} P_\mathrm{i}^\mathrm{j} E_\mathrm{i} \tag{A-3}$$

where propagator $P_\mathrm{i} = e^{i\frac{\pi}{2}p_\mathrm{i}v}$. Assuming trace $W(f, x - j\Delta x), j = 1, 2, ..., N$ is known, trace $W(f,x)$ can be predicted by a N points prediction filter

$$W(f,x) = \sum_{\mathrm{j}=1}^{N} C_j W(f, x - j\Delta x) \tag{A-4}$$

Inserting equation (A-2) into equation (A-4)

$$\sum_{\mathrm{i}=1}^{N} E_{\mathrm{i}} = \sum_{\mathrm{j}=1}^{N} C_j \sum_{\mathrm{i}=1}^{N} P_i^{-j} E_i = \sum_{\mathrm{i}=1}^{N} E_i \sum_{\mathrm{j}=1}^{N} P_i^{-j} C_j \tag{A-5}$$

For each $E_{\mathrm{i}}, i = 1, 2, ..., N$

$$1 = \sum_{\mathrm{j}=1}^{N} P_i^{-j} C_j \tag{A-6}$$

Equation (A-6) can be expressed in matrix form

$$\begin{bmatrix} 1 \\ 1 \\ . \\ 1 \end{bmatrix} = \begin{bmatrix} P_1^{-1} & P_1^{-2} & . & . & . & P_1^{-\mathrm{N}} \\ P_2^{-1} & p_2^{-2} & . & . & . & P_2^{-\mathrm{N}} \\ . & . & . & . & . & . \\ P_{\mathrm{N}}^{-1} & P_{\mathrm{N}}^{-2} & . & . & . & P_{\mathrm{N}}^{-\mathrm{N}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ . \\ C_{\mathrm{N}} \end{bmatrix} \tag{A-7}$$

Equation (A-7) is a Van der Monde system. This system guarantees that there is one solution. So every C_{i} is a function of P_{j}. This means that the prediction filter relies on frequency in the case of frequency-independent grid; whereas in the case of frequency-dependent grids, we can still get a prediction filter which is independent from frequency.

Estimating the prediction filter coefficients

For sake of simplicity, we denote $W_{\mathrm{j}} = W(f, j\Delta x)$. W_{j} is predictable from $W_{\mathrm{j}-1}, W_{\mathrm{j}-2}, ..., W_{\mathrm{j}-\mathrm{N}}$.

$$W_{\mathrm{j}} = \sum_{\mathrm{z}=1}^{\mathrm{N}} C_{\mathrm{z}} W_{\mathrm{j}-\mathrm{z}} \tag{A-8}$$

Actually, equation (A-8) is a convolution operator. For each frequency `f`, we have an equation like (A-8). All frequencies have the same prediction filter $C_1, C_2, ..., C_{\mathrm{N}}$. Usually this is an over-determined system. We can use some optimization method to solve it, e.g., conjugate gradient, LSQR, Lanzcos, etc.