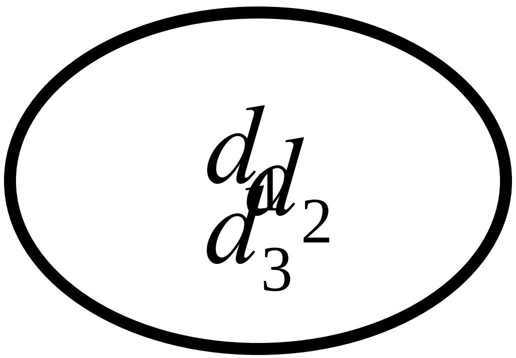
Radon Transforms and SSA

Mauricio D Sacchi

SAIG University of Alberta

Noise attenuation with “TRANSFORM-BASED” methods



[Q] I want to remove d_3

$$d = d_1 + d_2 + d_3$$

How?

One possibility is to use PhotsShop. However, it is too time consuming for seismic data processing…

Noise attenuation with “TRANSFORM-BASED” methods

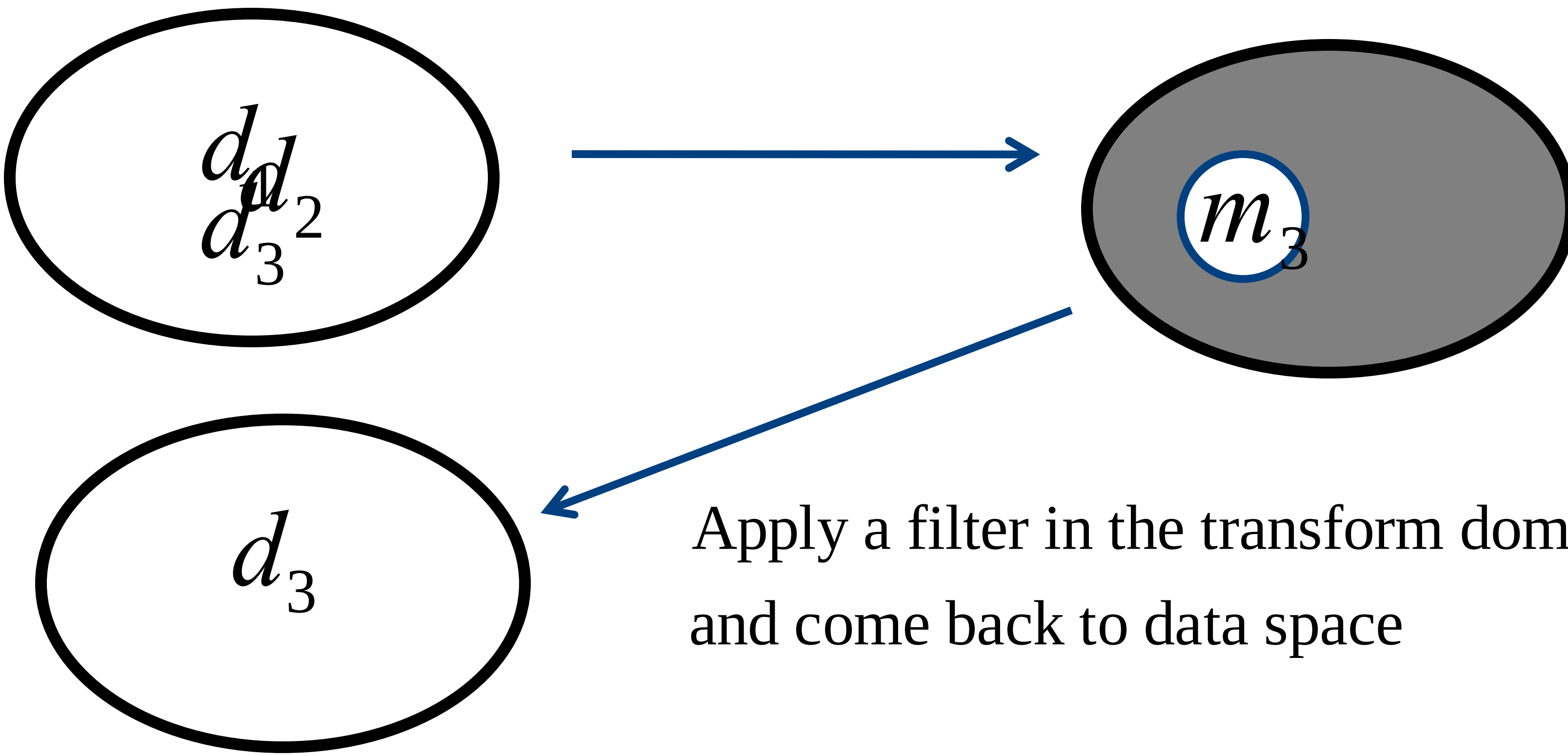
Note the isolation of
events in Transform Domain -> This is important for the transform to permit noise separation, reconstruction etc.

[A] Use a transform

$$d = d_1 + d_2 + d_3 \rightarrow m = m_1 + m_2 + m_3$$

Noise attenuation with “TRANSFORM-BASED” methods

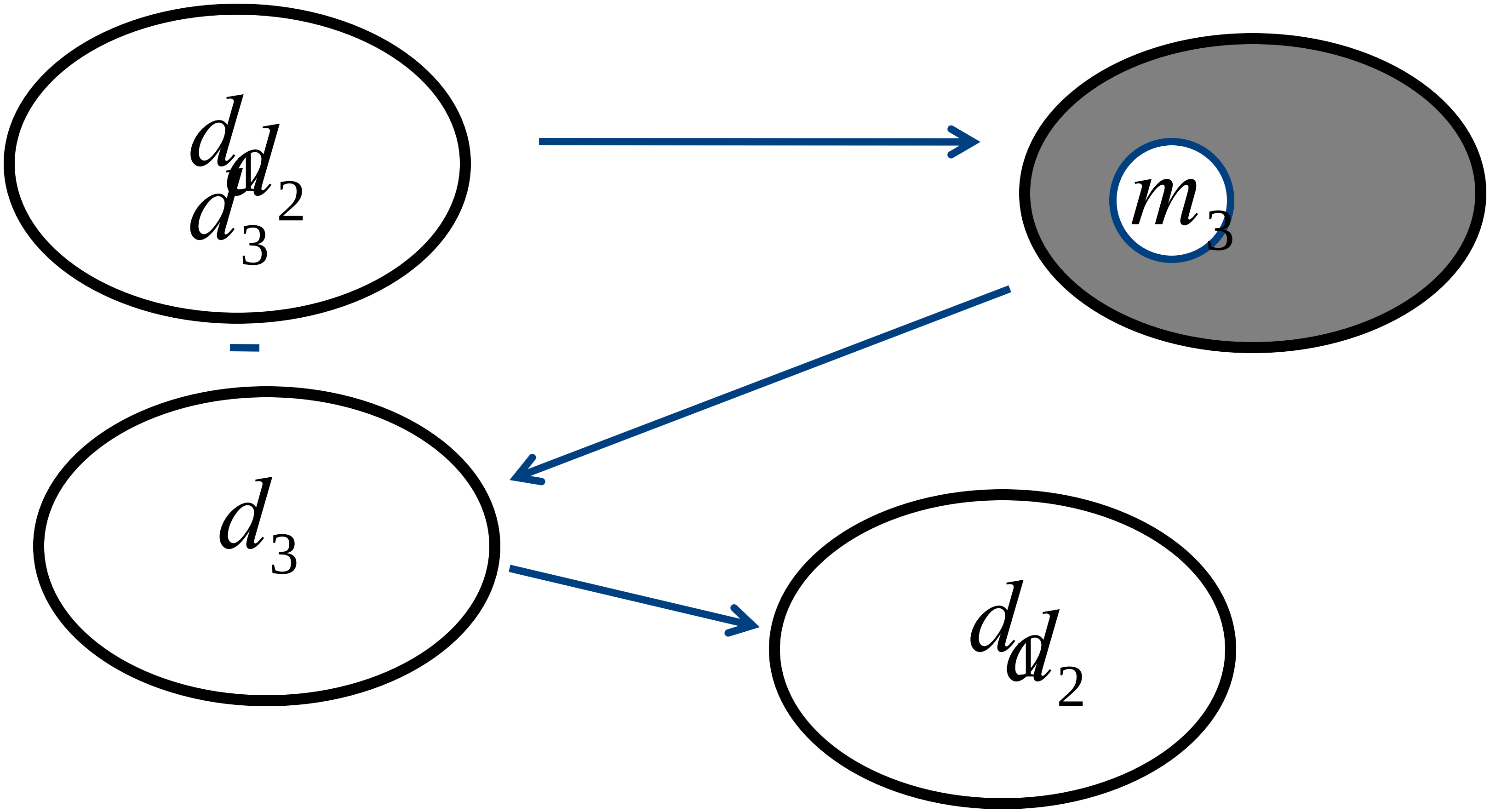
Note the isolation of
events in Transform Domain -> This is
important for the transform to permit noise
separation, reconstruction etc.



Apply a filter in the transform domain and come back to data space

Noise attenuation with “TRANSFORM-BASED” methods

Note the isolation of
events in Transform Domain -> This is
important for the transform to permit noise
separation, reconstruction etc.



Radon transforms

- Search for coherent waveforms along particular travel-time curves
- Three common implementations: linear, parabolic, hyperbolic
- Resolution is severely affected by amplitude variations and aperture
  - Resolution = Ability to distinguished events with similar moveout
- HR implementations can cope with aperture problems but not with AVO.
- AVO could be modeled but might be expensive and hard to parameterize for processing large batches of data.
- Radon transforms are often used as non-local operators (Operator spans the full aperture of the gather) This is a problem as it entails the assumption of high degree of matching of the signal to the transform.
    - Solution: Going local - but again hard to parameterize

- Companies often use the name

- HIGH RESOLUTION RADON → SPARSE RADON TRANSFOMR

- Is a RADON TRANFORM with FOCUSING of coefficients

- ***Time Variant and Time Invariant Transforms***
  - Parabolic and Linear → Time Invariant
  - Hyperbolic →Time Variant
  - In general, Time Invariant transforms lead to fast algorithms with operators with Toeplitz form.

Some refs.

- **High Resolution Hyperbolic RT** **[Time]**

  Thorson and Claerbout (GEOP, 1985)

- **Least-squares parabolic RT** **[Freq]**

  Dan Hampson (CSEG, 1986)

- **Sparse Parabolic RT** (high-resolution) **[Freq]**

  Sacchi and Ulrych – (GEOP, 1995)

- **De-aliased Parabolic RT** (high-resolution) **[Freq]**

  Phillippe Hermann et al – (SEG, 2000)

- **Latest views of the sparse Radon transform** **[Time and Freq]**

  Trad, Ulrych and Sacchi (Geop 2003)

Radon Integration Traveltime Paths

time invariant curves

$$t = t + qh \quad \text{Linear}$$

$$t = t + qh^2 \quad \text{Parabolic}$$

$$t = \sqrt{t^2 + qh^2} \quad \text{hyperbolic}$$

$$t = \sqrt{t^2 + q(h - h_0)^2} \quad \text{hyperbolic apex shifted}$$

This is also a post-stack migration algorithm

Fourier shift theorem (Important)

Fourier delay theorem is used to design f-domain Transforms:

$$f(t - t_0) \leftrightarrow F(\omega)\exp(-i\omega t_0)$$

Time and freq. domain forms for the PRT

$$m(t,q) = \sum_h d(t = t + qh^2, h)$$

$$M(w,q) = \sum_h D(w,h)\, e^{iwqh^2}$$

These are the transforms that we do use…

$$d(t,h) = \sum_q m(\tau = t - qh^2, q)$$

$$D(\omega,h) = \sum_q M(\omega,q)\, e^{-i\omega q h^2}$$

Forward/Adjoint or Forward/Transpose transform pair

$$d(t,h) = \sum_q m(\tau = t - q h^2, q)$$

$$D(\omega,h) = \sum_q M(\omega,q)\, e^{-i\omega q h^2}$$

Forward or Synthesis

$$\hat{m}(\tau,q) = \sum_h d(t = \tau + q h, h^2)$$

$$\hat{M}(\omega,q) = \sum_h D(\omega,h)\, e^{i\omega q h^2}$$

Adjoint

The hat is important (discuss it)

Key points about Radon

- **The adjoint is not the inverse (unless the operator is orthogonal)**
- **Time or Freq. Let's write the problem in Matrix or Operator form**

$$\mathbf{d} = \mathbf{L}\mathbf{m}$$

We write our data **d** as the sum of events in Radon space **m**

$$\mathbf{d} = \mathbf{L}\mathbf{m} + \mathbf{n}$$

We also add noise and modeling errors

Find the Radon Panel via the solution of an inverse problem

$$J = \| \mathbf{d} - \mathbf{L}\mathbf{m} \|_2^2 + \mu \mathbf{R}(m)$$

Time domain RT:	Solve all at once
Freq. domain RT:	Solve one small IP for each frequency

Some nice features of freq domain RT

- Toeplitz structure

$$J(\omega) = \| \mathbf{d}(\omega) - \mathbf{L}(\omega)\mathbf{m}(\omega) \|_2^2 + \mu \| \mathbf{m}(\omega) \|_2^2$$

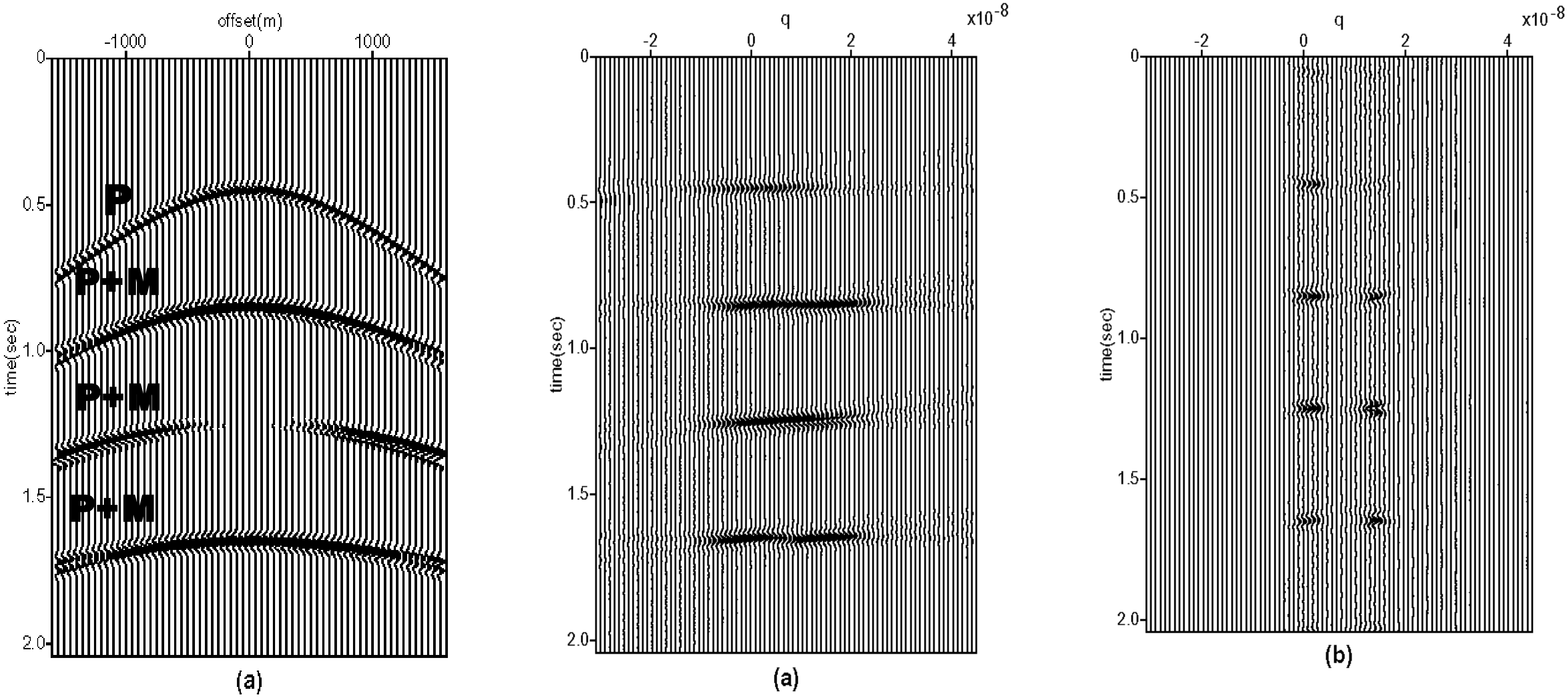
$$\nabla J(\omega) = 0 \rightarrow$$

$$\mathbf{m}(\omega) = (\mathbf{L}(\omega)^{\mathbf{H}} \mathbf{L}(\omega) + \mu \mathbf{I})^{-1} \mathbf{L}(\omega)^{H} \mathbf{d}(\omega)$$

•Toeplitz if and only if *q-axis* is regularly sampled. This leads to fast algorithms.

•Valid for the linear and parabolic cases (time-invariant) and for the time-invariant hyperbolic transform proposed by Mosher and Foster (Geophysics 1992)

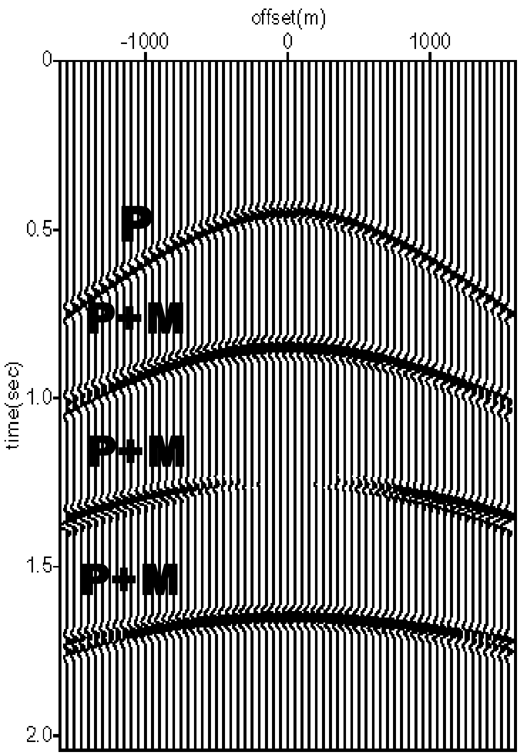
Improving resolution



smooth model

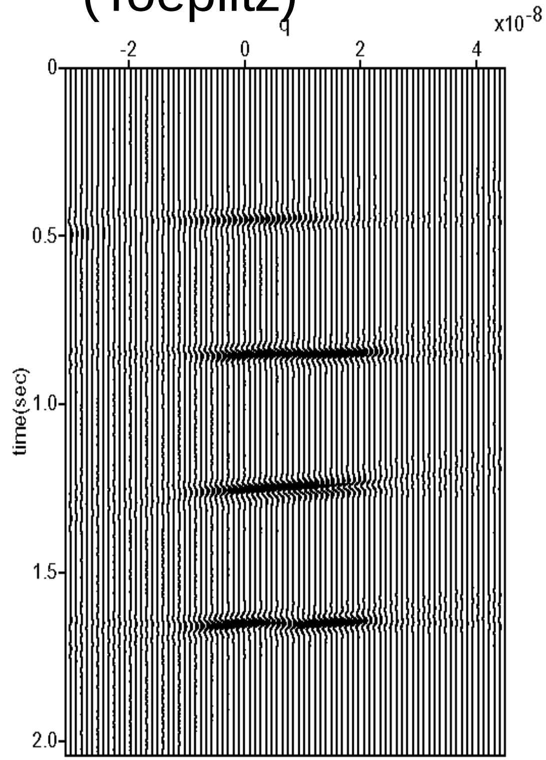
sparse model

Improving resolution



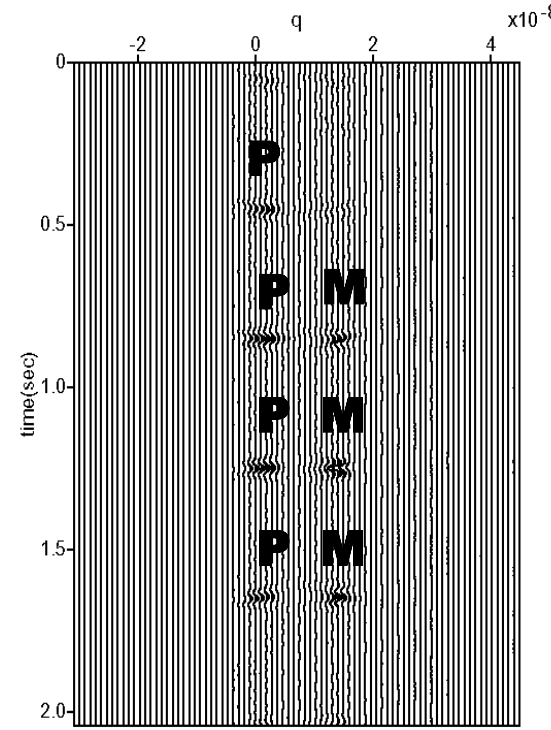
(a)

LS PRT Radon
(Toeplitz)



(a)

HR PRT Radon



(b)

smooth model

sparse model

Flavors of Freq. domain RT (Sparse or Hi-Res Radon)

$$J(\omega) = \| \mathbf{d}(\omega) - \mathbf{L}(\omega)\mathbf{m}(\omega) \|_2^2 + \mu R(\mathbf{m}(\omega)) \|_2^2$$

$$\nabla J(\omega) = 0 \rightarrow$$

$$\mathbf{m}(\omega) = (\mathbf{L}(\omega)^{\mathbf{H}}\mathbf{L}(\omega) + \mu\mathbf{Q}(\mathbf{m}))^{-1}\mathbf{L}(\omega)^{H}\mathbf{d}(\omega)$$

$$Q_{ii}(m(\omega)) \propto \frac{1}{|m(\omega)_i|}$$

All high res methods must use weights with that form. Why?

Flavors of Freq. domain RT

IRLS (Sacchi and Ulrych, GEO95)

$$\mathbf{m}^{k}(w) = (\mathbf{L}(w)^{\mathbf{H}}\mathbf{L}(w) + n\mathbf{Q}(\mathbf{m}^{k-1}(w)))^{-1}\mathbf{L}(w)^{H}\mathbf{d}(w)$$

CGG - P. Herrmann et al., 02

$$\mathbf{m}(w_{k}) = (\mathbf{L}(w_{k})^{\mathbf{H}}\mathbf{L}(w_{k}) + n\mathbf{Q}(\mathbf{m}(w_{k-1})))^{-1}\mathbf{L}(w_{k})^{H}\mathbf{d}(w_{k})$$

Flavors of Freq. domain RT (I. Moore)

2 - passes

$$1)\mathbf{m}(w) = (\mathbf{L}(w)^{\mathbf{H}}\mathbf{L}(w) + m\mathbf{I})^{-1}\mathbf{L}(w)^{H}\mathbf{d}(w)$$

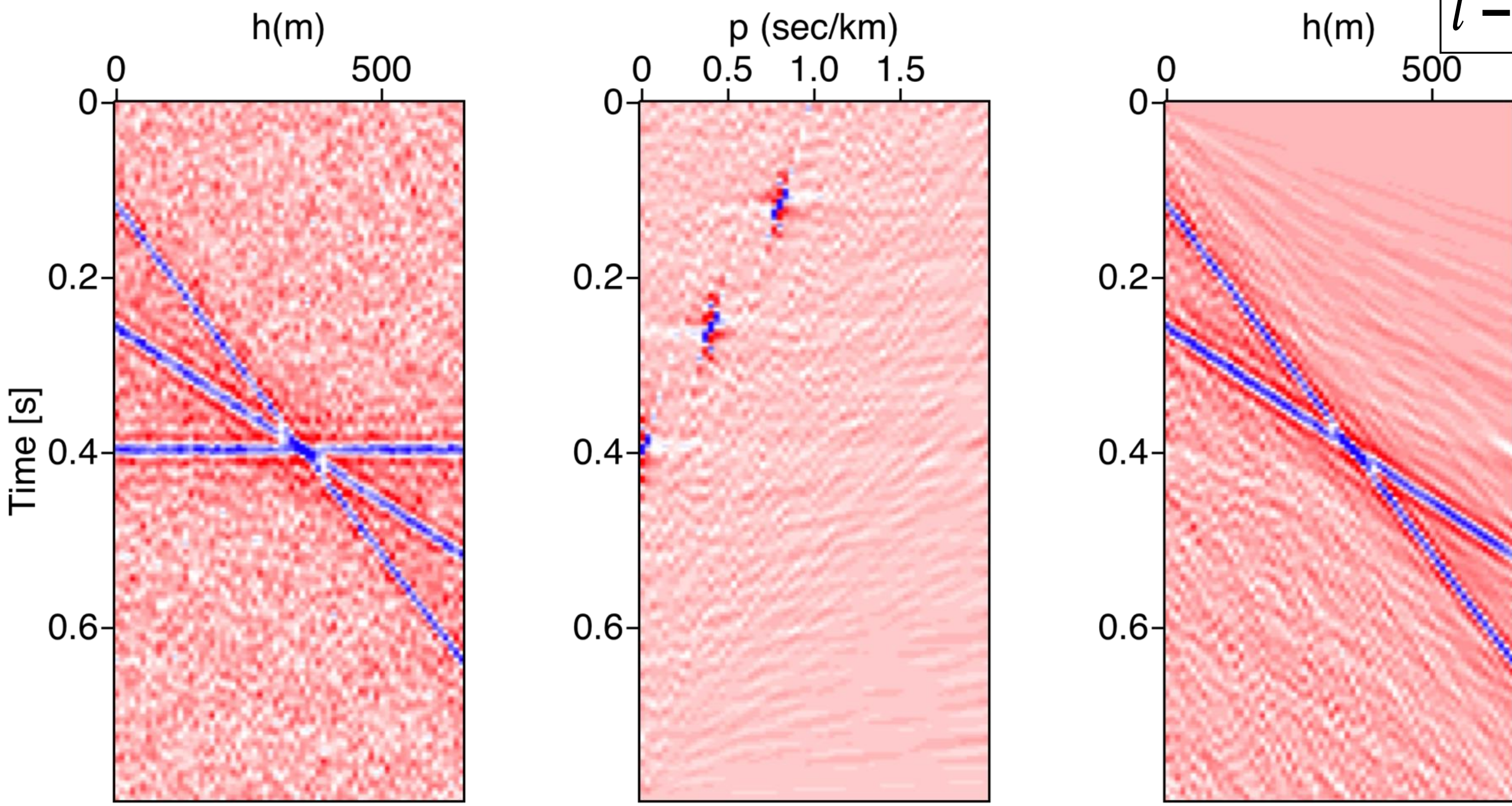
$$2) Q^{-1}\ \mu\ Semblance(m)$$

$$3)\mathbf{m}(w_k) = (\mathbf{L}(w_k)^{\mathbf{H}}\mathbf{L}(w_k) + m\mathbf{Q}(S(w)))^{-1}\mathbf{L}(w_k)^{H}\mathbf{d}(w_k)$$

Comparison of Least-squares (Smooth) versus Sparse (high-res) Radon transform → IRLS

Linear Radon Transform

$$t = t + qh$$



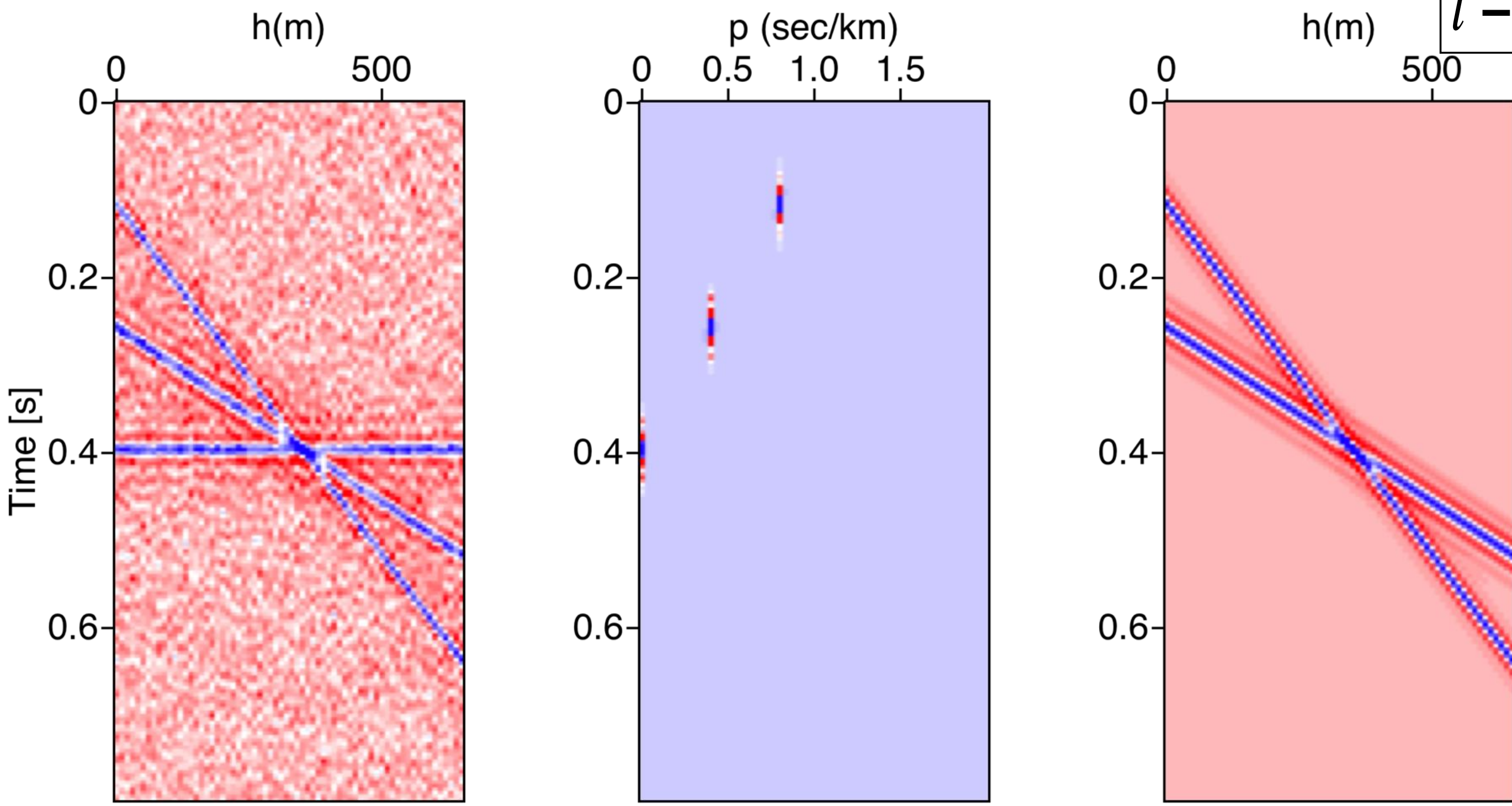
Data

Data in Radon Domain

Recovered data after filtering the horizontal event

Linear Radon Transform

$$t = t + qh$$

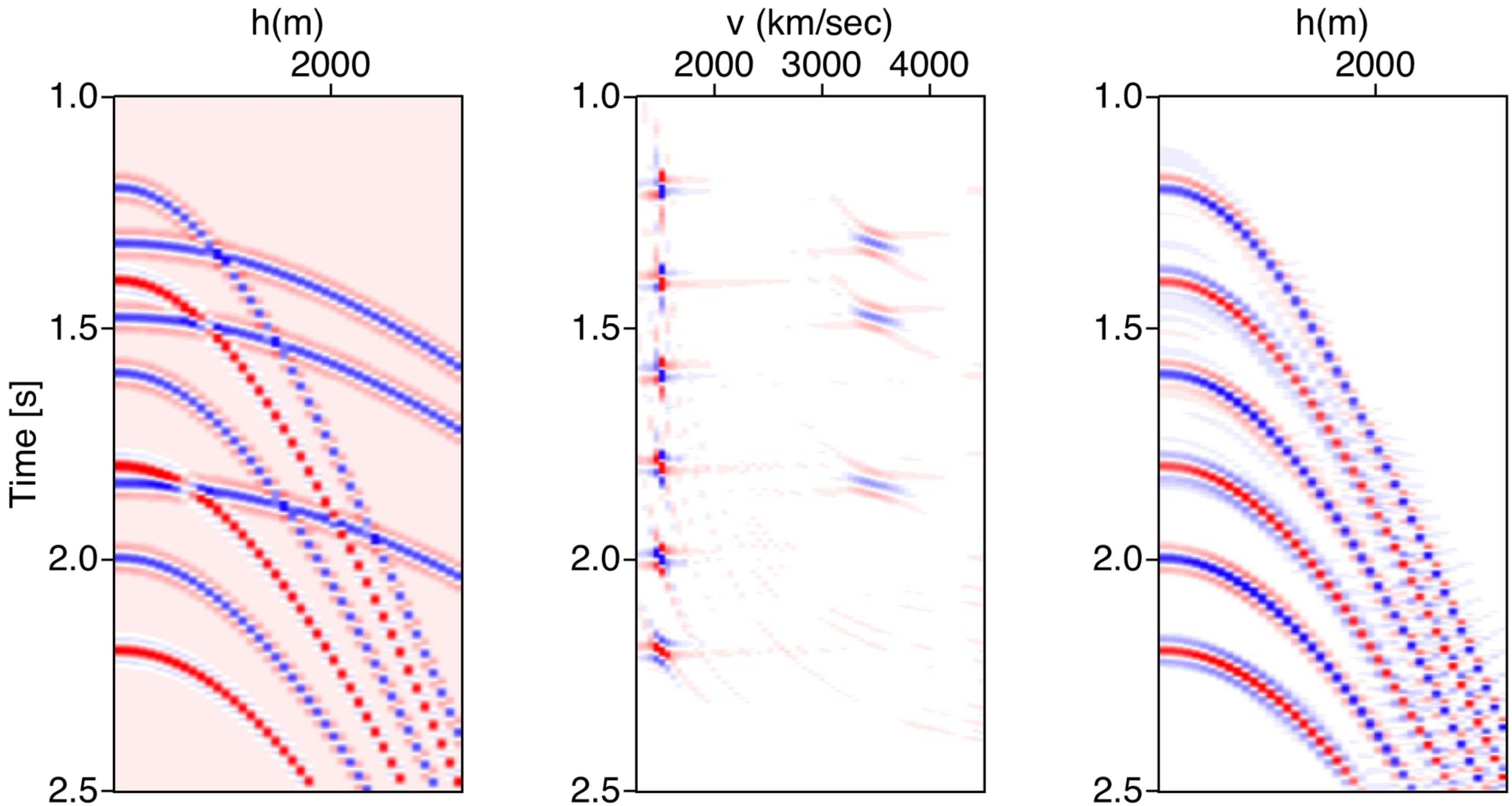


Data

Data in Radon Domain

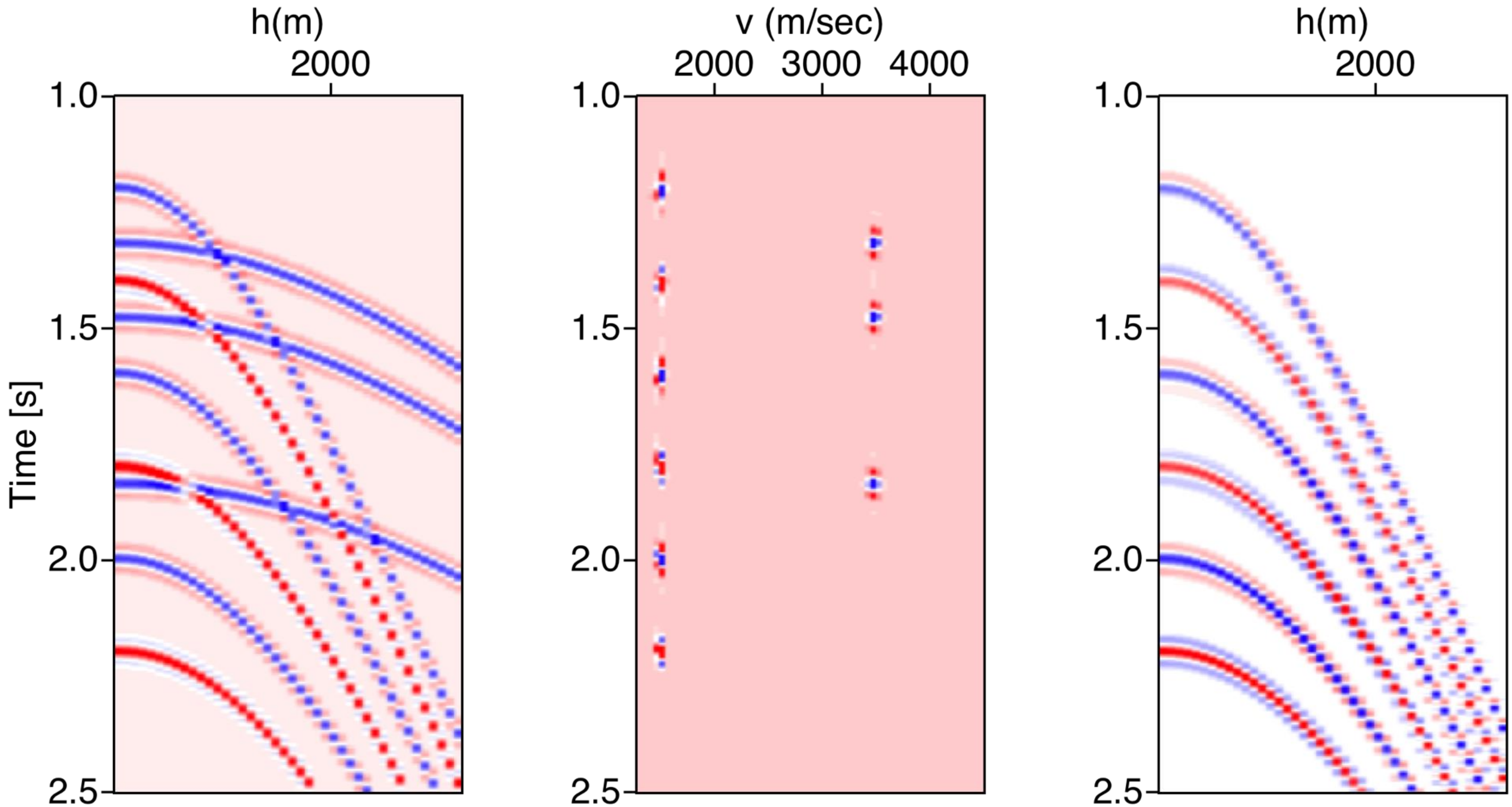
Recovered data after filtering the horizontal event

Hyperbolic Radon Transform



$$\mathrm{t} = \sqrt{t^2 + qh^2}$$

Hyperbolic Radon Transform



$$\mathrm{t} = \sqrt{t^2 + qh^2}$$

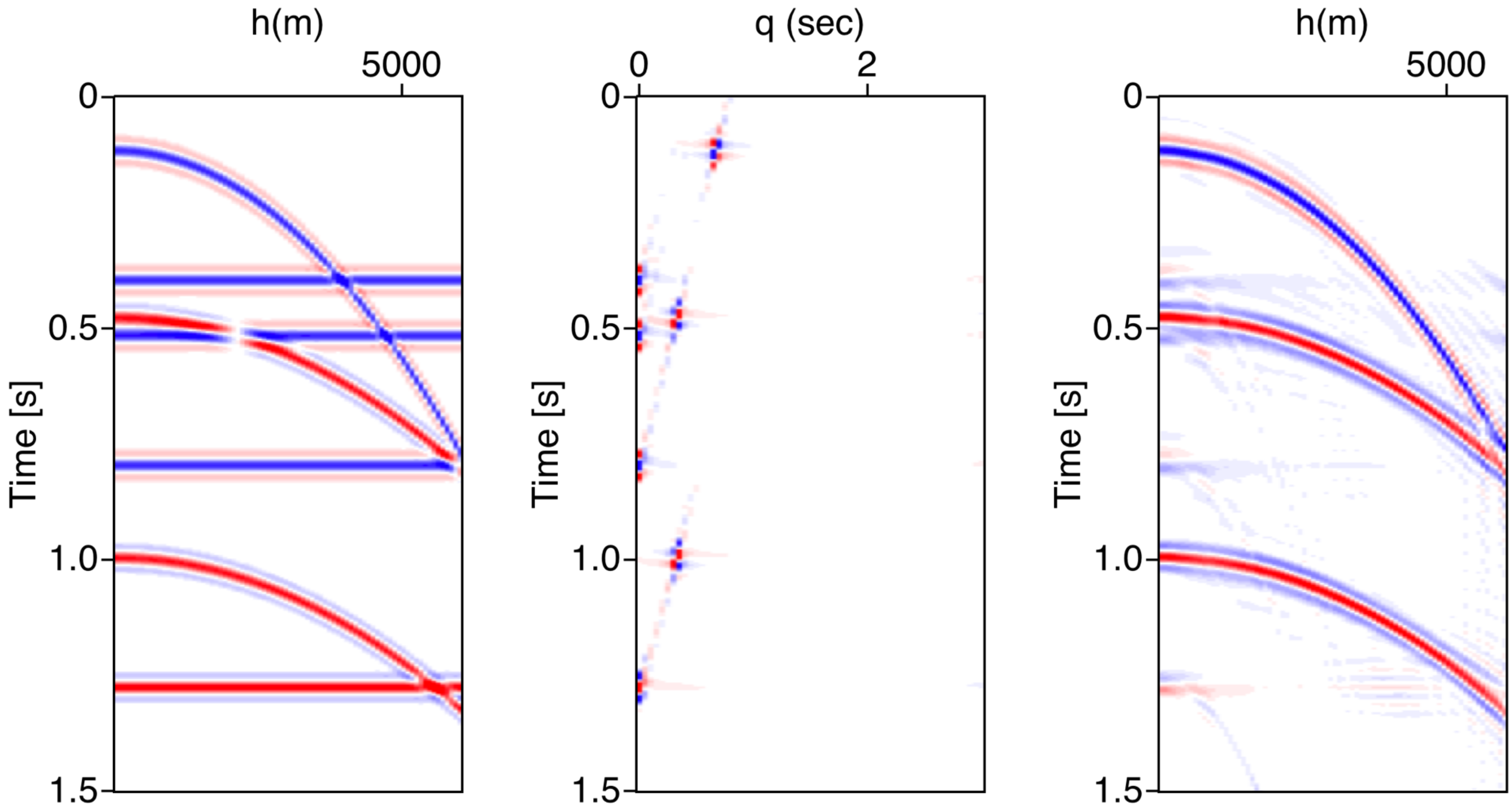
Parabolic Radon Transform: A note on q

- q is residual moveout at far offset (in secs)

$$\boxed{t = \tau + q h^2 \quad \rightarrow t = \tau + q {\frac{h}{h_{\max}^{2}}}^{2}}$$

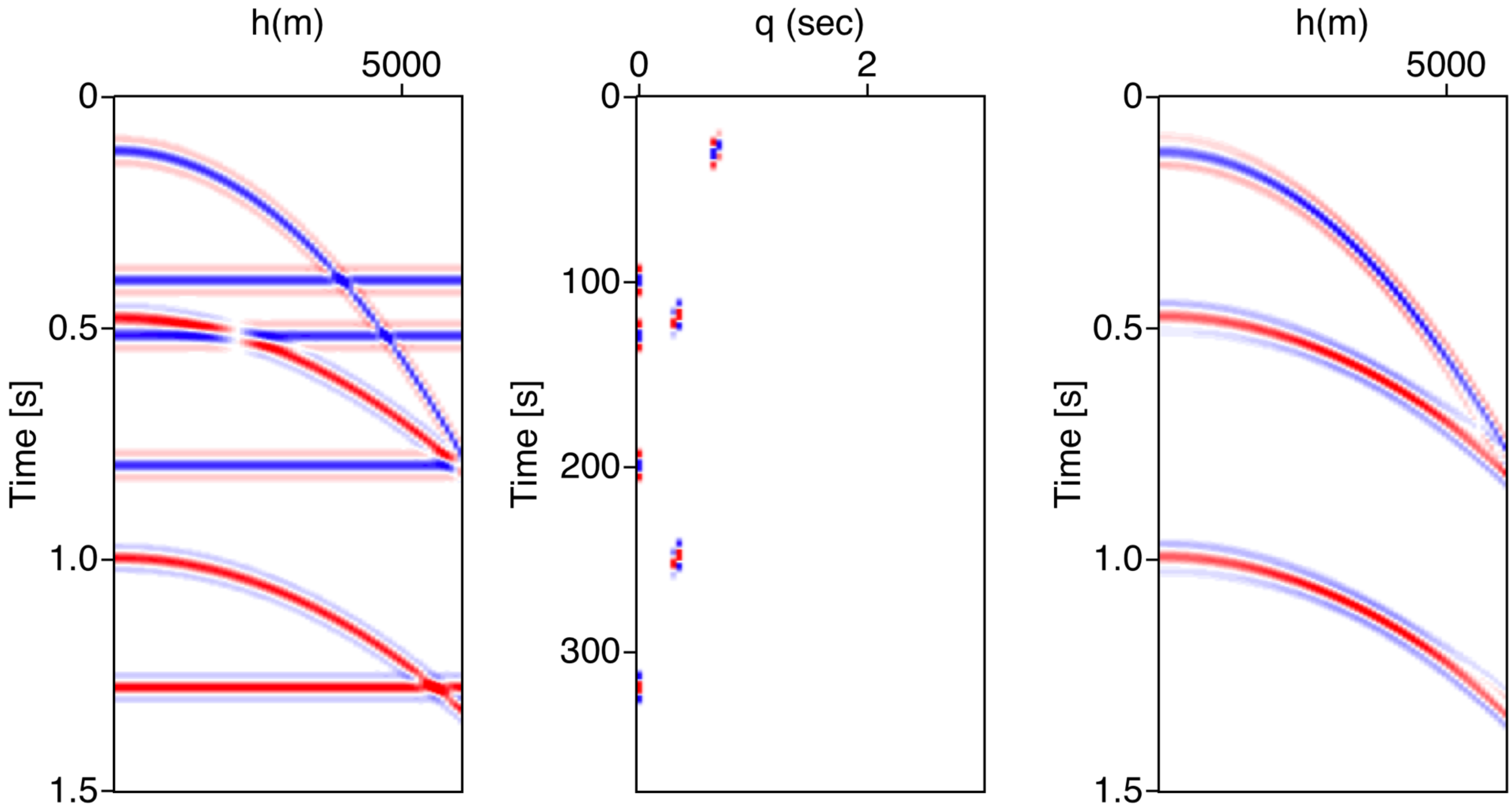
- Hyperbolic traveltimes after NMO correction can be approximated by parabolic traveltime curves.

Parabolic Radon Transform



$$\boxed{t = t + qh^2}$$

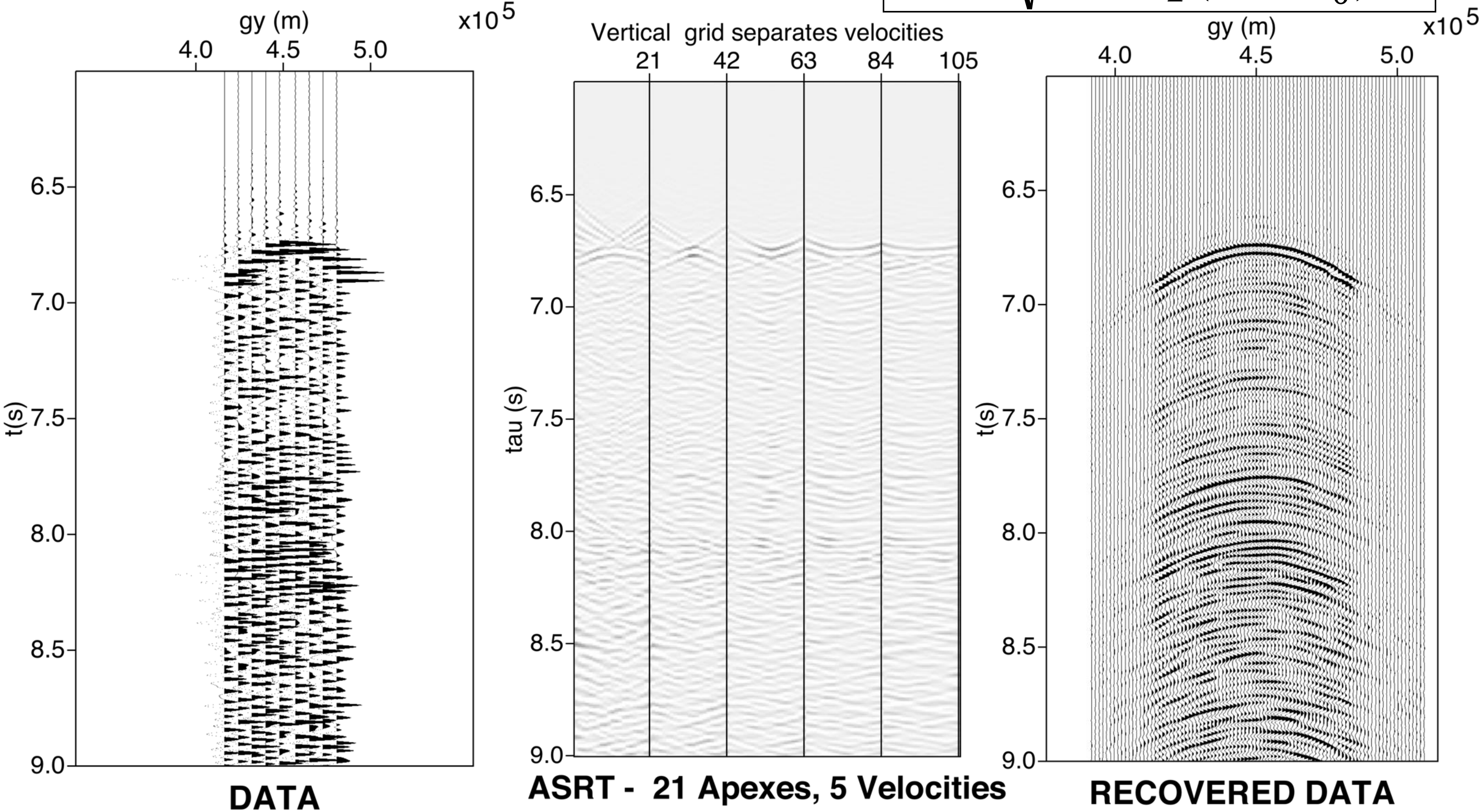
Parabolic Radon Transform



$$t = t + qh^2$$

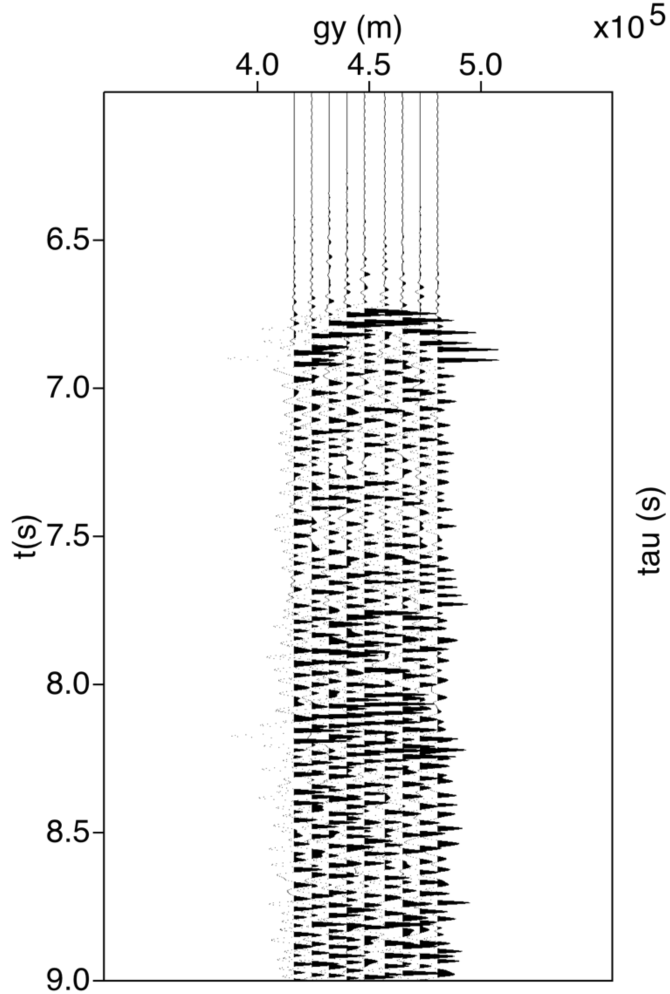
Hyperbolic Radon Transform with apex

$$t = \sqrt{t^2 + q(h - h_0)^2}$$

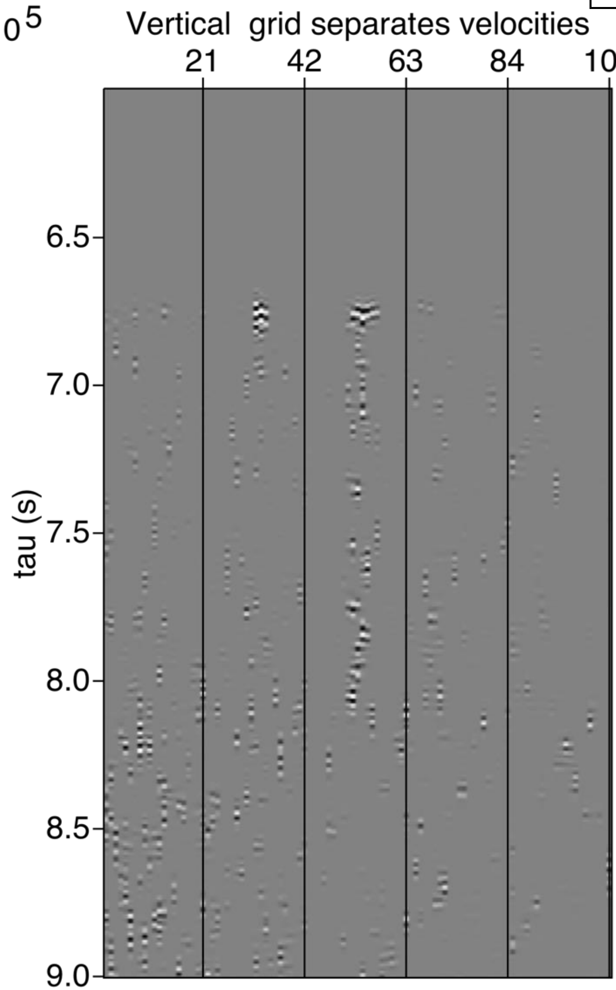


Hyperbolic Radon Transform with apex

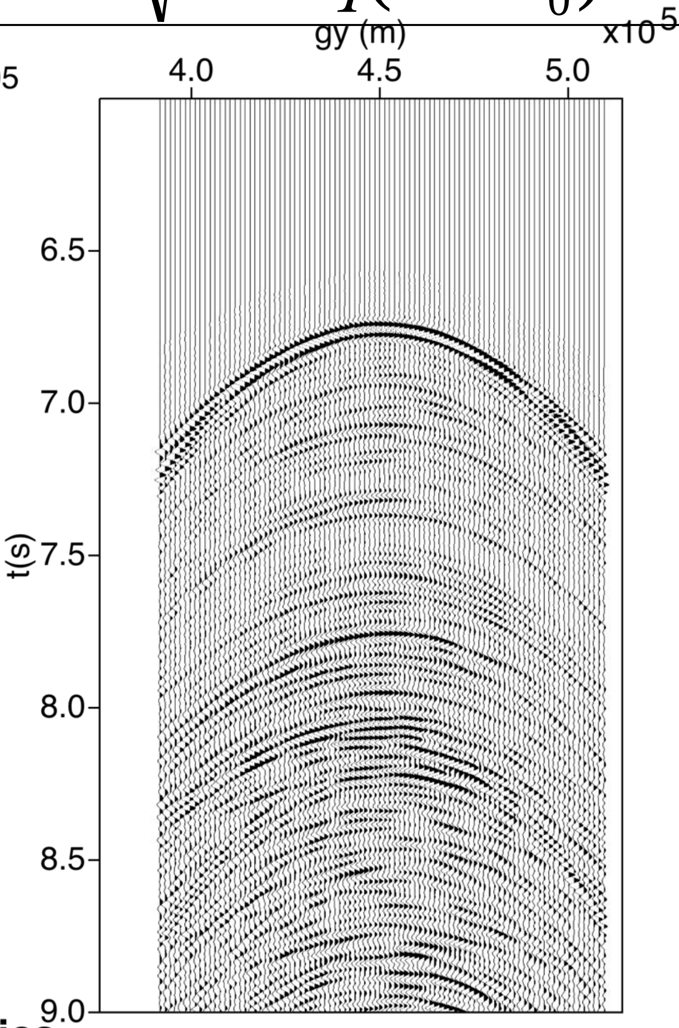
$$t = \sqrt{t^2 + q(h - h_0)^2}$$



DATA



ASRT - 21 Apexes, 5 Velocities

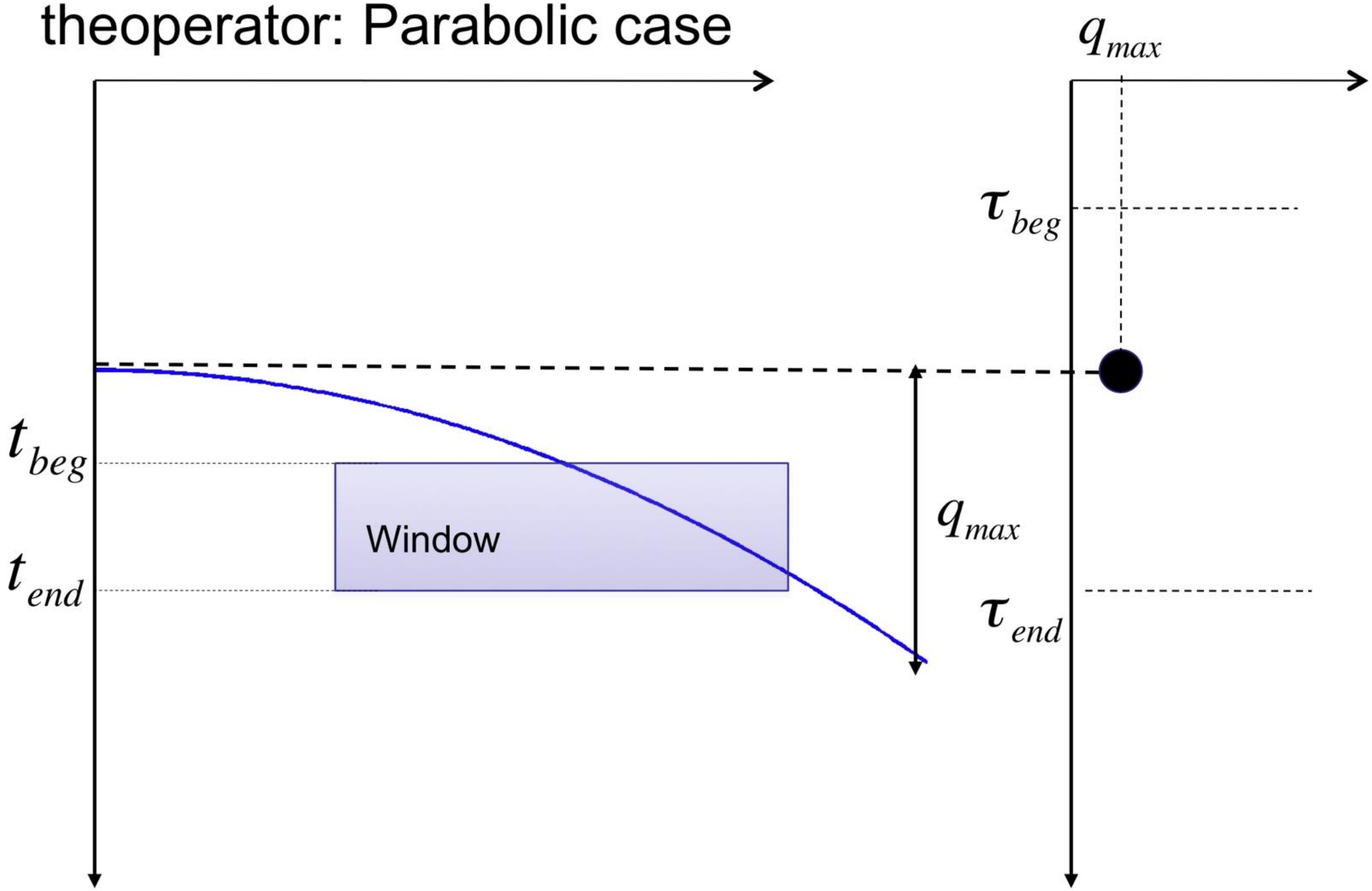


RECOVERED DATA

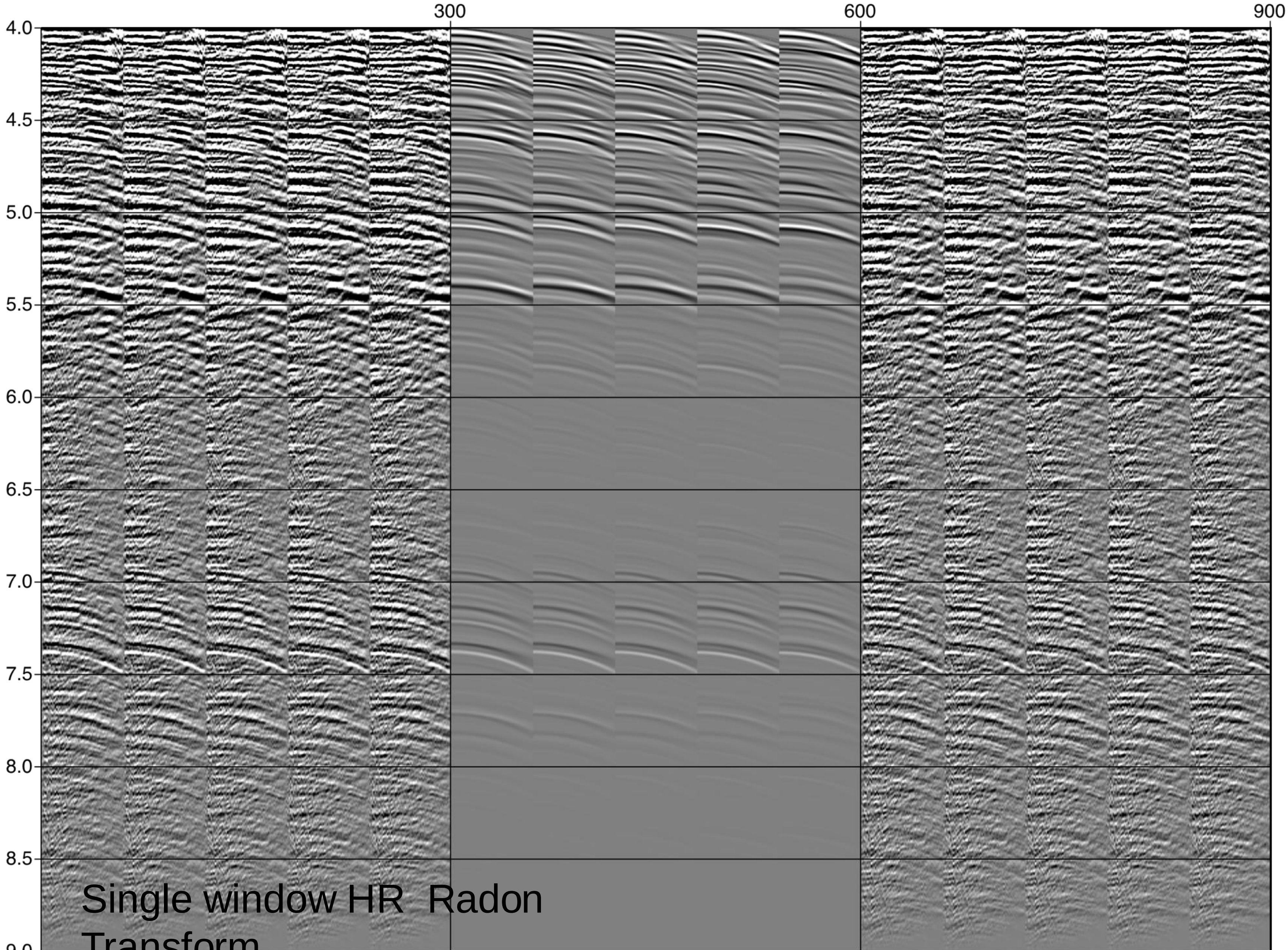
Warning: Sparsity works better in small windows

(similar problem in MED-Deconv, Sparse decon, etc)

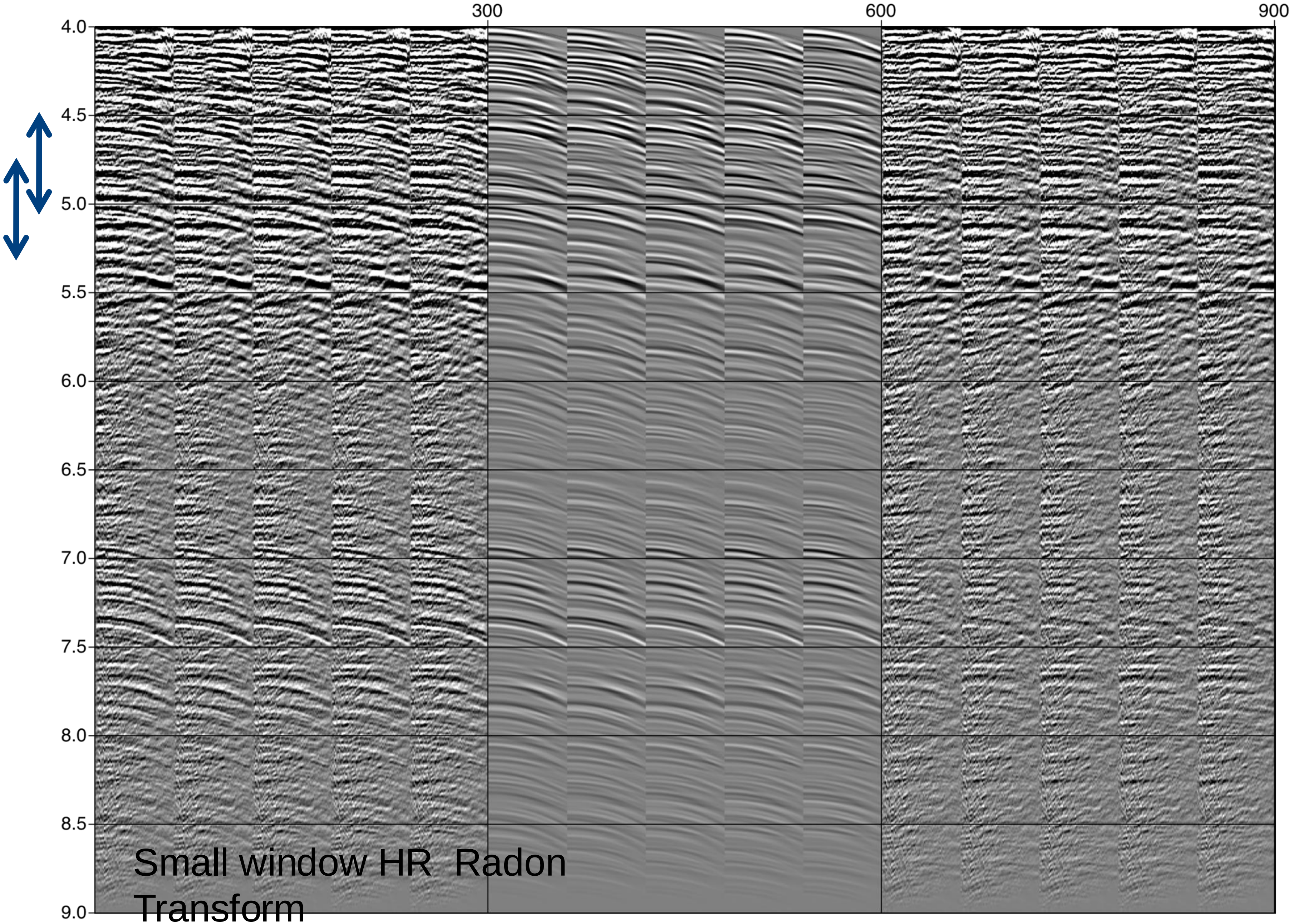
Parameterization of theoperator: Parabolic case



The Radon window should be bigger than the data window



Single window HR Radon Transform



Small window HR Radon Transform

Hybrid Radon Transform

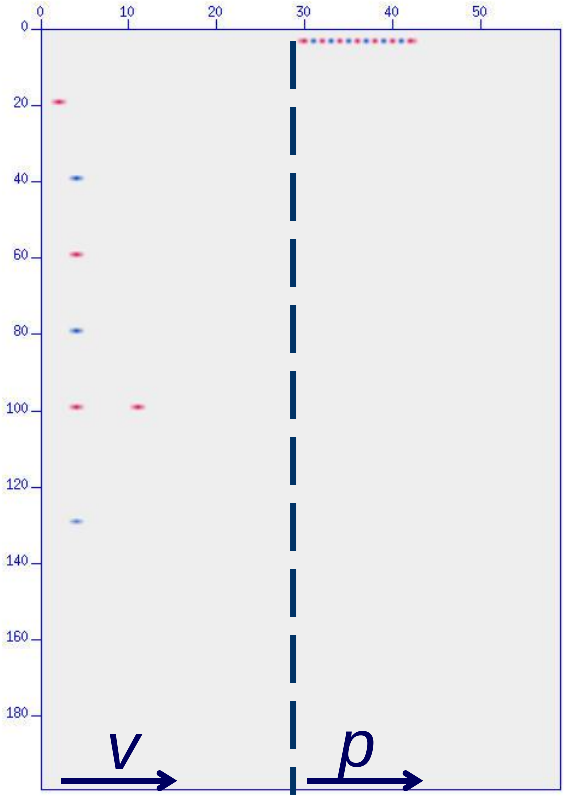
Like a dictionary of Radons

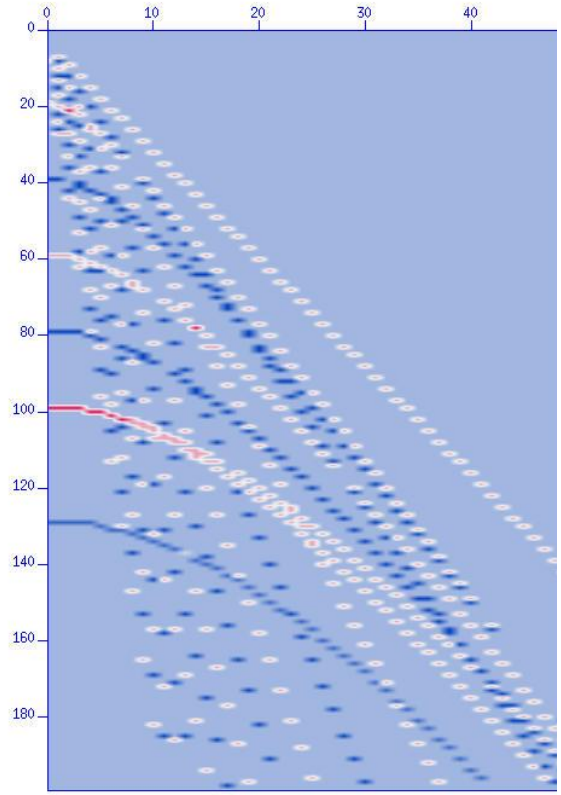
$$d = \begin{pmatrix} L_h & L_l \end{pmatrix} \begin{pmatrix} m_h \\ m_l \end{pmatrix} + n$$

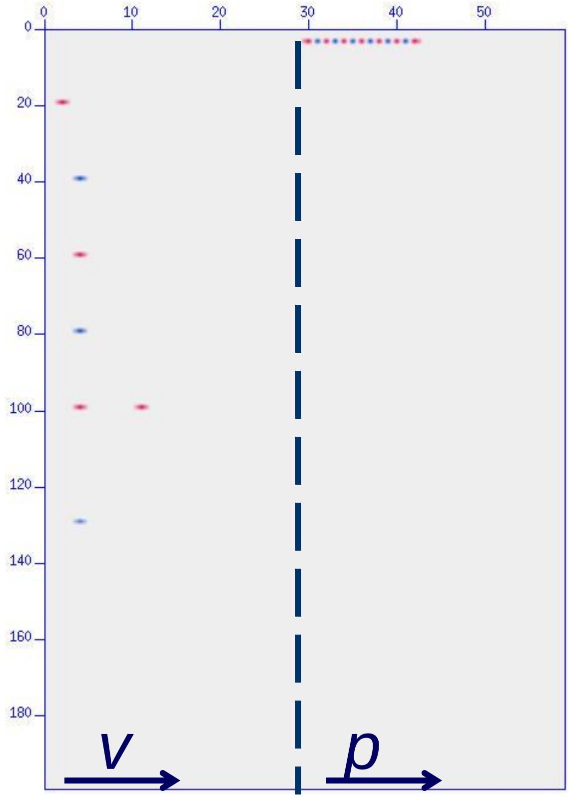
$$= Lm + n$$

Trad, Sacchi and Ulrych, 2001, JSE

Hybrid Radon Transform





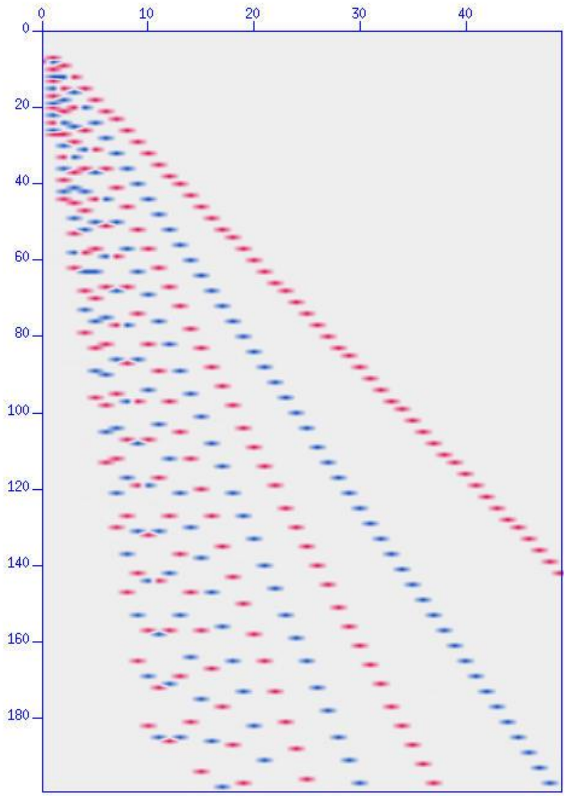


m

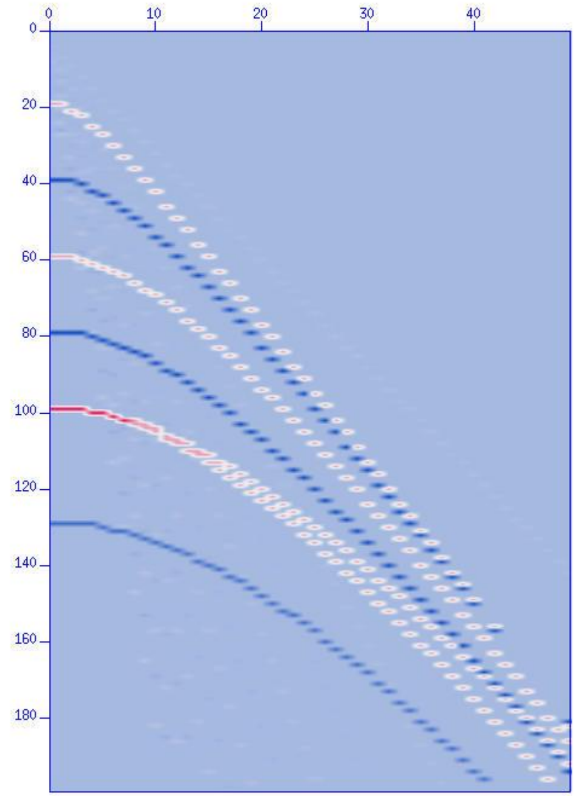
d

Inverted m
Sparse Inversion

Hybrid Radon Transform

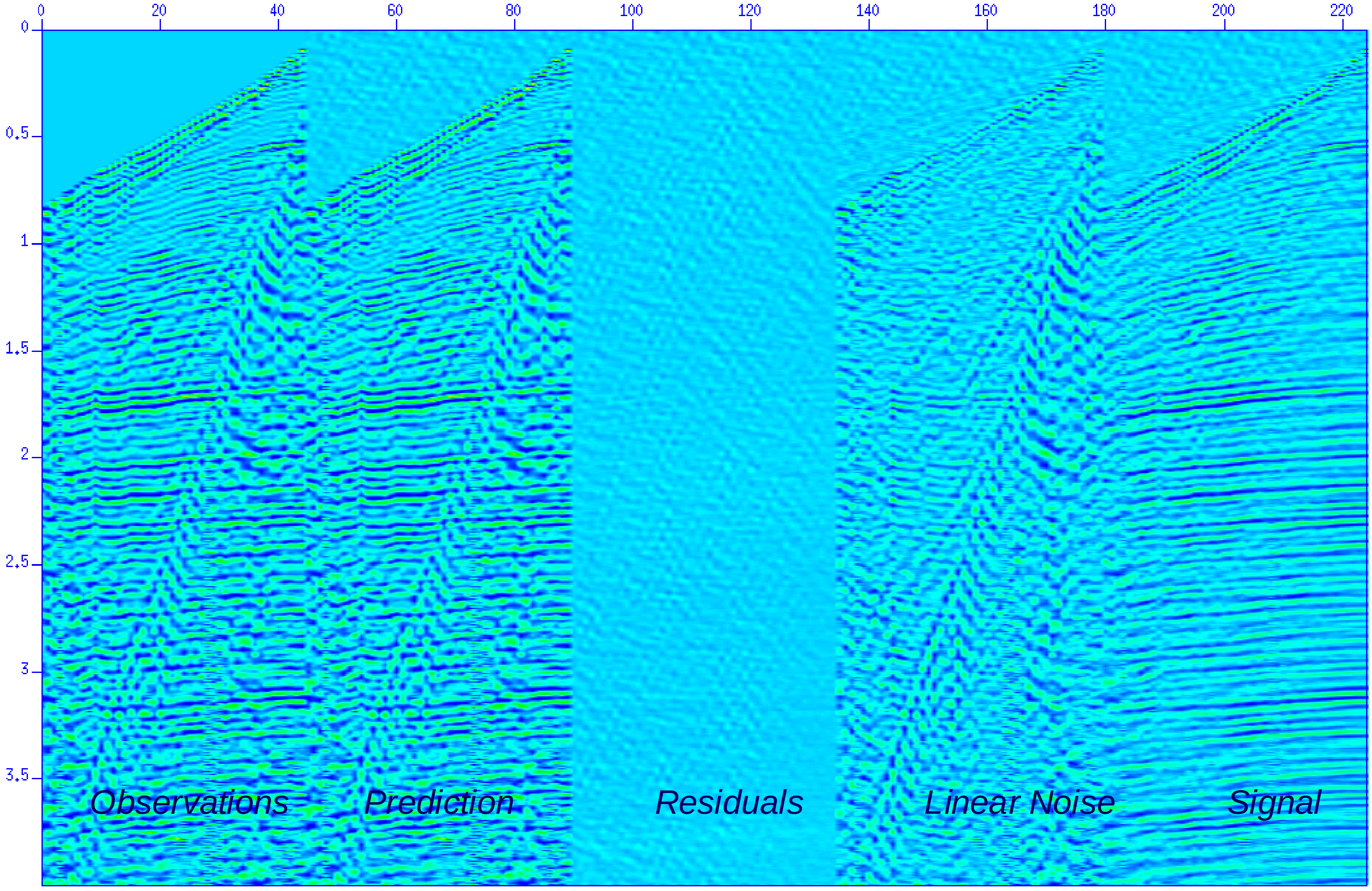


Recovered Model of linear events



Recovered model of hyperbolic events

Hybrid Radon



The Linear Event Model: FX Decon and Cazdow / SSA filters

Mauricio Sacchi
University of Alberta, Edmonton, AB, Canada

Deconvolution in F-X

- Why f-x?

- Linear events in t-x are predictable in f-x

A Model for FX data (One event)

- Time Domain Delay U Frequency Domain Linear Phase

$$s(x,t) = a(t - xp)$$

□

$$S(x,W) = A(W)\, e^{-iWpx}$$

A Model for FX data (One event)

- Time Domain Delay ⇔ Frequency Domain Linear Phase:

$$S(x,\omega) = A(\omega)\, e^{-i\omega p x}$$

$$x = (j-1).\Delta x, \quad \omega = \omega_l$$

$$S_{jl} = A_l\, e^{-i\omega_l\, p (j-1)\Delta x}$$

- Same model as a function of Wave-number:

$$\boxed{\begin{aligned} k_l &= \omega_l\, p \\ S_{jl} &= A_l\, e^{-ik_l (j-1)\Delta x} \end{aligned}}$$

A Model for FX data (One event)

- We now consider the problem as one of predicting data along the spatial coordinate for fix temporal frequency

$$S_j = A e^{-ik\,(j-1)\Delta x}, \quad j = 1, \ldots N$$

- Take for instance a linear event in *t-x* with the above representation in *f-x,* we write*:*

$$\begin{aligned} S_{j-1} &= A e^{-ik\,(j-2)\Delta x} \\ &= A e^{-ik\,(j-1)\Delta x}\, e^{ik\,\Delta x} \\ &= S_j\, e^{ik\,\Delta x} \end{aligned}$$

A Model for FX data (One envent)

- From last expression, we can derive a recursive relation between the signal recorder at receiver *j* as a function of the signal at receiver *j-1*

$$
\begin{aligned}
S_j &= S_{j-1}\, e^{-ik\,\Delta x} \\
&= a S_{j-1}
\end{aligned}
$$

- The last simple expression is the basis for *f-x* prediction/deconvolution and *snr* enhancement

- If the signal can be predicted, then the non-predictable part must be the noise

Canales, L. L., 1984, Random noise reduction: 54th Annual International Meeting, SEG, Expanded Abstracts, Session:S10.1.

Gulunay, N., 1986, FX decon and complex Wiener prediction filter: 56th Annual International Meeting, SEG, Expanded Abstracts, Session:POS2.10.

Gulunay, N., 2000, Noncausal spatial prediction filtering for random noise reduction on 3-D poststack data: Geophysics, 65, 1641-1653.

A Model for FX data (One event)

- From last expression, we can derive a recursive relation between the signal recorder at reciever *j* and a function of the signal at receiver *j-1*

$$S_j = aS_{j-1} + N_j, \quad j = 1 \ldots nx$$

- Determine *a* from available data the estimate noise

$$\left.\begin{aligned} S_2 &= aS_1 + N_2 \\ S_3 &= aS_2 + N_3 \\ S_4 &= aS_3 + N_4 \\ &. \\ &. \\ S_N &= aS_{N-1} + N_{nx} \end{aligned}\right\} \mathbf{s} = \mathbf{Ma} + \mathbf{n}$$

System of equations from where one can estimate *a* and the noise (Least Squares!)

A Model for FX data (*NP* events)

- We can prove that if the data consists of *NP* linear events

$$s(x,t) = \sum_{n=1}^{NP} a(t - p_n x)$$

$$\Updownarrow$$

$$S(x,\omega) = \sum_{n=1}^{NP} A(\omega)\, e^{-i\omega p_n x}$$

$$x = (j-1).\Delta x, \quad \omega = \omega_l$$

$$S_{j\,l} = \sum_{n=1}^{NP} A_l\, e^{-i\omega_l\, p_n (j-1)\Delta x}$$

A Model for FX data (*NP* events)

- We can prove that if the data consists of *NP* linear events

$$S_j = \sum_{n=1}^{NP} A\, e^{-K_n (j-1)\Delta x}$$

$$= \underbrace{a_1 S_{j-1} + a_2 S_{j-2} + a_3 S_{j-3} + \cdots + a_{NP} S_{j-NP}}_{\text{Recursion of order NP}}$$

- The coefficients of the recursion are also called prediction error coefficients - They are related to the wavenumber of each linear event

A Model for FX data (Adding noise)

LF (not *NP*) coefficients ??

- We now consider also noise:

$$S_j = a_1 S_{j-1} + a_2 S_{j-2} + a_3 S_{j-3} + \cdots + a_{LF} S_{j-LF} + N_j$$

- Suppose $LF=3$ and $nx=24$ channels

Notice that I have added noise

$$\begin{aligned}
S_4 &= a_1 S_3 + a_2 S_2 + a_3 S_1 + N_4 \\
S_5 &= a_1 S_4 + a_2 S_3 + a_3 S_2 + N_5 \\
S_6 &= a_1 S_5 + a_2 S_4 + a_3 S_3 + N_6 \\
&\dots \\
S_{24} &= a_1 S_{23} + a_2 S_{22} + a_3 S_{21} + N_{24}
\end{aligned}$$

A Model for FX data (Using the available information)

Ignoring non-recorded data (Using available information *only* !)

$$
\begin{aligned}
S_4 &= a_1 S_3 + a_2 S_2 + a_3 S_1 + N_4 \\
S_5 &= a_1 S_4 + a_2 S_3 + a_3 S_2 + N_5 \\
S_6 &= a_1 S_5 + a_2 S_4 + a_3 S_3 + N_6 \\
&\ldots \\
S_{24} &= a_1 S_{23} + a_2 S_{22} + a_3 S_{21} + N_{24}
\end{aligned}
$$

$\Rightarrow$

$$
\left.
\begin{pmatrix} S_4 \\ S_5 \\ S_6 \\ \vdots \\ N_{24} \end{pmatrix}
=
\begin{pmatrix} S_3 & S_2 & S_1 \\ S_4 & S_3 & S_2 \\ S_5 & S_4 & S_3 \\ \vdots & \vdots & \vdots \\ S_{23} & S_{22} & S_{21} \end{pmatrix}
\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}
+
\begin{pmatrix} N_4 \\ N_5 \\ N_6 \\ \vdots \\ N_{24} \end{pmatrix}
\right\} \quad \mathbf{s} = \mathbf{M}\mathbf{a} + \mathbf{n}
$$

A Model for FX data (Assuming data outside aperture=0)

Considering non-recorded data = *0*

$$
\begin{aligned}
S_2 &= a_1 S_1 + N_2 \\
S_3 &= a_1 S_2 + a_2 S_1 + N_3 \\
S_4 &= a_1 S_3 + a_2 S_2 + a_3 S_1 + N_4 \\
S_5 &= a_1 S_4 + a_2 S_3 + a_3 S_2 + N_5 \\
S_6 &= a_1 S_5 + a_2 S_4 + a_3 S_3 + N_6
\end{aligned}
$$

□

...

$$
\begin{aligned}
S_{24} &= a_1 S_{23} + a_2 S_{22} + a_3 S_{21} + N_{24} \\
0 &= a_1 S_{24} + a_2 S_{23} + a_3 S_{22} + N_{25} \\
0 &= a_2 S_{24} + a_3 S_{23} + N_{26} \\
0 &= a_3 S_{24} + N_{27}
\end{aligned}
$$

A Model for FX data (Assuming data outside aperture=0)

Considering non-recorded data = *0*

$$
\left.
\begin{pmatrix} S_2 \\ S_3 \\ S_4 \\ S_5 \\ \vdots \\ S_{24} \\ 0 \\ 0 \\ 0 \end{pmatrix}
=
\begin{pmatrix}
S_1 & 0 & 0 \\
S_2 & S_1 & 0 \\
S_3 & S_2 & S_1 \\
S_4 & S_3 & S_2 \\
S_5 & S_4 & S_3 \\
\vdots & \vdots & \vdots \\
S_{24} & S_{23} & S_{22} \\
0 & S_{24} & S_{23} \\
0 & 0 & S_{24}
\end{pmatrix}
\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}
+
\begin{pmatrix} N_2 \\ N_3 \\ N_4 \\ N_5 \\ \vdots \\ N_{24} \\ N_{25} \\ N_{26} \\ N_{27} \end{pmatrix}
\right\} \quad \mathbf{s} = \mathbf{C}\mathbf{a} + \mathbf{n}
$$

A Model for FX data (*NP* events)

Two formulations: $\mathbf{s} = \mathbf{Ma} + \mathbf{n}$ (Transient free case)

$\mathbf{s} = \mathbf{Ca} + \mathbf{n}$ (Toeplitz case)

Two Least Squares Solutions:

$$\hat{\mathbf{a}}_1 = (\mathbf{M}^H \mathbf{M})^{-1} \mathbf{M}^H \mathbf{s}$$

$\mathbf{M}^H \mathbf{M}$ is a Non-Toeplitz correlation data matrix

$$\hat{\mathbf{a}}_2 = (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{C}^H \mathbf{s}$$

$\mathbf{C}^H \mathbf{C}$ is the Toeplitz correlation data matrix

A Model for FX data (Algorithm)

Algorithm to recover the noise

1 - Original data in t - x $\rightarrow$ f - x

2 -

For each frequency :

Find $\hat{\mathbf{a}}$ that solves $\mathbf{s} = \mathbf{M}\mathbf{a} + \mathbf{n}$,

Use $\hat{\mathbf{a}}$ to predict the noise $\hat{\mathbf{n}} = \mathbf{s} - \mathbf{M}\hat{\mathbf{a}}$

End for

3 - f - x $\rightarrow$ t - x

4 - Substract noise from original data

A Model for FX data (Trade-off parameters)

- *NP* does not define the filter length *LF* *(*Number of coefficients of the prediction error operator ***a***.)

  LP=LF only if the data is a superposition of *NP* linear events and *noise=0*

- In general, events in the data are approximated by linear events. In addition, data are contaminated with random noise, then we choose *LF* > *NP*

- Similar to the deconvolution problem, we can also add a stability factor:

$$\hat{\mathbf{a}} = (\mathbf{M}^{H}\mathbf{M} + \mu\mathbf{I})^{-1}\mathbf{M}^{H}\mathbf{s}$$

A Model for FX data (Trade-off parameters)

- *LF* and n control the degree of noise that is modeled
- Long operators will attempt to model the signal and the noise and therefore, the noise estimate will be contaminated by the signal
- Short filters will only partially model the noise

- FX DECON needs small overlapping windows. Why?.. Because assumes windows with constant dip data…… and a small number of dips. OK!!

Why the name F-X **Deconvolution** ?

$$S_j = a_1 S_{j-1} + a_2 S_{j-2} + a_3 S_{j-3} + \cdots + a_{LF} S_{j-LF} + N_j$$

or

$$S_j - a_1 S_{j-1} - a_2 S_{j-2} - a_3 S_{j-3} - \cdots - a_{LF} S_{j-LF} = N_j$$

$$f_0 = 1$$
$$f_k = -a_k, k = 1, \ldots, LF$$

Þ

$$f_0 S_j + f_1 S_{j-1} + f_2 S_{j-2} + f_3 S_{j-3} + \cdots + f_{LF} S_{j-LF} = N_j$$

$$\boxed{f * S = N}$$

f predicts white noise - in spiking deconvolution *f* also predicts white noise (the reflectivity)

Prediction error filter - A wave number view

S : spatial signal with one or more dominant wavenumber k

N : spatial white noise (same energy at all wavenumbers)

$$f * S = N \text{ Þ}$$

f : Will have to filter the dominant signals and let the noise pass (Notch filter?)

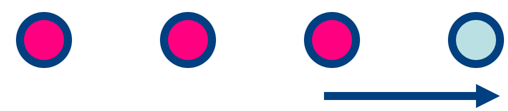
Fxy Filters

- 2D Kernels - same idea but not clear to what kind of signal model one requires to justify 2D prediction error filters

- 1D Prediction can be performed with Forward or backward operators or a combination of both

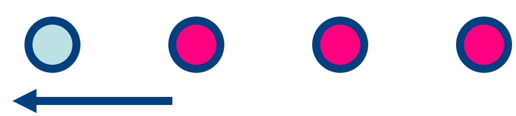
Predicting from the left :

$$S_j = a_1 S_{j-1} + a_2 S_{j-2} + a_3 S_{j-3} + \cdots + a_{LF} S_{j-LF} + N_j$$



Predicting from the right :

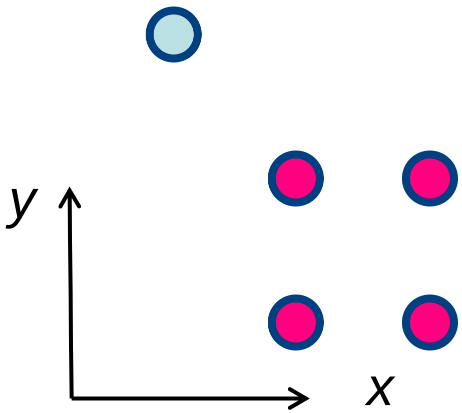
$$S_j = b_1 S_{j+1} + b_2 S_{j+2} + b_3 S_{j+3} + \cdots + b_{LF} S_{j+LF} + N_j$$



Fxy Filters

2D Prediction:

$$S_{ij} = a_{11}S_{i-1\,j-1} + a_{12}S_{i-1\,j-2} + a_{12}S_{i-2\,j-1} + a_{22}S_{i-2\,j-2} + \cdots + N_{ij}$$



Solution: could use Block Matrices and Block Solvers or operate on the flight with the Method of Conjugate Gradients

SSA Singular Spectrum Analysis or Cadzow Filter

- SSA: an alternative to FX deconvolution ?
- ++
  - Easy to code and to understand
  - Easy to extend to ND
  - Possibility of creating noise attenuation methods that simultaneously operate in frequency and space
- -+
  - Memory?

Note:

SSA arises from people working in dynamical systems as a method for rank reduction of 1D time series. Cadzow decomposition is a more general approach to image/signal enhancement that leads to an algorithm very similar to SSA when used to enhance complex sinusoids.

References

- S. Trickett, 2008,F-xy Cadzow noise suppression, CSPG CSEG Convention
- Broomhead, D.S., and King, G., 1986: Extracting qualitative dynamics from experimental data, Physica D, **20, 217-236.**
- Vautard, R., and Ghil, M., 1989: Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series, Physica D, **35, 395-424.**
- V Oropeza and M D Sacchi, 2011, Simultaneous seismic data de-noising and reconstruction via Multichannel Singular Spectrum Analysis (MSSA), Geophysics, 76, V25-V32.

Theory (an easy tour of SSA)

Assume we are in FX working with data in x for fixed f

$$\mathbf{M} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2 & s_3 & s_4 & s_5 \\ s_3 & s_4 & s_5 & s_6 \\ s_4 & s_5 & s_6 & s_7 \end{pmatrix}$$

Hankel Matrix of a signal of length 7 with a view of length $M=4$ (Trajectory Matrix)

$$\mathbf{M} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2 & s_3 & s_4 & s_5 \\ s_3 & s_4 & s_5 & s_6 \\ s_4 & s_5 & s_6 & s_7 \end{pmatrix}$$

Hankel Matrix of a signal of length 7 with a view of length $M=4$ (Trajectory Matrix)

Average Along anti-diagonals

$$A(\mathbf{M}) = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix}$$

Linear event in *TX* = Complex Sinusoid in *FX*

$$s(t,x) = w(t - px)$$

$$S(\omega,x) = W(\omega)e^{-i\omega px}$$

$$S(\omega, x) = W(\omega)e^{-i\omega p x}$$

$$x = n\,\Delta x$$

$$S_n = W\,e^{-i\alpha n}$$

$$\boxed{S_n = P S_{n-1}}$$

Prediction operator of length 1 is able to propagate signal from channel *n-1* to channel *n.*

Single linear event in *FX*

$$S_n = PS_{n-1}$$

$$\mathbf{M} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2 & s_3 & s_4 & s_5 \\ s_3 & s_4 & s_5 & s_6 \\ s_4 & s_5 & s_6 & s_7 \end{pmatrix}$$

Single linear event in *FX*

$$S_n = P S_{n-1}$$

$$\mathbf{M} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2 & s_3 & s_4 & s_5 \\ s_3 & s_4 & s_5 & s_6 \\ s_4 & s_5 & s_6 & s_7 \end{pmatrix} = \begin{pmatrix} s_1 & P s_1 & P^2 s_1 & P^3 s_1 \\ s_2 & P s_2 & P^2 s_2 & P^3 s_2 \\ s_3 & P s_3 & P^2 s_3 & P^3 s_3 \\ s_4 & P s_4 & P^2 s_4 & P^3 s_4 \end{pmatrix}$$

$\mathbf{M}$ is a Hankel matrix of $Rank = 1$

p linear events

- It can be shown that for a superposition of p complex sinusoids (p events of different slope in FX):

$$\boxed{\mathbf{M} \text{ is a Hankel matrix of } Rank = p}$$

- **We will use rank reduction to attenuate random noise**

- **The spectra of singular values can be used to determine the number of complex sinusoids (plane waves) in the window of analysis**

SSA *FX* algorithm = Rank Reduction of $\mathbf{M}$

$$s(t,x) \rightarrow s(\omega,x) \rightarrow \mathbf{s}(\omega)$$

Select p

Select $M(view)$

For $\omega = \omega_{\min}, \omega_{\max}$

$\quad \mathbf{M} = Hankel[\mathbf{s}(\omega)]$

$\quad \mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^H$ SVD

$\quad \mathbf{M}_p = \mathbf{U}_p\mathbf{U}_p^H\mathbf{M}$ SVD - Filtering to reduce rank

$\quad \mathbf{s}_p(\omega) = A(\mathbf{M}_p)$ Sum along anti - diags

End

$$\mathbf{s}_p(\omega) \rightarrow s_p(\omega,x) \rightarrow s_p(t,x)$$

The Cazdow/SSA filter

- Hankelization Operator

- Rank Reduction Operator

$$\mathbf{M}_p = \mathbf{U}_p \mathbf{U}_p^H \mathbf{M}$$
$$= R_p[\mathbf{M}]$$

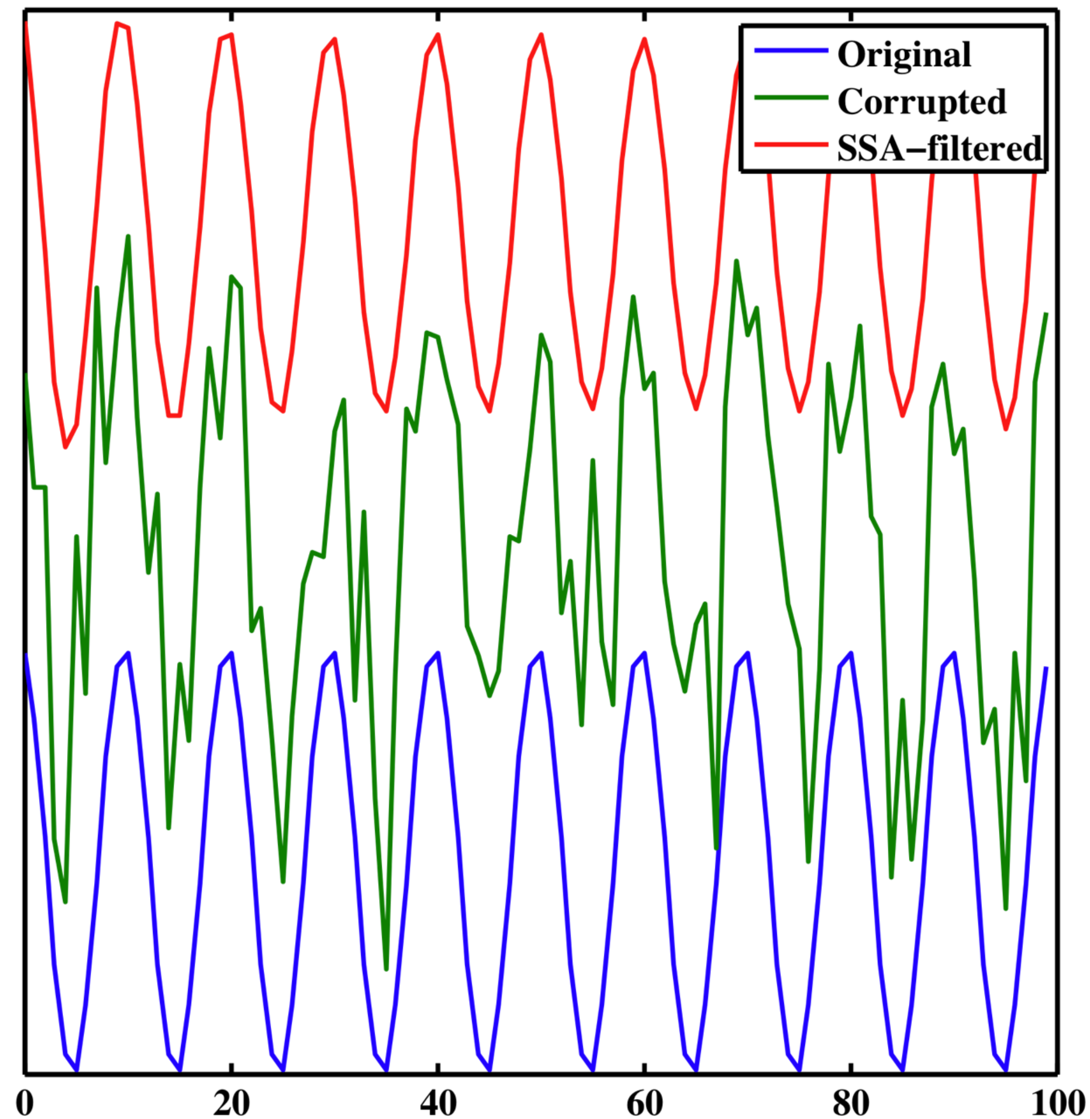
- Anti-diagonal Averaging Operator

$$\mathbf{s} = H[\mathbf{M}]$$

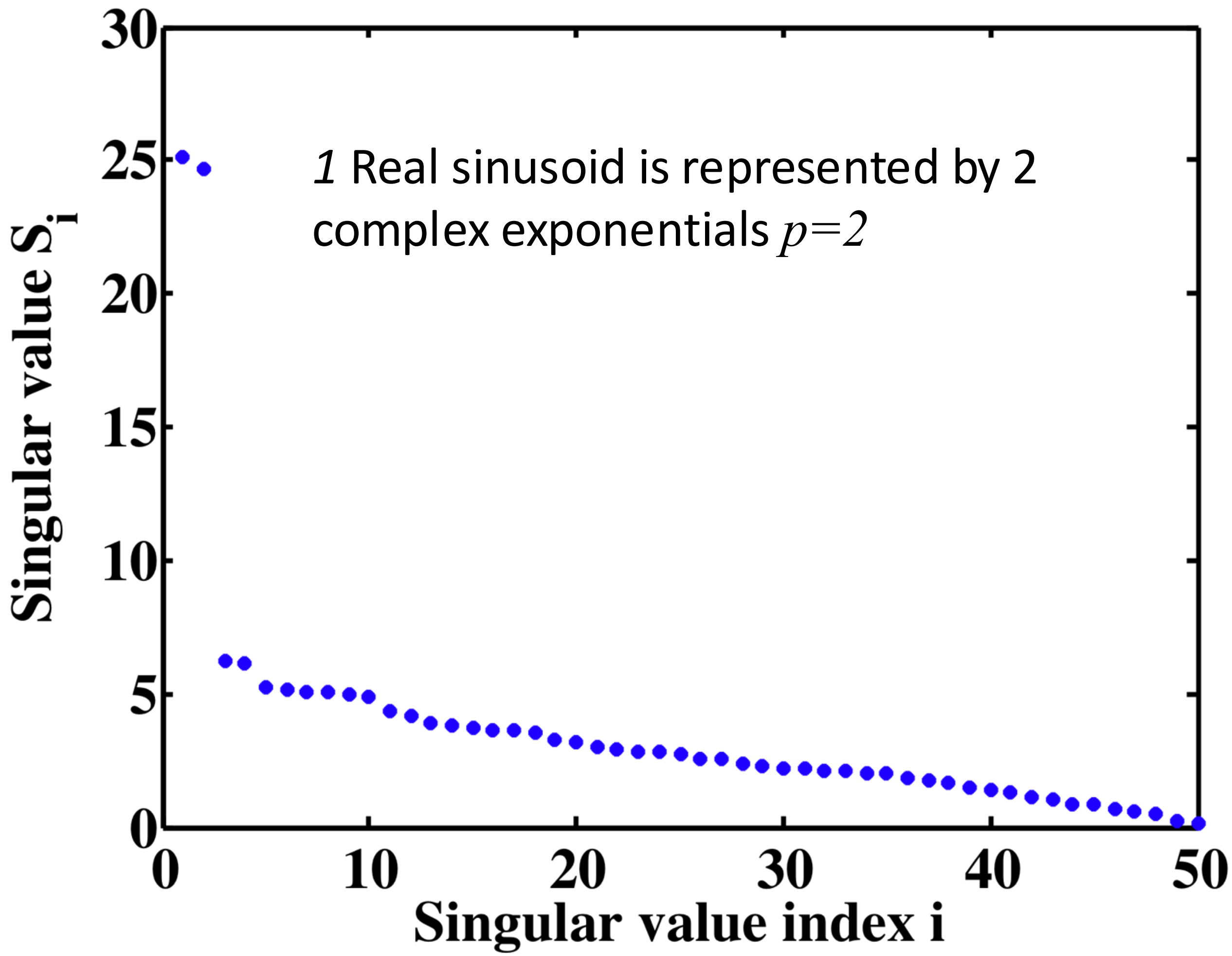
- Cazdow/SSA Filter

$$\hat{\mathbf{s}} = A[R_p[H[s]]]$$

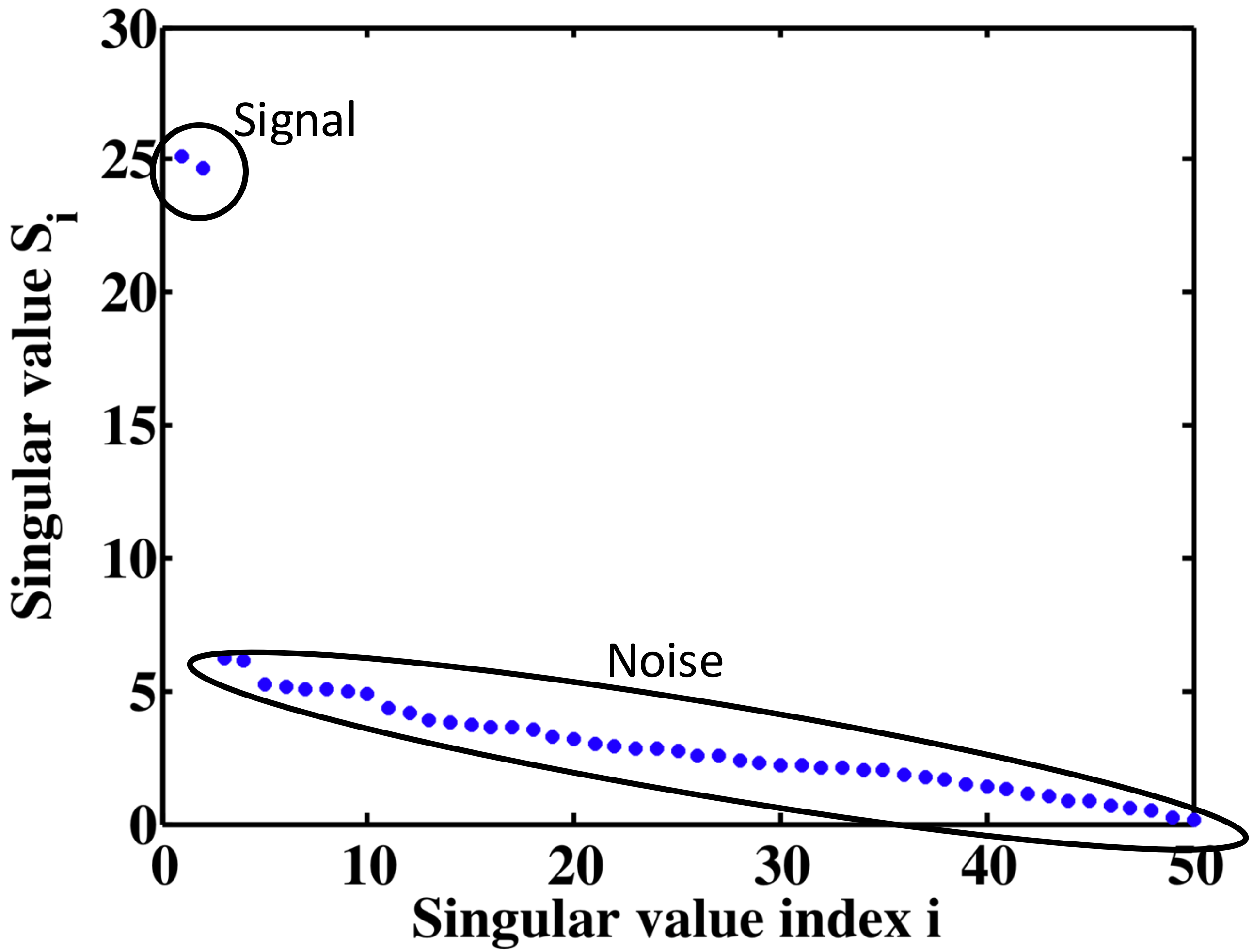
1D Example



1D example


30
25
20
15
10
5
0
0
10
20
30
40
50
Singular value S_i
Singular value index i
1 Real sinusoid is represented by 2 complex exponentials *p=2*

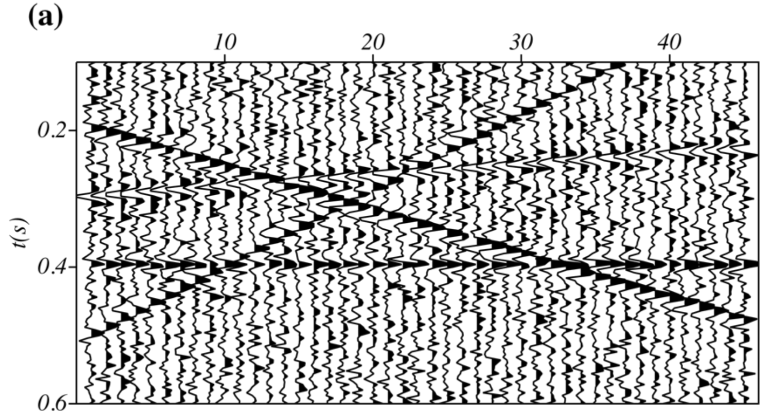
1D example


30
25
20
15
10
5
0
0
10
20
30
40
50
Signal
Noise
Singular value S_i
Singular value index i

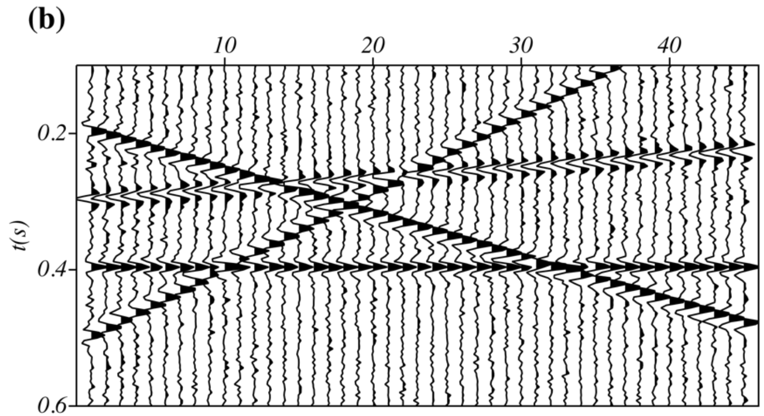
2D example (1D in *FX*)

- M *(view)* is selected to make $\mathbf{M}$ (*Hankel Matrix*) close to a square matrix
- Rest of the algorithm is very similar to *FX- decon*:
  - Decide min-max frequencies for noise attenuation
  - Window patching/unpatching
  - Adjust filter length (p *in ssa)* by monitoring residual panel
  - Use frequency domain symmetries to save time

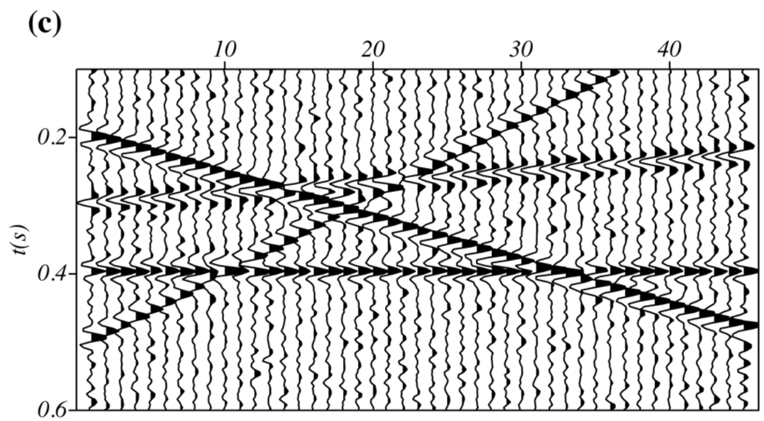
Data



FX SSA



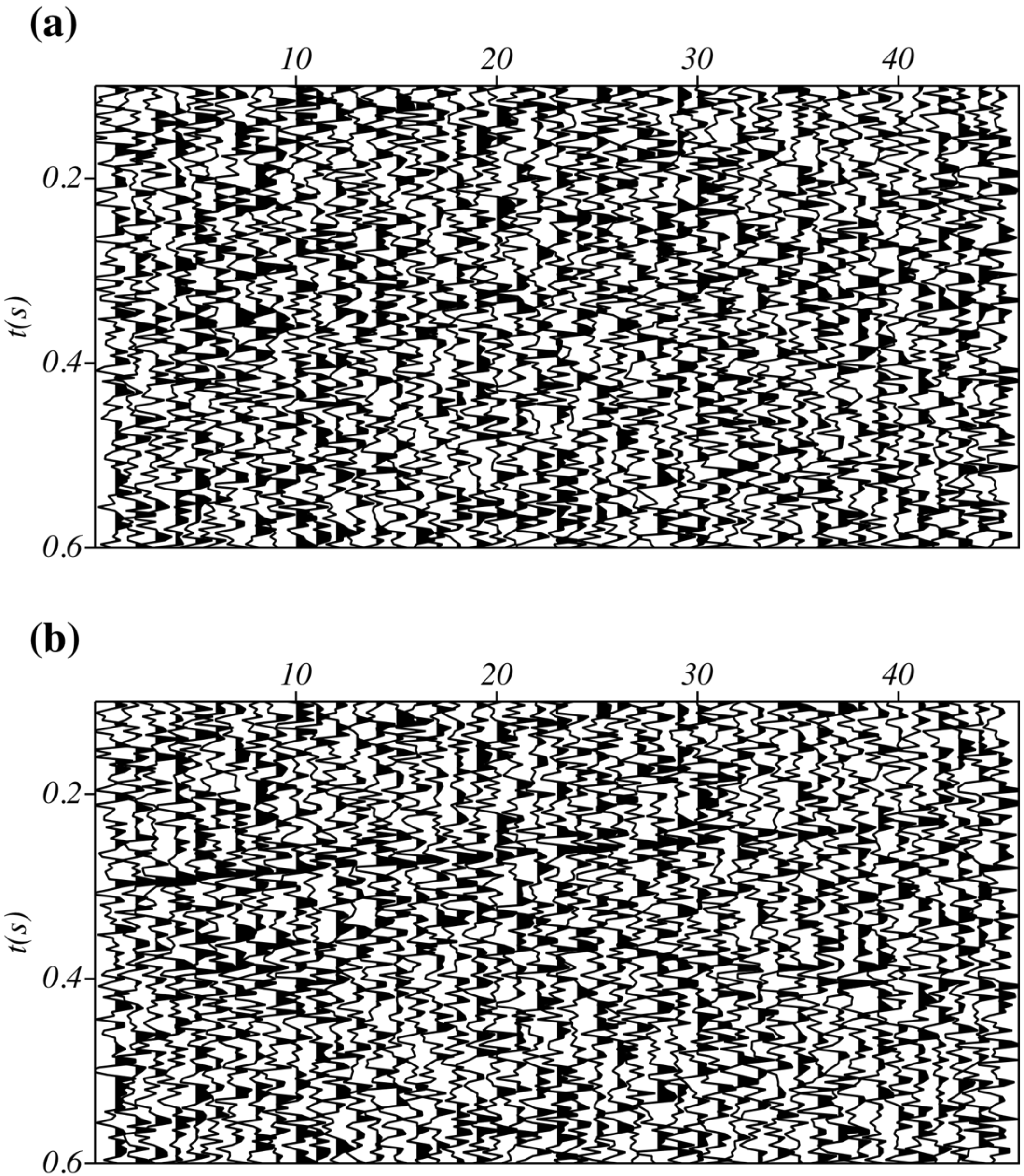
FX Decon



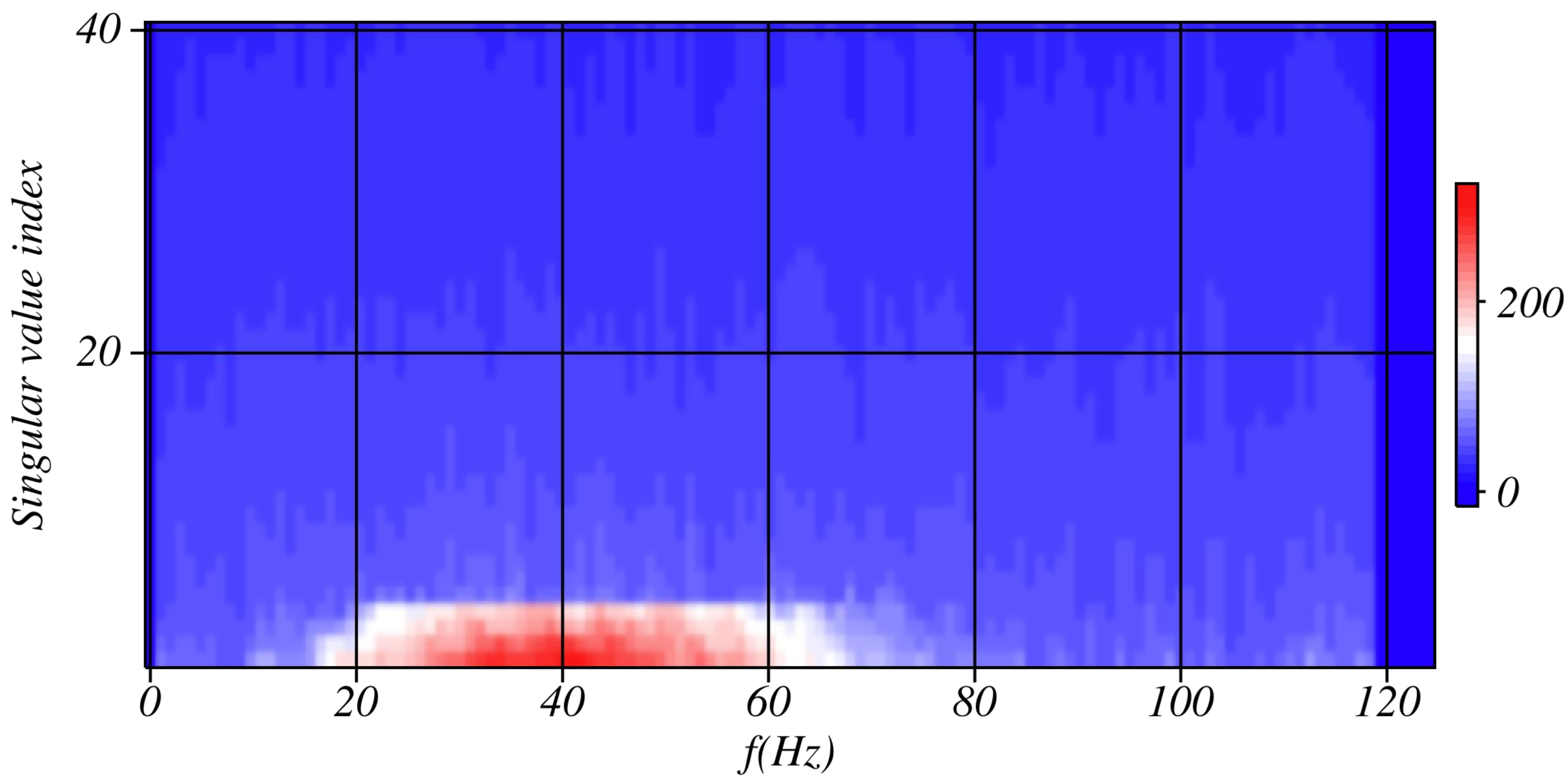
Error Panels

FX-SSA

FX-Decon

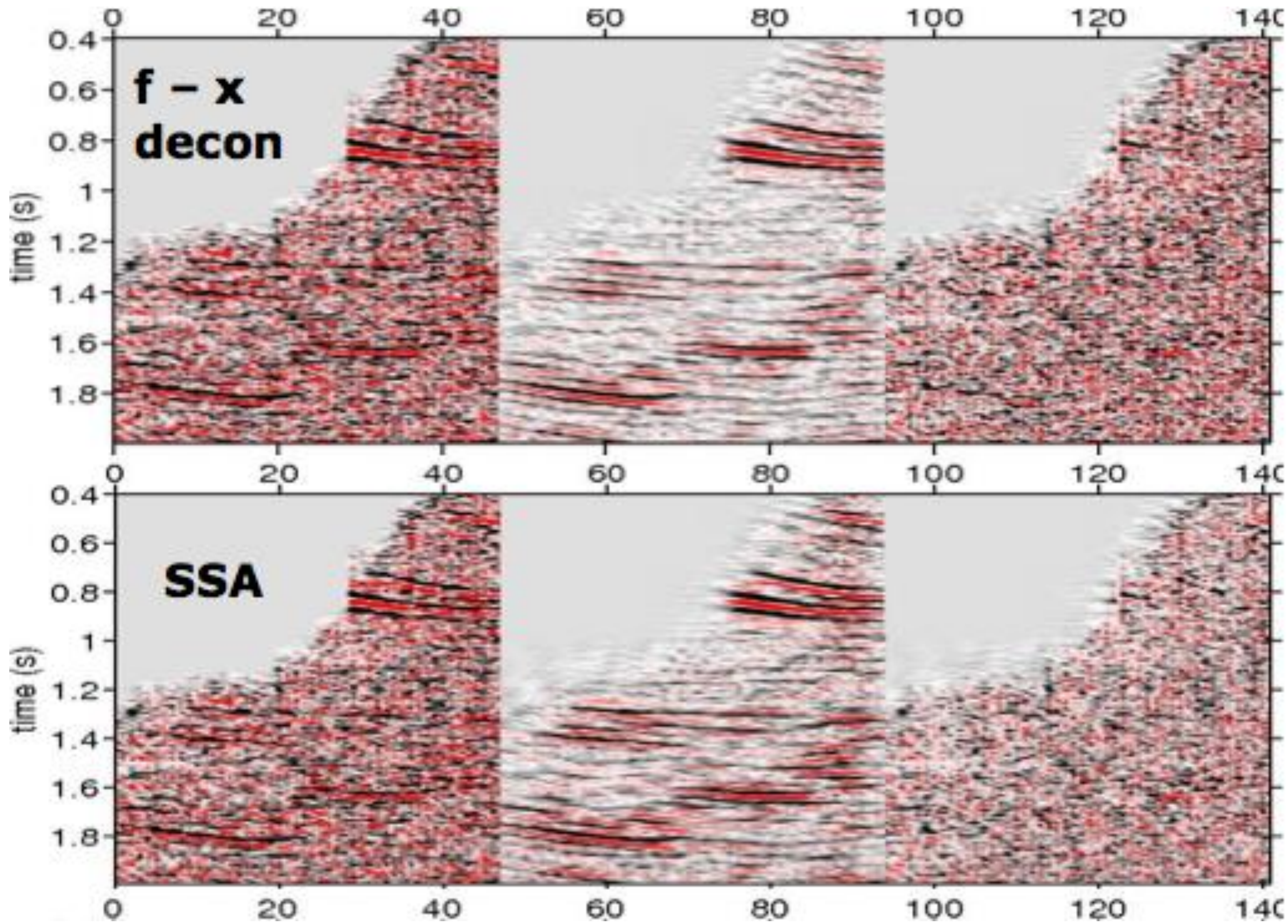


Spectrum of singular values vs. freq.

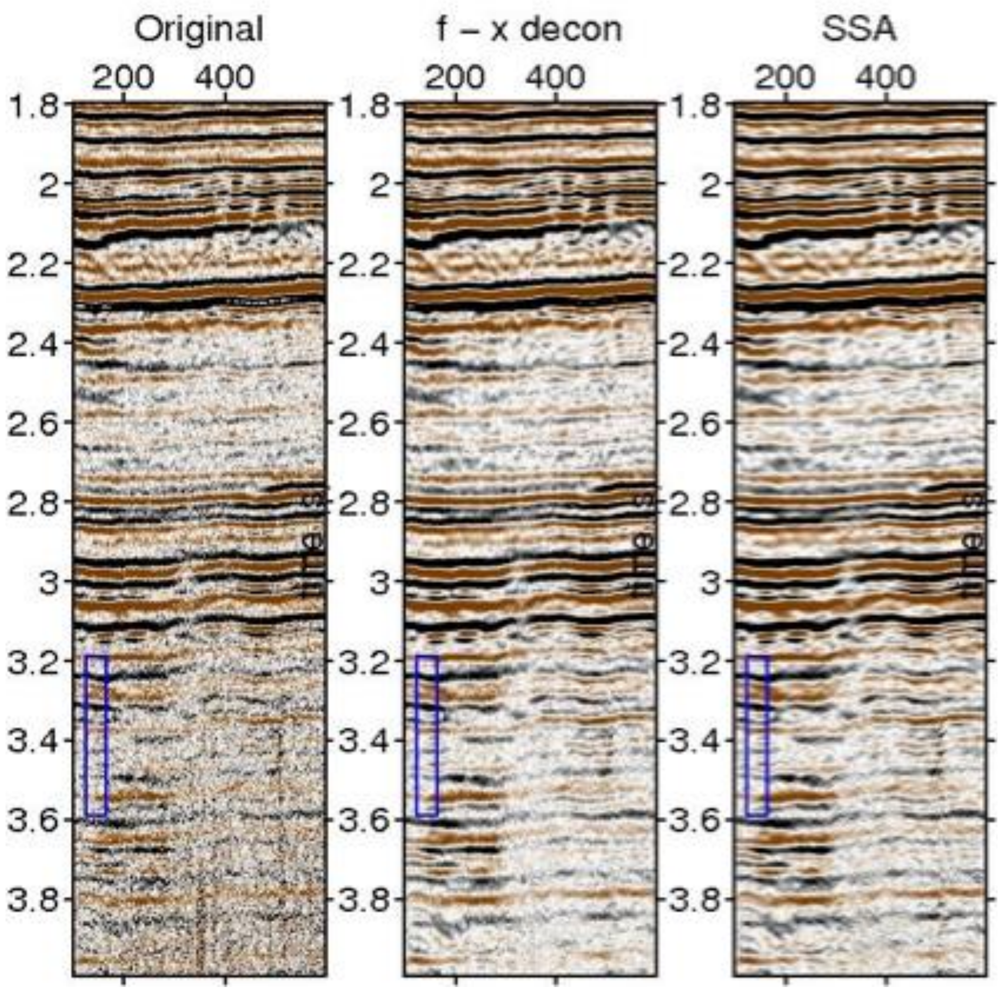


Spectra cut-off is at 4 singular values

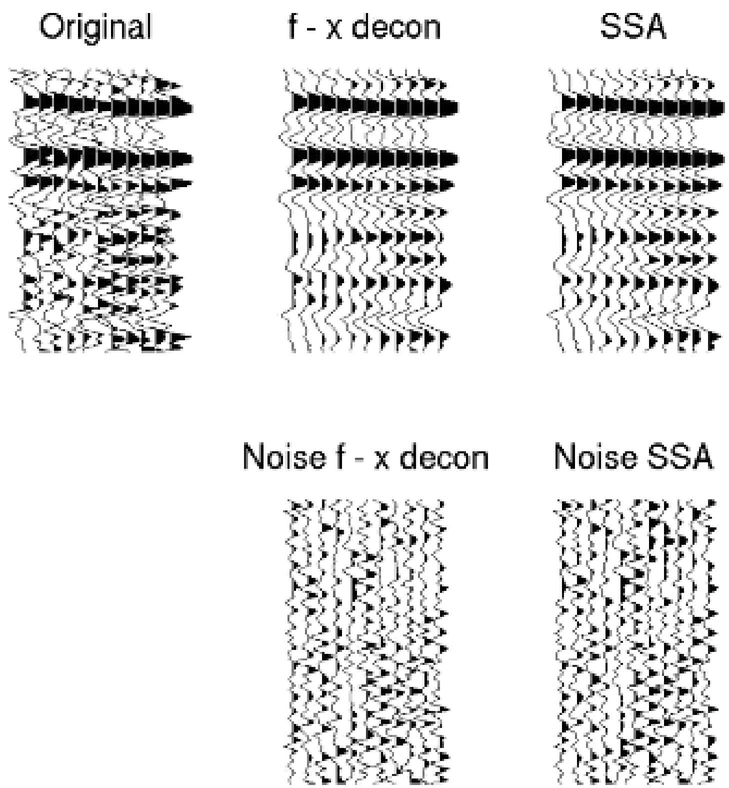
Marmousi gathers.



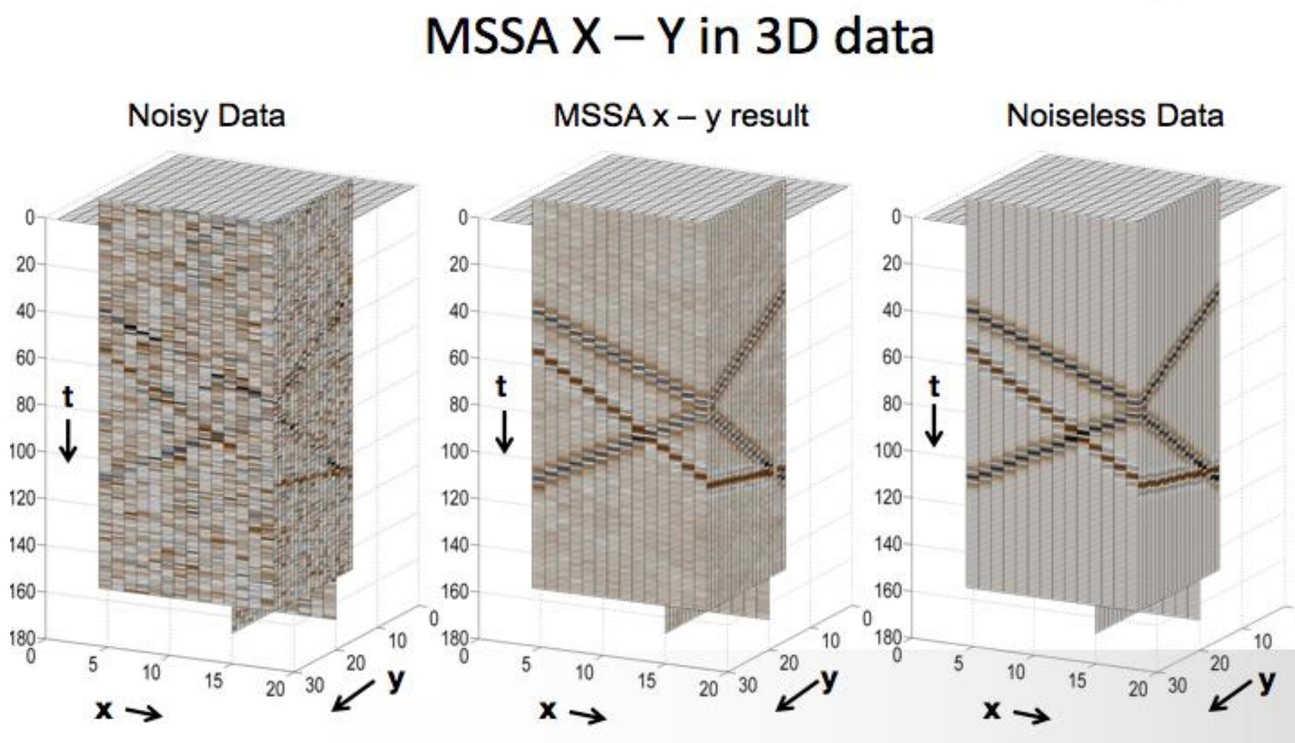
Real data comparison (Processing by V. Oropeza)



Comparison

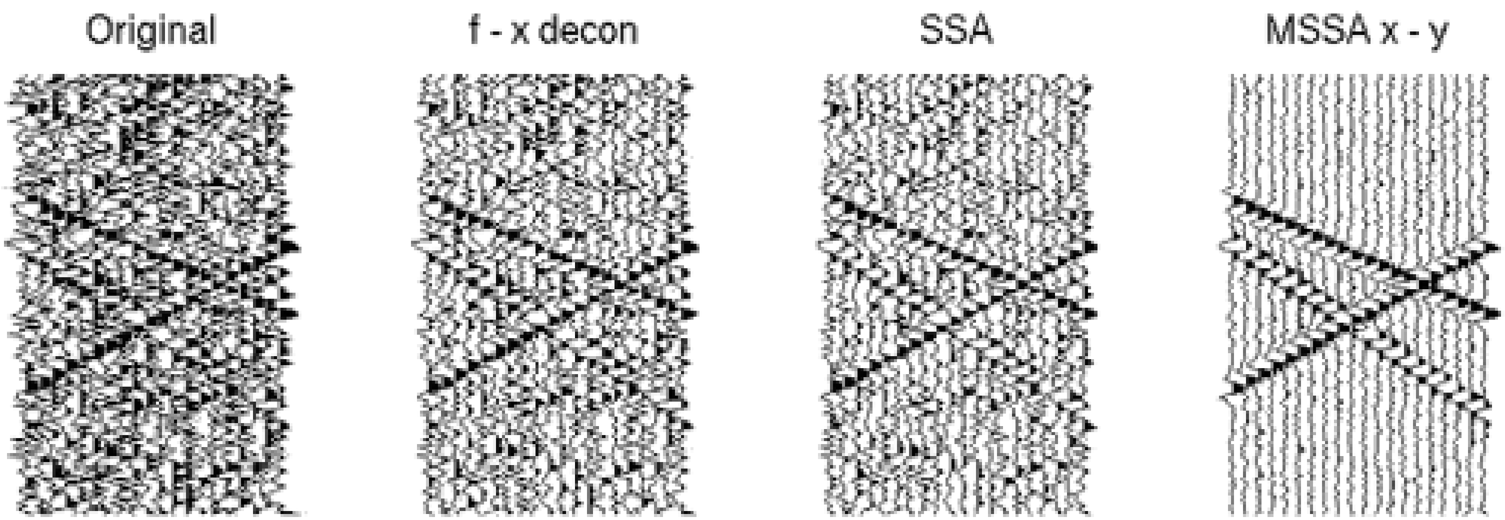


SSA for 3D cubes



SSA works better in *x-y-f* than *x-f* (more data)

MSSA – XY in 3D data using 6 Singular Values



Notes:

- RSVD (randomized SVD) can be used to speed up the SVD

- For data reconstruction we used fast Lanczos solvers that exploits the structure of the Hankel matrix for fast rank reduction

- These tricks are also used for data reconstruction

Gao, J., Sacchi, M., and Chen, X. (2013). ”A fast reduced-rank interpolation method for prestack seismic volumes that depend on four spatial dimensions.” *GEOPHYSICS, 78(1), V21–V30.*