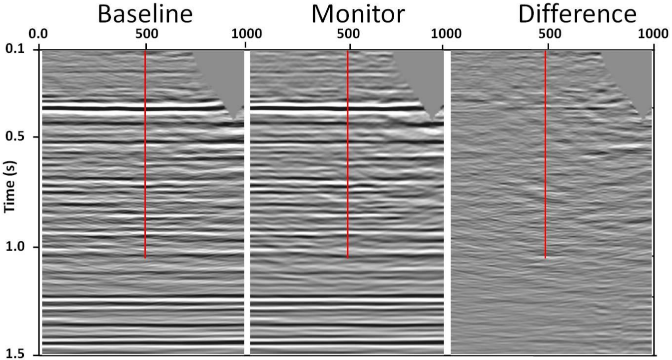
Application of surface-consistent matching filters (SCMF) to time-lapse data set

Mahdi Almutlaq and Gary F. Margrave

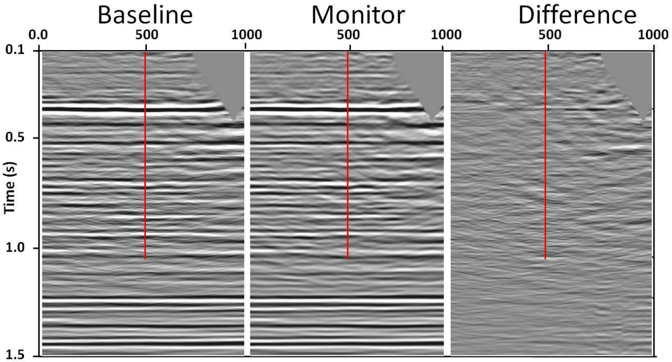
University of Calgary, Department of Geoscience, CREWES

December 02, 2011

What is the purpose of this study?



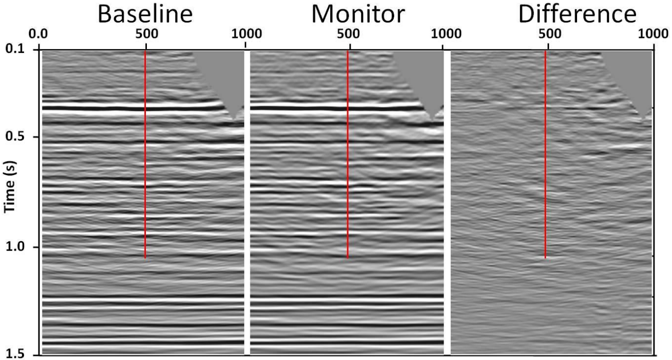
What is the purpose of this study?



Our goal is to design matching filters that

- scale data to same amplitude level

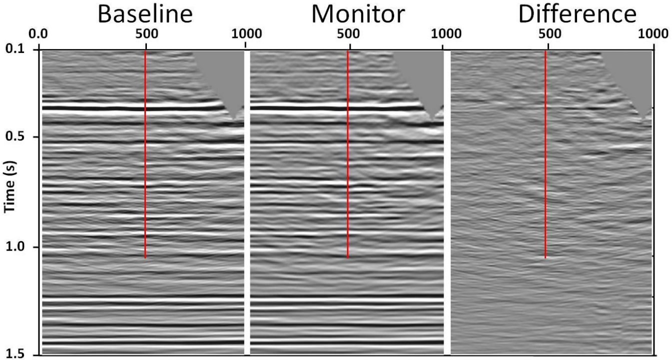
What is the purpose of this study?



Our goal is to design matching filters that

- scale data to same amplitude level
- equalize phase and bandwidth

What is the purpose of this study?



Our goal is to design matching filters that

- scale data to same amplitude level
- equalize phase and bandwidth
- compensate for time-shifts

Outline

Surface-consistent model

Surface-consistent model

The seismic trace can be modeled as

$$d_{ij}(t) = s_i(t) * r_j(t) * h_k(t) * y_l(t)$$

where

- d_{ij}: *seismic trace*
- s_i : *source response at location* i
- r_j : *receiver response at location* j
- h_k: *offset response at location* k; $k = |i - j|$
- y_l : *subsurface response at* l; $l = \frac{(i+j)}{2}$

Matching filters

Time domain:

$$m * s1 = s2 \tag{1}$$

Fourier domain:

$$M(\omega) = \frac{S_2(\omega)}{S_1(\omega)}. \tag{2}$$

Matching filters

Time domain:

$$m * s1 = s2 \tag{1}$$

Fourier domain:

$$M(\omega) = \frac{S_2(\omega)}{S_1(\omega)}. \tag{2}$$

Important remarks:

- a perfect matching filter is a spectral ratio; however
- spectral ratio is unstable in presence of noise; and

Matching filters

Time domain:

$$m * s1 = s2 \tag{1}$$

Fourier domain:

$$M(\omega) = \frac{S_2(\omega)}{S_1(\omega)}. \tag{2}$$

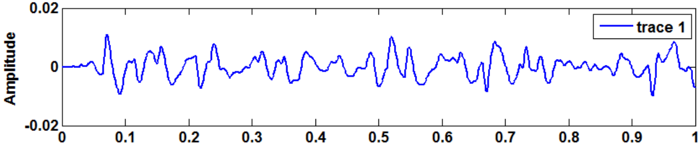
Important remarks:

- a perfect matching filter is a spectral ratio; however
- spectral ratio is unstable in presence of noise; and

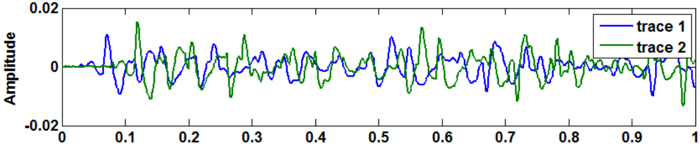
What do we do?

- We solve time-domain (equation 1) in LSQ sense and Fourier transform the solution which is a good approximation to the spectral ratio method.

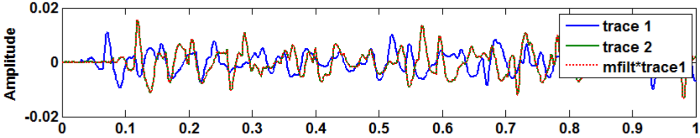
Matching filters



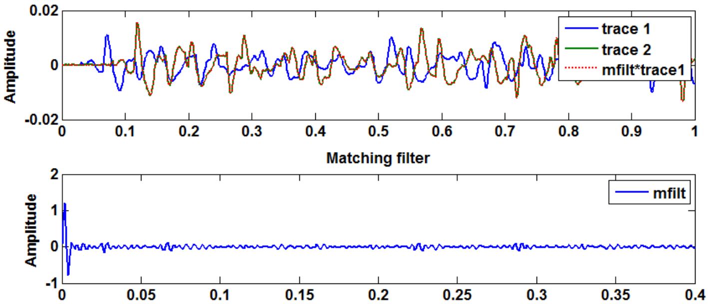
Matching filters



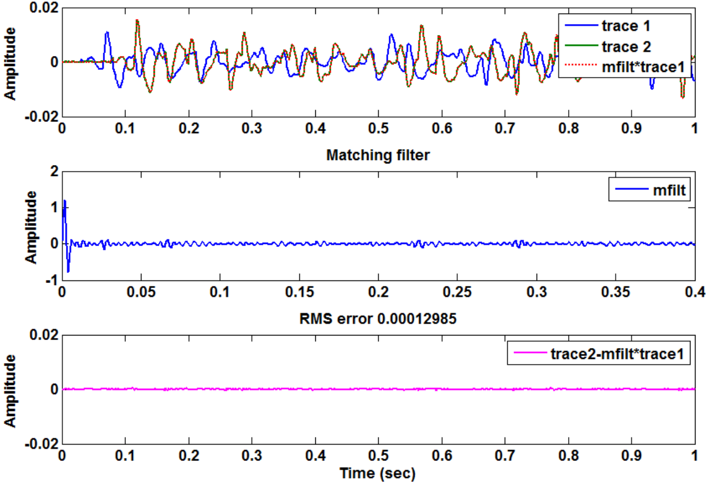
Matching filters



Matching filters



Matching filters



In time-lapse we normally have 2 data sets:

- Baseline survey; and
- Monitoring survey

Their surface-consistent model is:

$$
\begin{aligned}
d1_{ij}(t) &= s1_i(t) * r1_j(t) * h1_k(t) * y1_l(t) \\
d2_{ij}(t) &= s2_i(t) * r2_j(t) * h2_k(t) * y2_l(t)
\end{aligned}
\tag{3}
$$

Surface-consistent matching filters

Several steps to obtain a SCMF:

Surface-consistent matching filters

Several steps to obtain a SCMF:

- design $\mathbf{m}$ such that

$$\mathbf{d1} = \mathbf{D2m}, \tag{4}$$

  where $\mathbf{D2}$ is the convolution matrix formed from $\mathbf{d2}$

Surface-consistent matching filters

Several steps to obtain a SCMF:

- design $\mathbf{m}$ such that
$$\mathbf{d1} = \mathbf{D2m}, \tag{4}$$
where $\mathbf{D2}$ is the convolution matrix formed from $\mathbf{d2}$
- solve for $\mathbf{m}$
$$\mathbf{m} = (\mathbf{D2}^T\mathbf{D2} + \alpha^2\mathbf{I})^{-1}\mathbf{D2}^T\mathbf{d1} \tag{5}$$

Surface-consistent matching filters

Several steps to obtain a SCMF:

- design $\mathbf{m}$ such that
$$\mathbf{d1} = \mathbf{D2m}, \tag{4}$$
where $\mathbf{D2}$ is the convolution matrix formed from $\mathbf{d2}$
- solve for $\mathbf{m}$
$$\mathbf{m} = (\mathbf{D2}^T\mathbf{D2} + \alpha^2\mathbf{I})^{-1}\mathbf{D2}^T\mathbf{d1} \tag{5}$$
- FFT $\mathbf{m}$

Surface-consistent matching filters

Several steps to obtain a SCMF:

- design $\mathbf{m}$ such that
$$\mathbf{d1} = \mathbf{D2m}, \tag{4}$$
where $\mathbf{D2}$ is the convolution matrix formed from $\mathbf{d2}$
- solve for $\mathbf{m}$
$$\mathbf{m} = (\mathbf{D2}^T\mathbf{D2} + \alpha^2\mathbf{I})^{-1}\mathbf{D2}^T\mathbf{d1} \tag{5}$$
- FFT $\mathbf{m}$
- take the log of FFT($\mathbf{m}$)

Surface-consistent matching filters

Several steps to obtain a SCMF:

- design $\mathbf{m}$ such that

$$\mathbf{d1} = \mathbf{D2m}, \tag{4}$$

  where $\mathbf{D2}$ is the convolution matrix formed from $\mathbf{d2}$
- solve for $\mathbf{m}$

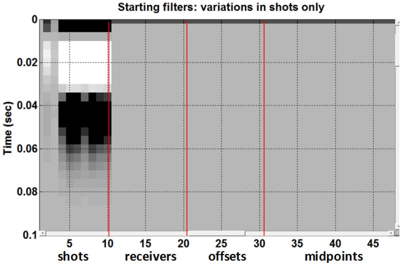
$$\mathbf{m} = (\mathbf{D2}^T\mathbf{D2} + \alpha^2\mathbf{I})^{-1}\mathbf{D2}^T\mathbf{d1} \tag{5}$$

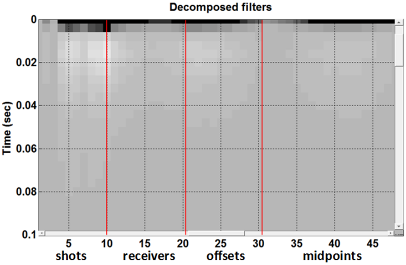
- FFT $\mathbf{m}$
- take the log of FFT($\mathbf{m}$)

The above are trace-by-trace process. Once that complete, we

- decompose the log(FFT($\mathbf{m}$)) into four-components

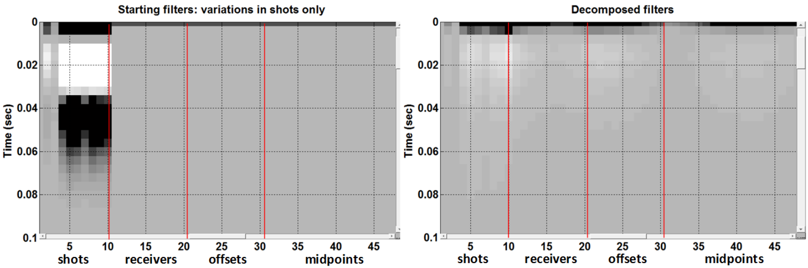
1D example: variations in source





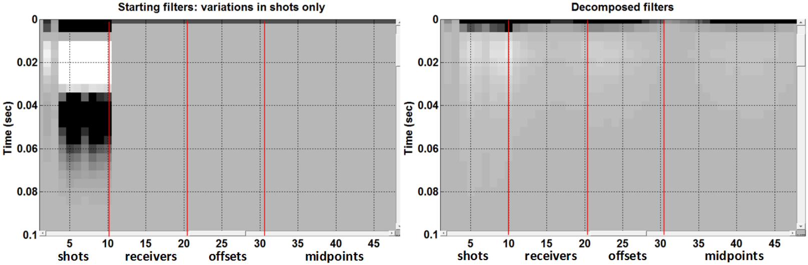
- variation in input shot filters only

1D example: variations in source



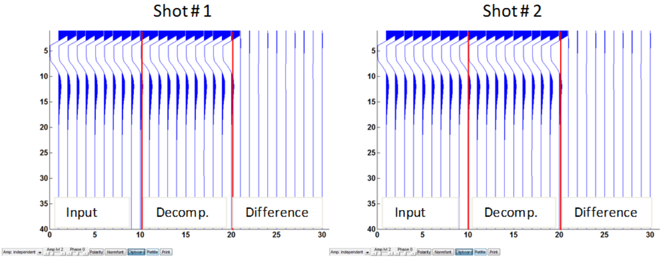
- variation in input shot filters only
- decomposed filters show variation in all components

1D example: variations in source

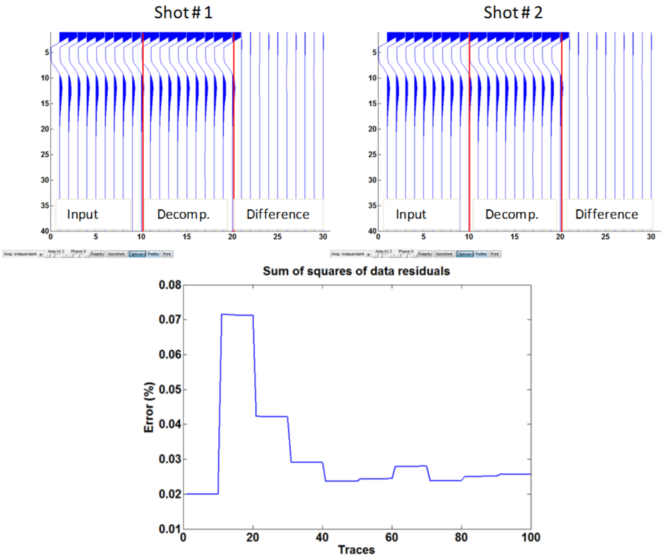


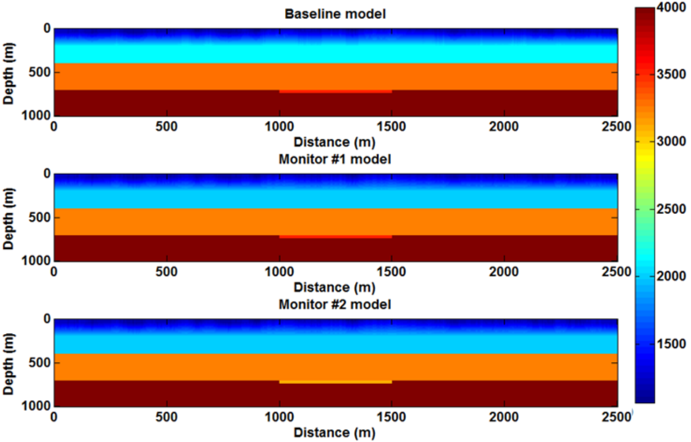
- variation in input shot filters only
- decomposed filters show variation in all components
- comparing the input data with the decomposed data we see the following:

1D example: variations in source

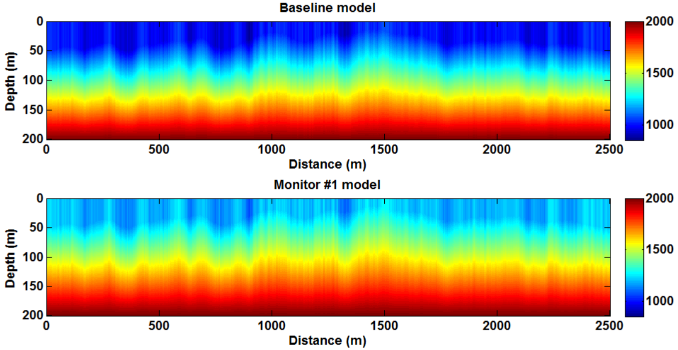


1D example: variations in source

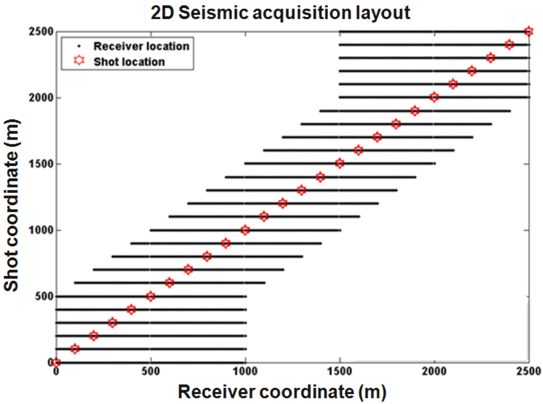



Baseline model
Monitor #1 model
Monitor #2 model
Depth (m)
Distance (m)
0
500
1000
1500
2000
2500
4000
3500
3000
2500
2000
1500

Zoom in:

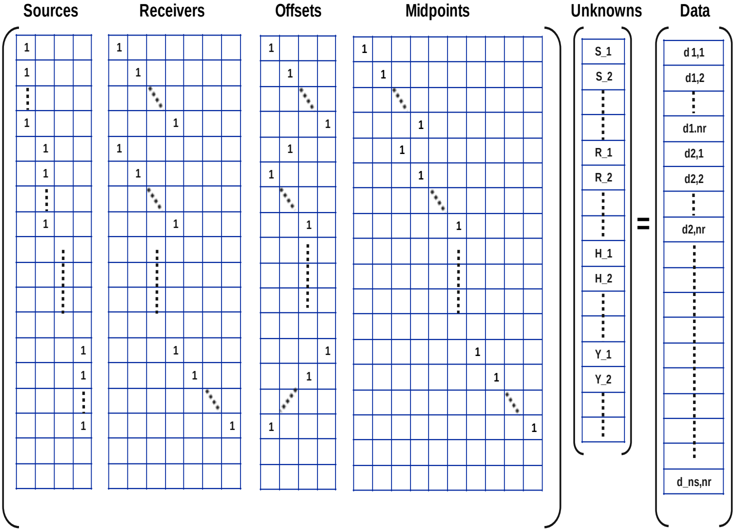

Baseline model
Depth (m)
0
50
100
150
200
0
500
1000
1500
2000
2500
Distance (m)
2000
1500
1000
Monitor #1 model
Depth (m)
0
50
100
150
200
0
500
1000
1500
2000
2500
Distance (m)
2000
1500
1000

Data acquisition



shot spacing = 100m
receiver spacing = 10m
total number of shots = 26
101 receivers per shot

How do we get the SCMF?



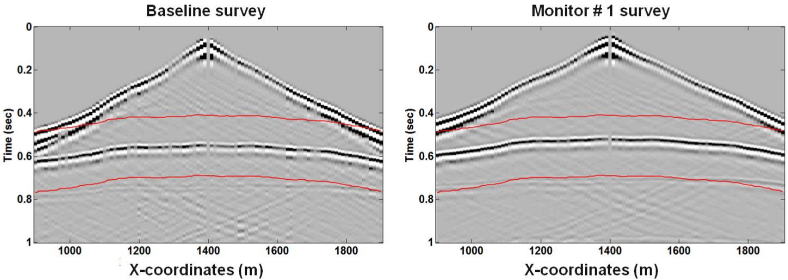
How do we construct the SCMF?

$$\mathbf{G}\mathbf{x} = \mathbf{p}, \tag{6}$$

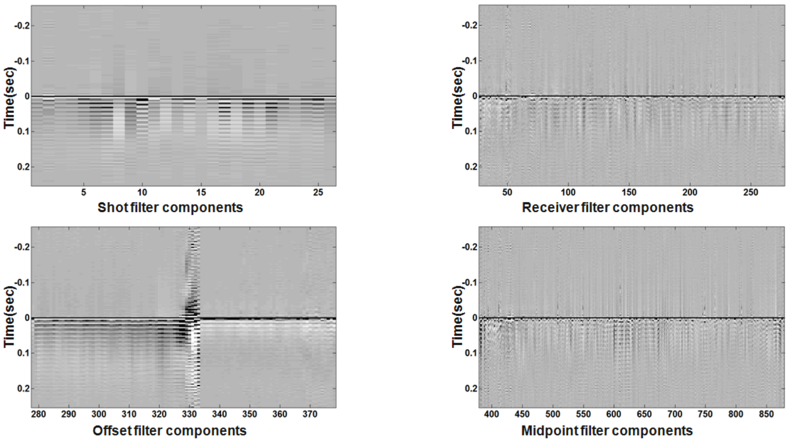
where $\mathbf{G}$ is the geometry matrix and $\mathbf{x}$ contains the unknown parameters. Recall that we have the $\log(\mathrm{FFT}(\mathbf{m}))$, we denote it $\mathbf{p}$, and the decomposition is:

$$\mathbf{x} = (\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{I})^{-1}\mathbf{G}^T\mathbf{p}, \tag{7}$$

Constructing the SCMF



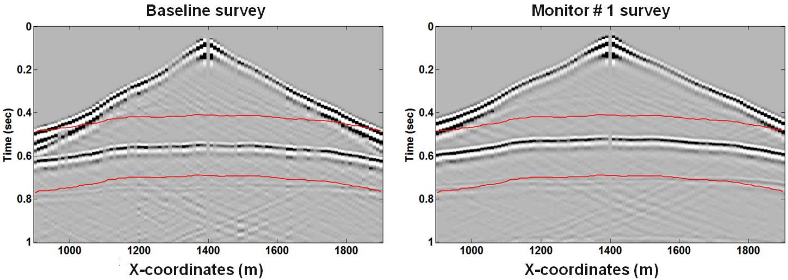
Constructing the SCMF



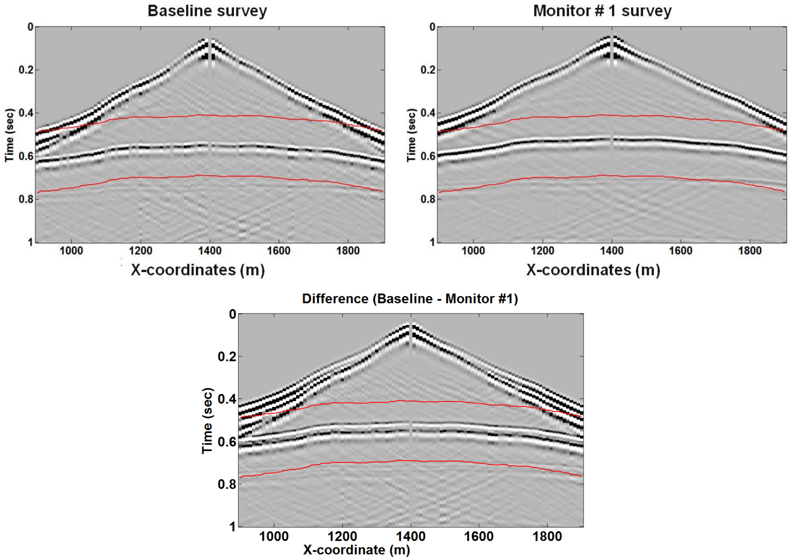
We will show two examples:

1. matching monitor # 1 survey to baseline survey
2. matching monitor # 2 survey to monitor # 1 survey

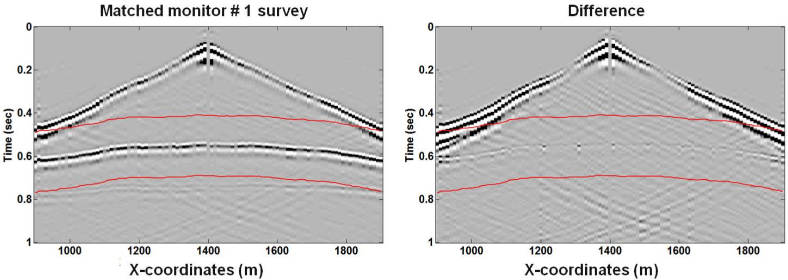
2D example 1


Baseline survey
Monitor # 1 survey
Time (sec)
0
0.2
0.4
0.6
0.8
1
1000
1200
1400
1600
1800
X-coordinates (m)

2D example 1


Baseline survey
Time (sec)
X-coordinates (m)
Monitor # 1 survey
Time (sec)
X-coordinates (m)
Difference (Baseline - Monitor #1)
Time (sec)
X-coordinate (m)

2D example 1

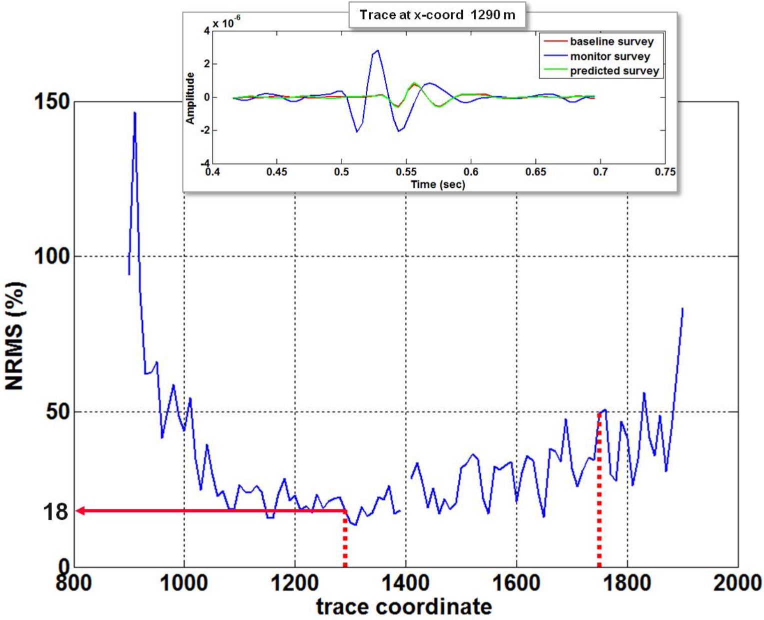

Matched monitor # 1 survey
Difference
Time (sec)
0
0.2
0.4
0.6
0.8
1
1000
1200
1400
1600
1800
X-coordinates (m)

What is NRMS?

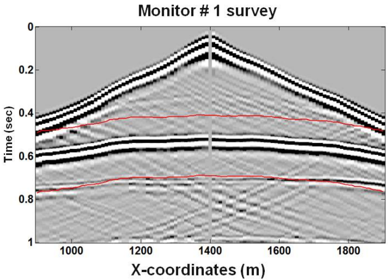
Nrms is considered as the metric system that measures the repeatability between two traces a_t and b_t in a time gate:

$$NRMS = 2\frac{RMS(a_t - b_t)}{RMS(a_t) + RMS(b_t)}. \tag{8}$$

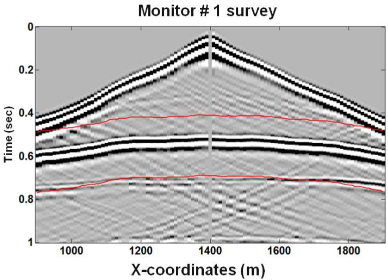
2D example 1: NRMS



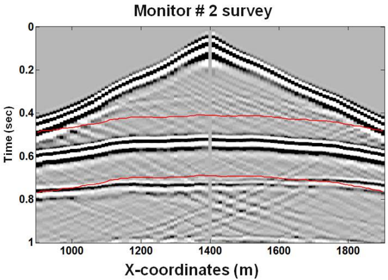
2D example 2


Monitor # 1 survey
Monitor # 2 survey
Time (sec)
0
0.2
0.4
0.6
0.8
1
1000
1200
1400
1600
1800
X-coordinates (m)

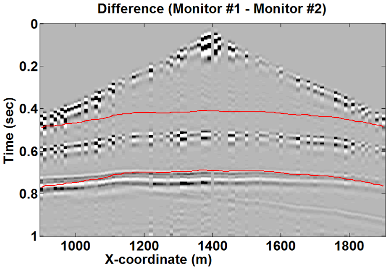
2D example 2



Monitor # 1 survey
0
0.2
0.4
0.6
0.8
1
Time (sec)
1000
1200
1400
1600
1800
X-coordinates (m)

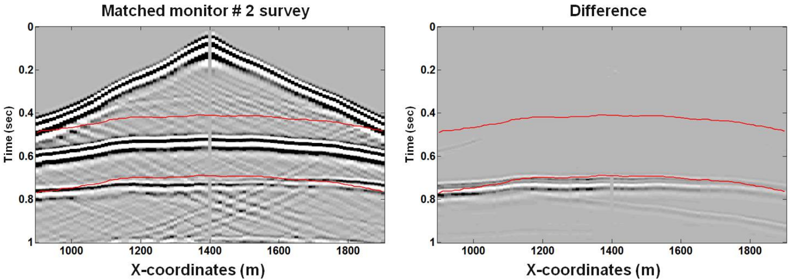


Monitor # 2 survey
0
0.2
0.4
0.6
0.8
1
Time (sec)
1000
1200
1400
1600
1800
X-coordinates (m)

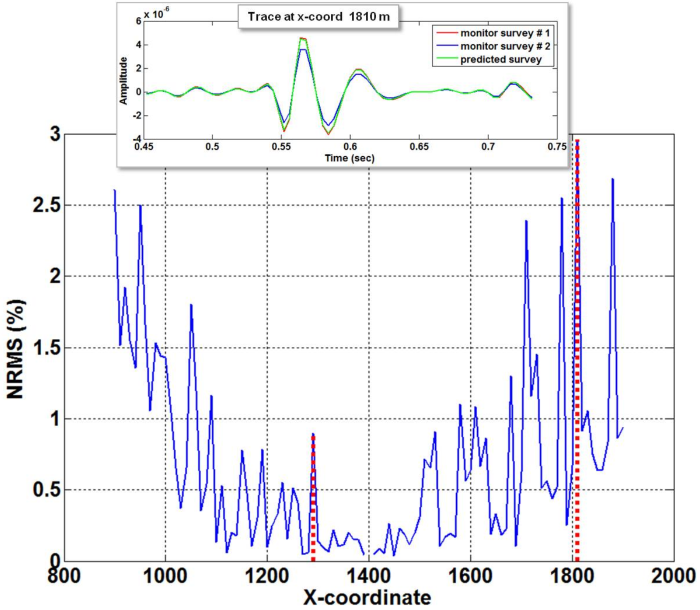


Difference (Monitor #1 - Monitor #2)
0
0.2
0.4
0.6
0.8
1
Time (sec)
1000
1200
1400
1600
1800
X-coordinate (m)

2D example 2


Matched monitor # 2 survey
Difference
Time (sec)
0
0.2
0.4
0.6
0.8
1
1000
1200
1400
1600
1800
X-coordinates (m)

2D example 2: NRMS



Conclusions

- We've deomonstrated the theory of surface-consistent matching filters
- It's analogous to surface-consistent deconvolution **except** we consider **two** data set
- Synthetic time-lapse data sets: baseline, first monitor survey, second monitor survey
- Constructed the 4-component surface-consistent matching filters
- Applied the matching filters to two time-lapse examples
- Balanced amplitudes, equalized phase and bandwidth and compensated for time-shifts

The authors thank the following:

- The sponsors CREWES.
- CREWES staff and students, in particular Faranak Mahmoudian, Hassan Khaniani, Marcus Wilson, A.Naser Yousef Zadeh, Rolf Maier, Kevin Hall, and David Henley.
- Saudi Aramco for sponsoring Mahdi Almutlaq's study program.

Thank you ...
Questions?