Curvelet-Based Primary-Multiple Separation from a Bayesian Perspective

Rayan Saab

Department of Electrical and Computer Engineering, UBC
in collaboration with
Seismic Laboratory for Imaging and Modelling (SLIM), UBC

September 26, 2007

Co-authors

- Deli Wang (Jilin University)
- Felix Herrmann (Earth and Ocean Sciences, Univesity of British Columbia)
- Ozgur Yilmaz (Mathematics Department, UBC)

Introduction

- We introduce a new primary-multiple separation scheme that
  1. Utilizes the sparsity of primaries and multiples in the curvelet domain and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- The algorithm can be derived from a Bayesian formulation that assumes
  1. A sparsity enforcing Laplacian prior distribution
  2. an assumption of Gaussian noise and errors
- The algorithm uses soft-thresholding operations, no matrix inversions, makes great progress and almost converges in only a few iterations (for this type of problems)

Introduction

- We introduce a new primary-multiple separation scheme that
  1. Utilizes the sparsity of primaries and multiples in the curvelet domain and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- The algorithm can be derived from a Bayesian formulation that assumes
  1. A sparsity enforcing Laplacian prior distribution
  2. an assumption of Gaussian noise and errors
- The algorithm uses soft-thresholding operations, no matrix inversions, makes great progress and almost converges in only a few iterations (for this type of problems)

Introduction

- We introduce a new primary-multiple separation scheme that
  1. Utilizes the sparsity of primaries and multiples in the curvelet domain and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- The algorithm can be derived from a Bayesian formulation that assumes
  1. A sparsity enforcing Laplacian prior distribution
  2. an assumption of Gaussian noise and errors
- The algorithm uses soft-thresholding operations, no matrix inversions, makes great progress and almost converges in only a few iterations (for this type of problems)

Problem and Scope

- Suppose that we have
  1. Seismic data:
  $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
  composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
  $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
  which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Problem and Scope

- Suppose that we have
  1. Seismic data:
     $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
     composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
     $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
     which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Problem and Scope

- Suppose that we have
  1. Seismic data:
  $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
  composed of the true primaries ($\mathbf{s}_1$), multiples ($\mathbf{s}_2$), noise ($\mathbf{n}$)
  2. Predictions of the multiples (e.g. from SRME or other methods):
  $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
  which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Problem and Scope

- Suppose that we have
  1. Seismic data:
     $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
     composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
     $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
     which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Sparsity

What is Sparsity ?

- A signal is
  said to be “sparse” if most of its values are zero, or almost zero.

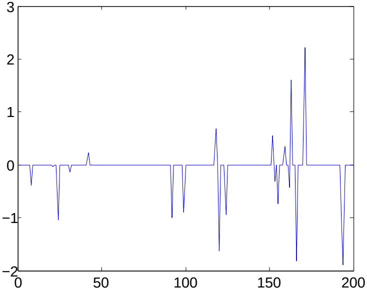


Figure: An Example of a Sparse Signal

Sparsity

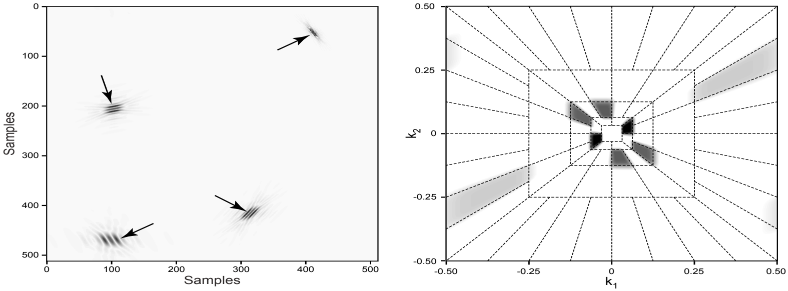
What is Sparsity ?

- A signal is
  said to be "sparse" if most of its values are zero, or almost zero.
- If a signal $\mathbf{s}$ is not sparse, sometimes we can find a representation $\mathbf{s} = \mathbf{A}\mathbf{x}$ where $\mathbf{x}$ is sparse.
- Primaries and multiples are sparse in the curvelet domain.
- In other words, a seismic signal can be represented as $\mathbf{s} = \mathbf{A}\mathbf{x}$ where
  - $\mathbf{A} = \mathbf{C}^{\mathbf{H}}$ is the synthesis curvelet operator and
  - $\mathbf{x}$ is the vector of curvelet coefficients.

Curvelets

- Curvelets are localized 'little plane-waves' that are oscillatory in one direction and smooth in the other direction(s).
- They are multiscale and multi-directional.
- Curvelets have an anisotropic shape – they obey the so-called parabolic scaling relationship, yielding a width $\propto$ length^2 for the support of curvelets.
- Very good for detecting wavefronts

Curvelets



(a) (b)

Figure: Curvelet examples. **(a)-(b)** spatial and frequency representation of four different curvelets in the spatial domain at three different scales and in the Fourier domain

Seismic Primary Multiple Separation

- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Seismic Primary Multiple Separation

- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Seismic Primary Multiple Separation

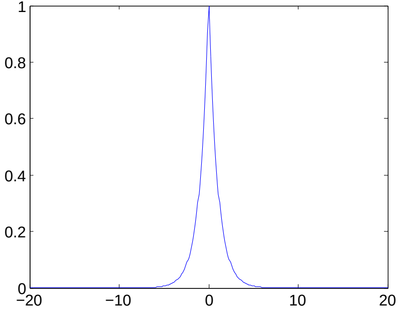
- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Seismic Primary Multiple Separation

- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Sparsity Enforcing Bayesian Prior

- We know that primaries and multiples are sparse in curvelets, and we want to use that knowledge.
- A good sparsity enforcing prior distribution is the Laplacian (Cauchy) distribution $p(x) = ce^{-a|x|}$



Sparsity Enforcing Bayesian Prior

- We know that primaries and multiples are sparse in curvelets, and we want to use that knowledge.
- A good sparsity enforcing prior distribution is the Laplacian (Cauchy) distribution $p(x) = ce^{-a|x|}$
- We also have predictions of the multiples (and primaries), so we use a weighted laplacian prior instead.
- $p(\mathbf{x}_1) = ce^{-\mathbf{w}_1|\mathbf{x}_1|}$ with $\mathbf{w}_1 = \lambda_1 \mathbf{A^H b_2}$
- $p(\mathbf{x}_2) = ce^{-\mathbf{w}_2|\mathbf{x}_2|}$ with $\mathbf{w}_2 = \lambda_2 \mathbf{A^H b_1}$
- In other words we make it unlikely that the curvelet coefficients of the primaries are high where there are high coefficients for the multiples and vice versa

MAP estimator

- We want to find the curvelet coefficients of the primaries and multiples ($\mathbf{x}_1$ and $\mathbf{x}_2$) knowing that
- $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- Maximize $P(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{b}_1, \mathbf{b}_2)$
- This leads to the following formulation

$$
\begin{aligned}
\arg\max_{\mathbf{x_1},\mathbf{x_2}} P(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{b}_1, \mathbf{b}_2) &= \arg\max_{\mathbf{x_1},\mathbf{x_2}} P(\mathbf{x}_1, \mathbf{x}_2) P(\mathbf{n}) P(\mathbf{n}_2) \\
&= \arg\max_{\mathbf{x_1},\mathbf{x_2}} e^{-\alpha_1 \|\mathbf{x}_1\|_{1,\mathbf{w}_1}} e^{-\alpha_2 \|\mathbf{x}_2\|_{1,\mathbf{w}_2}} e^{-\frac{\|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2}{\sigma_1^2}} e^{-\frac{\|\mathbf{A}(\mathbf{x}_1+\mathbf{x}_2)-(\mathbf{b}_1+\mathbf{b}_2)\|_2^2}{\sigma^2}} \\
&= \arg\max_{\mathbf{x_1},\mathbf{x_2}} - \Big(\alpha_1 \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \alpha_2 \|\mathbf{x}_2\|_{1,\mathbf{w}_1} + \frac{\|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2}{\sigma_2^2} \\
&\qquad + \frac{\|\mathbf{A}(\mathbf{x}_1+\mathbf{x}_2)-(\mathbf{b}_1+\mathbf{b}_2)\|_2^2}{\sigma^2} \Big) \\
&= \arg\min_{\mathbf{x_1},\mathbf{x_2}} f(\mathbf{x}_1, \mathbf{x}_2)
\end{aligned}
$$

- Here $\|\mathbf{x}_i\|_{1,\mathbf{w}_i} = \sum_\mu |w_{i,\mu} x_{i,\mu}|$, $\mu \in \mathcal{M}$

Objective Function

$$f(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \|\mathbf{x}_2\|_{1,\mathbf{w}_2} + \|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2 + \eta\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2$$

Objective Function

$$f(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \|\mathbf{x}_2\|_{1,\mathbf{w}_2} + \|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2 + \eta\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2$$

Bayesian Interpretation

Minimizing $f(\mathbf{x}_1, \mathbf{x}_2)$ is equivalent to finding the MAP estimator assuming that the coefficients of the sources follow independent weighted Laplacian prior and noise (error) is Gaussian.

Objective Function

$$f(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \|\mathbf{x}_2\|_{1,\mathbf{w}_2} + \|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2 + \eta\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2$$

Bayesian Interpretation

Minimizing $f(\mathbf{x}_1, \mathbf{x}_2)$ is equivalent to finding the MAP estimator assuming that the coefficients of the sources follow independent weighted Laplacian prior and noise (error) is Gaussian.

Separation Algorithm

$$\begin{aligned}
\mathbf{x_1^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_1}}{\mathbf{2}\eta}}\left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n} + \mathbf{x_1^n}\right] \\
\mathbf{x_2^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_2}}{\mathbf{2(1+\eta)}}}\left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{x_2^n} + \frac{\eta}{\eta+1}\left(\mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n}\right)\right]
\end{aligned}$$

Iterative Thresholding

- Thus our algorithm can be described as

$$\begin{aligned}
\mathbf{x_1^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_1}}{\mathbf{2}\eta}}\left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n} + \mathbf{x_1^n}\right] \\
\mathbf{x_2^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_2}}{\mathbf{2(1+\eta)}}}\left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{x_2^n} + \frac{\eta}{\eta+1}\left(\mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n}\right)\right]
\end{aligned}$$

- Here $\mathbf{S}_\alpha$ is the soft-thresholding operator acting *elementwise* as

$$S_{\alpha_\mu}(v_\mu) = \mathrm{sgn}(v_\mu) \cdot \max(0, |v_\mu| - |\alpha_\mu|).$$

- The algorithm provably converges to the minimizer of the objective function (Similar to work of Daubechies04).

Description of Parameters

- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

Description of Parameters

- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

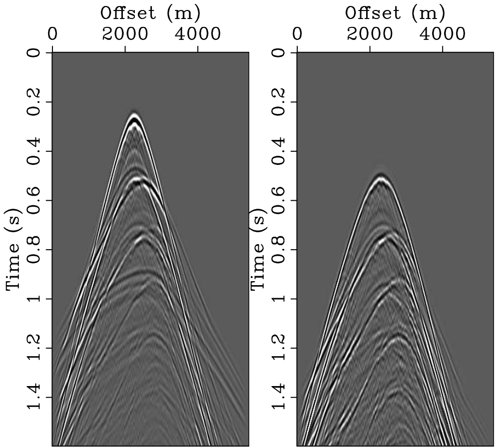
Description of Parameters

- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

Description of Parameters

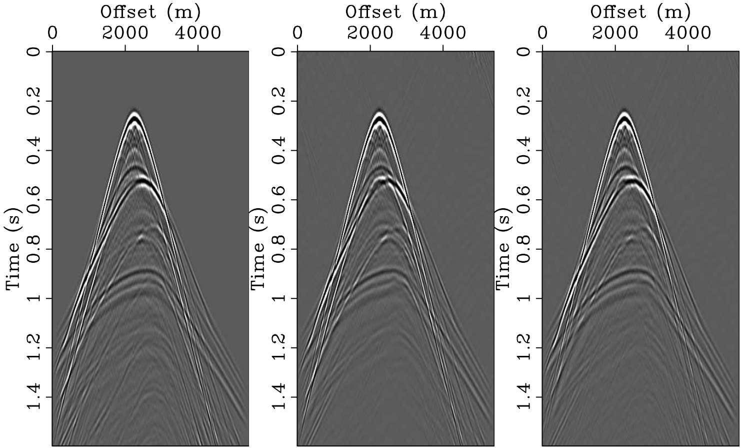
- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

Total Data and Predicted Multiples



Separation Results

- SRME Single threshold (Herrmann07) proposed algorithm



Conclusion

- We introduced a primary-multiple separation algorithm that
  1. Utilizes the sparsity of primaries and multiples in the curvelet domain and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- can be derived from a Bayesian formulation that assumes
  1. A sparsity enforcing Laplacian prior distribution
  2. an assumption of Gaussian noise and errors
- The algorithm uses soft-thresholding operations, no matrix inversions, converges in only a few iterations (for this type of problems)

Conclusion

- We introduced a primary-multiple separation algorithm that
  1. Utilizes the sparsity of primaries and multiples in the curvelet domain and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- can be derived from a Bayesian formulation that assumes
  1. A sparsity enforcing Laplacian prior distribution
  2. an assumption of Gaussian noise and errors
- The algorithm uses soft-thresholding operations, no matrix inversions, converges in only a few iterations (for this type of problems)

Conclusion

- We introduced a primary-multiple separation algorithm that
  1. Utilizes the sparsity of primaries and multiples in the curvelet domain and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- can be derived from a Bayesian formulation that assumes
  1. A sparsity enforcing Laplacian prior distribution
  2. an assumption of Gaussian noise and errors
- The algorithm uses soft-thresholding operations, no matrix inversions, converges in only a few iterations (for this type of problems)

The authors would like to thank

- The authors of CurveLab (Demanet, Ying, Candes, Donoho)
- William Symes for the reverse-time migration code.
- These results were created with Madagascar developed by Sergey Fomel.
- This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE (334810-05) of F.J.H. This research was carried out as part of the SINBAD project with support, secured through ITF (the Industry Technology Facilitator), from the following organizations: BG Group, BP, Chevron,ExxonMobil and Shell.