Sparsity promoting least-squares migration for long offset sparse OBN

Mathias Louboutin, Ziyi Yin, Yijun Zhang, and Felix J. Herrmann

SLIM

Georgia Institute of Technology

Sparsity promoting least-squares migration for long offset sparse OBN

Mathias Louboutin, Ziyi Yin, Yijun Zhang, and Felix J. Herrmann

SLIM
Georgia Institute of Technology

Context

“Wave-equation based imager’s dream…”

- create high-resolution high-fidelity images w/ no data processing
- increase acquisition productivity = work w/ sparse OBN simultaneous data
- improve computational performance = work w/ randomized subsets of shots

Recent trends: “Inversion is the way to go…”

- high-frequency FWI = hybrid of FWI & nonlinear “LS RTM”
- data- or image-space LS-RTM
- FWI & LS-RTM “equivalent” if background model kinematically correct
- locally convex GN converges faster

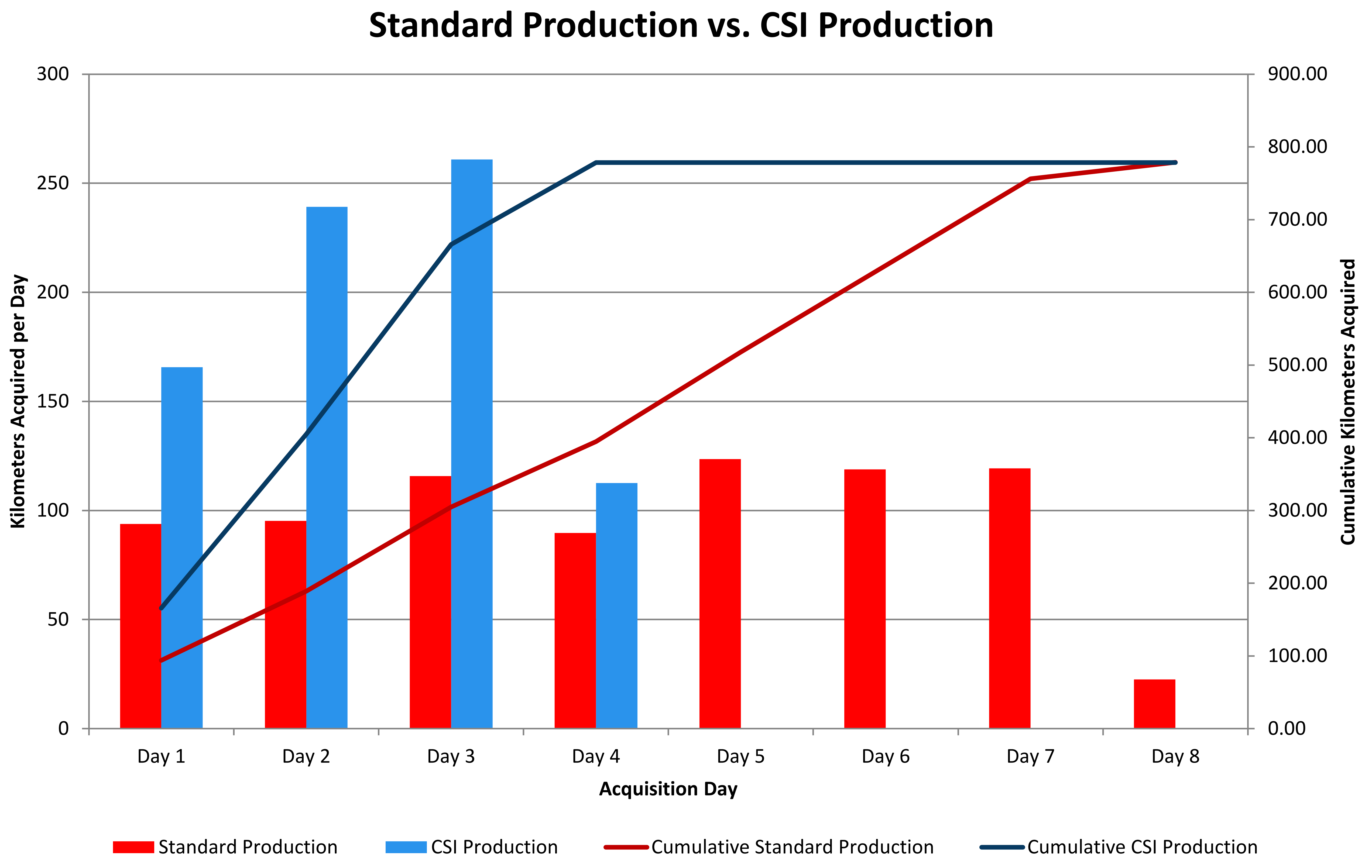
Established – acquisition savings

Compressive Sensing:

- exploits randomness & structure
- economic subsampled data
- recovers dense data via structure-promoting inversion

Output:

- improved quality artifact-fact free long-offset wide azimuth data
- **5 X – 10 X** productivity
- **saved $100's of millions**



adapted from Mosher et.al.

“Established” – computational savings

SP-LSRTM imaging framework:

- Devito-a just-in-time compiler for finite-differences
  (**at least 20% savings** ⟷ industry implementations TTI)
- Compressive least-squares migration w/ correct stable adjoints
  (**2–3X savings** w/ source subsampling ⟷ LS-RTM)
- Alternative checkpointing strategy
  (**2–3X savings** w/ on-the-fly Fourier transforms ⟷ optimal checkpointing)
- Serverless implementation in the Cloud (AWS/Azure)
  (**2–3X savings** w/ idle time reduction & spot pricing)

Anticipated total cost savings of 10–20X ⟷ LS-RTM @ 6X RTM

JUDI – The Julia Devito Inversion framework

Julia

- linear opeartors, data containers, IO
- ↓ parallel modeling function
- parallelization: distribute sources, data
- ↓ serial modeling function
- interface to Devito (Python)
- ↓

```
g = J'*(Pr*A_inv*Ps'*q - d_obs)
```

Python

- Devito: symbolic definition of PDE
- ↓ automatic code generation and JIT compilation

```
# Symbolic wavefield
u = TimeFunction(name='u', space_order=16)

# Acoustic wave equation
pde = model.m * u.dt2 - u.laplace
```

C

- solve PDE w/ OpenMP parallelism

```
float (*rec_coords)[2] = (Float (*[2]) rec_coords_vec;
{
  struct timeval start_kernel, endkernel;
    gettimeofday(&start_kernel, NULL);
    int t0;
    int t1;
    int t2;
    ;
    for (int i3 = 0; i3 < 3; i3+=1)
    {
      flops->kernel += 2.000000;
      {
        {
          t0 = (i3)%(3);
```

Example 1: Compressive seismic imaging

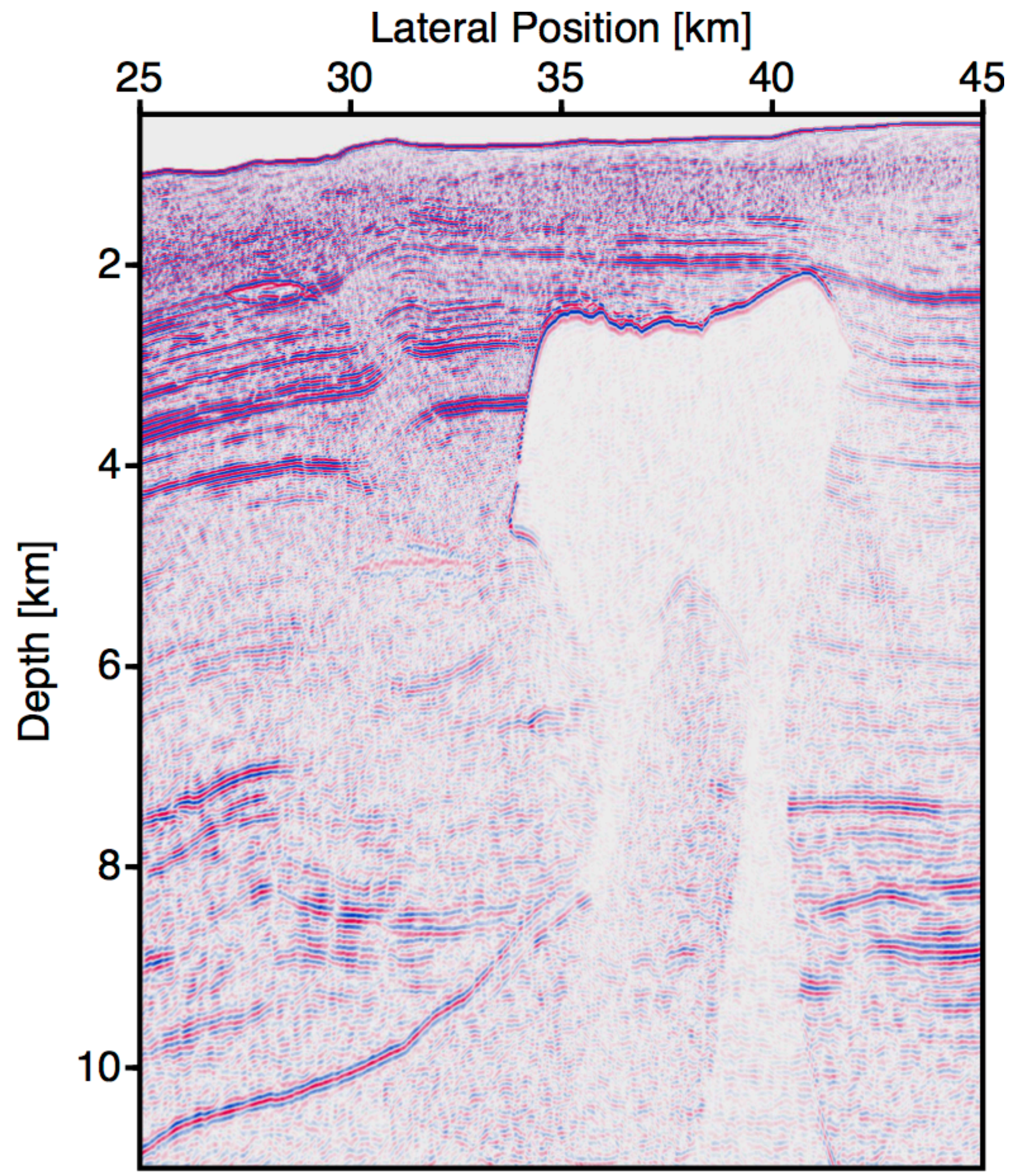
Linearized Bregman method with JUDI:

```
for j=1:maxiter

    # Compute residual and gradient
    i = randperm(d_obs.nsrc)[1:batchsize_source]
    select_frequencies!(J, batchsize_freq)
    r = Ml*J[i]*Mr*x - Ml*d_obs[i]
    g = Mr'*J[i]'*Ml'*proj_l2(r)

    # Residual and function value
    res[j] = norm(r, 2)
    fval[j] = λ*norm(C*z, 1) + .5f0*norm(C*z, 2)^2

    # Update variables
    global z -= α*g
    global x = C*soft_thresholding(C*z, λ)
end
```



Example 1: Compressive seismic imaging

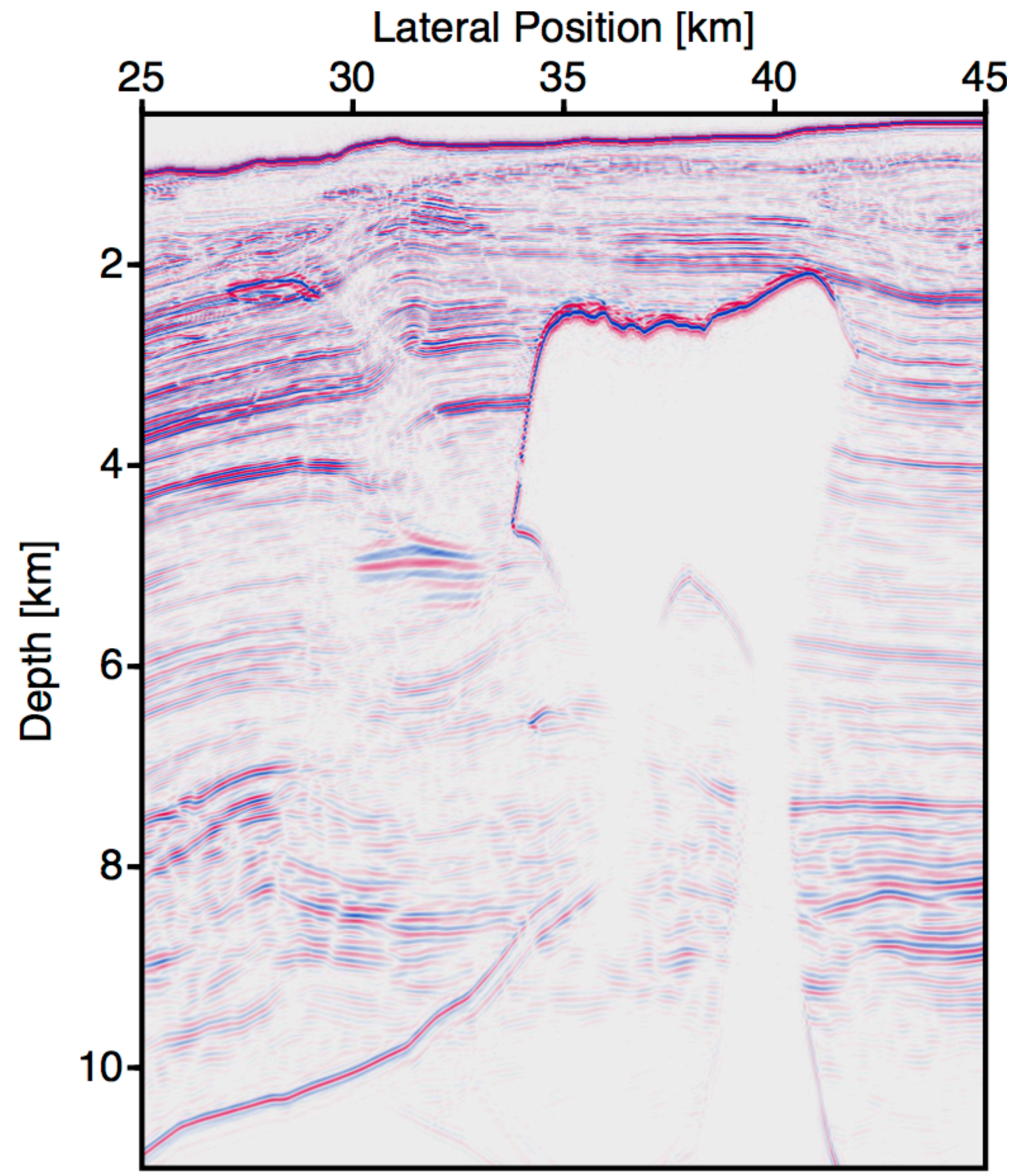
Linearized Bregman method with JUDI:

```
for j=1:maxiter

    # Compute residual and gradient
    i = randperm(d_obs.nsrc)[1:batchsize_source]
    select_frequencies!(J, batchsize_freq)
    r = Ml*J[i]*Mr*x - Ml*d_obs[i]
    g = Mr'*J[i]'*Ml'*proj_l2(r)

    # Residual and function value
    res[j] = norm(r, 2)
    fval[j] = λ*norm(C*z, 1) + .5f0*norm(C*z, 2)^2

    # Update variables
    global z -= α*g
    global x = C*soft_thresholding(C*z, λ)
end
```



Example server less in the Cloud

Sparsity-promoting SP-LSRTM on the BP Synthetic 2004 model

a)

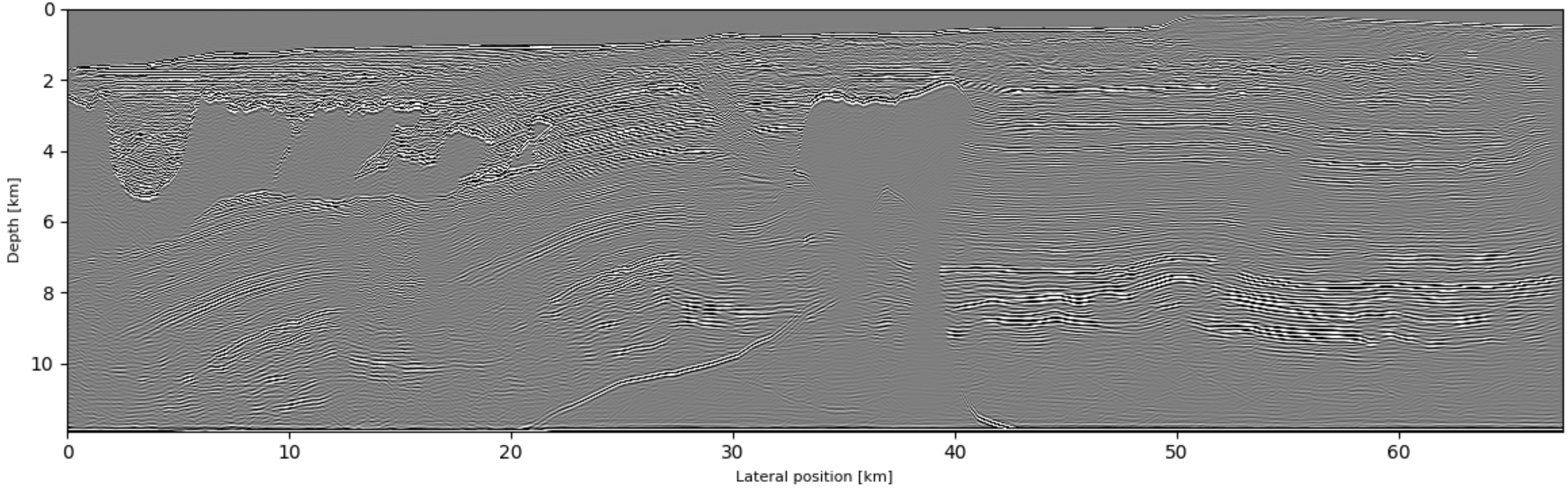
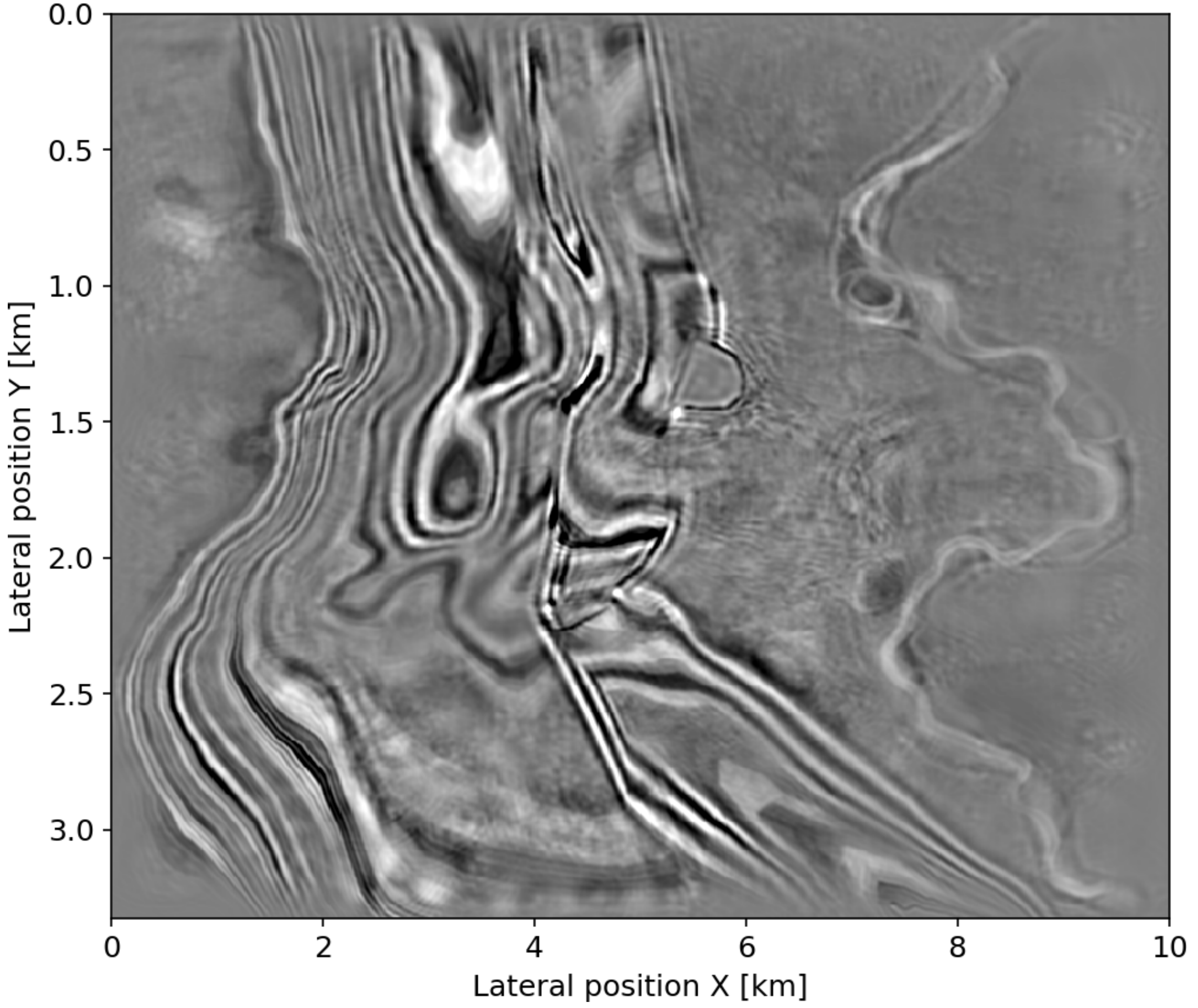


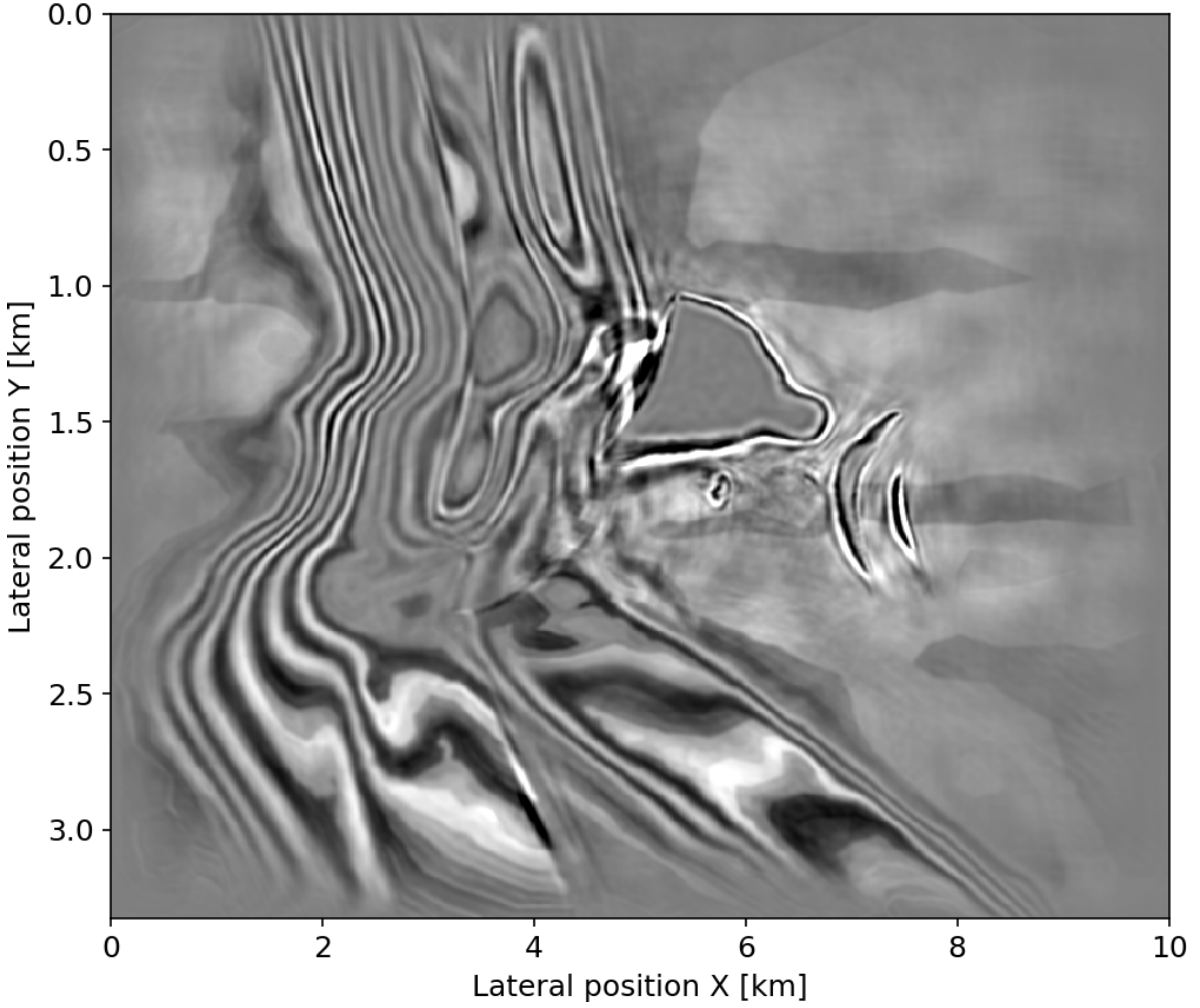
Image after ≈3 data passes (total cost of < 120 $)

3D TTI RTM on Azure

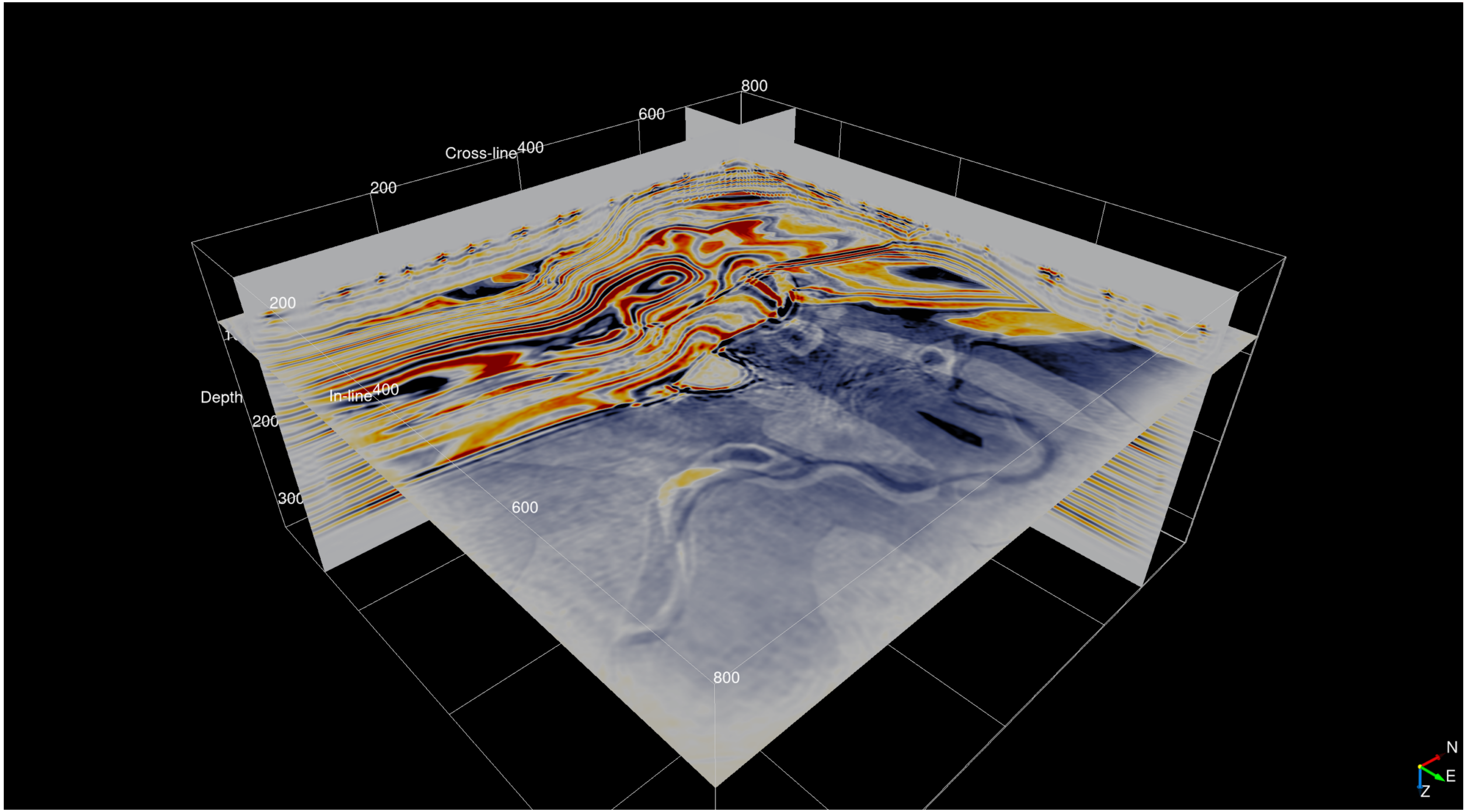
Depth slice 725 m



Depth slice 1250 m



3D TTI RTM on Azure for $10k



Challenges – sparse & ultra long offset

Make SP-LSRTM feasible via randomization

- randomized imaging w/ CS – randomized source sampling & checkpointing
- randomized acquisition w/ CS – sparse multi sim. source acquisition

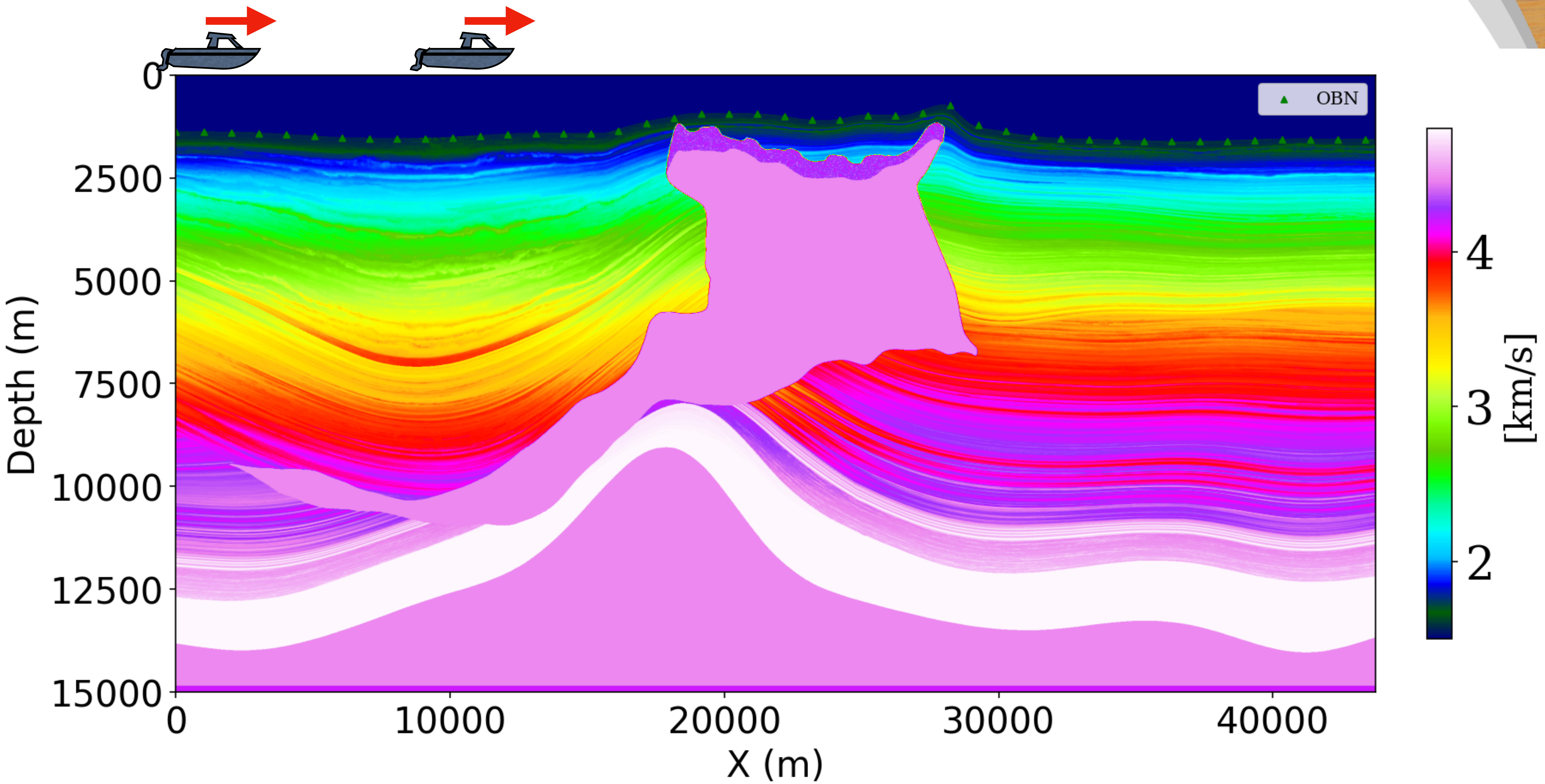
Deal w/ free surface not via

- LS-RTM w/ EPSI – adding deghosted upgoing wavefield as areal source
- **but instead** via SP-LSRTM w/ multiples by adding free-surface BC

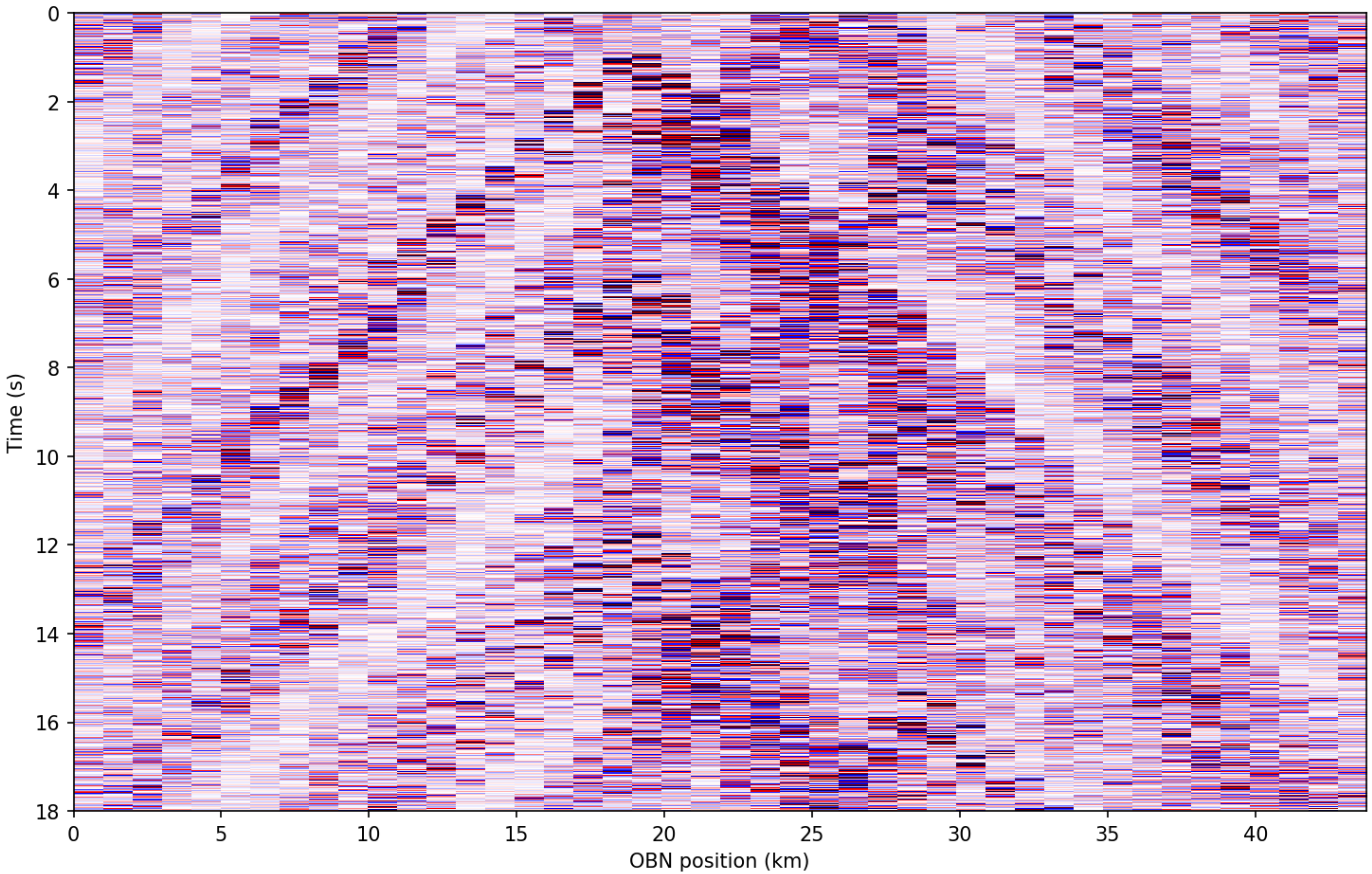
One in all solution – SP-LSRTM w/ blended data & w/ free surface

- no need to deblend
- no need to “demultiple” & deghost
- **cheap (10-20 X reduction imaging costs & 10X reduction acquisition costs)**

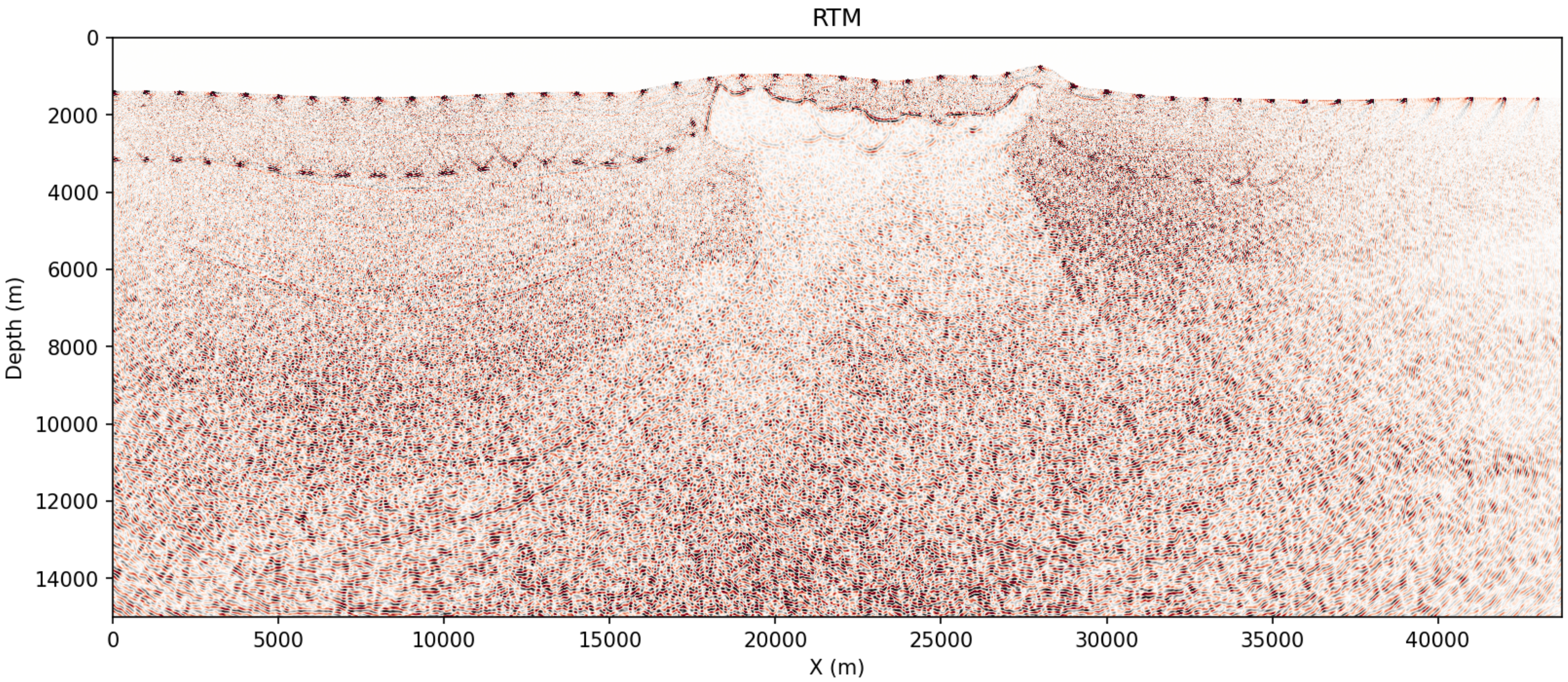
Stretched SEAM – max offset 43.75 km ($\Delta x \rightarrow 12.5\mathrm{m}$)



Challenge: turn blended sparse OBN data /w multiples



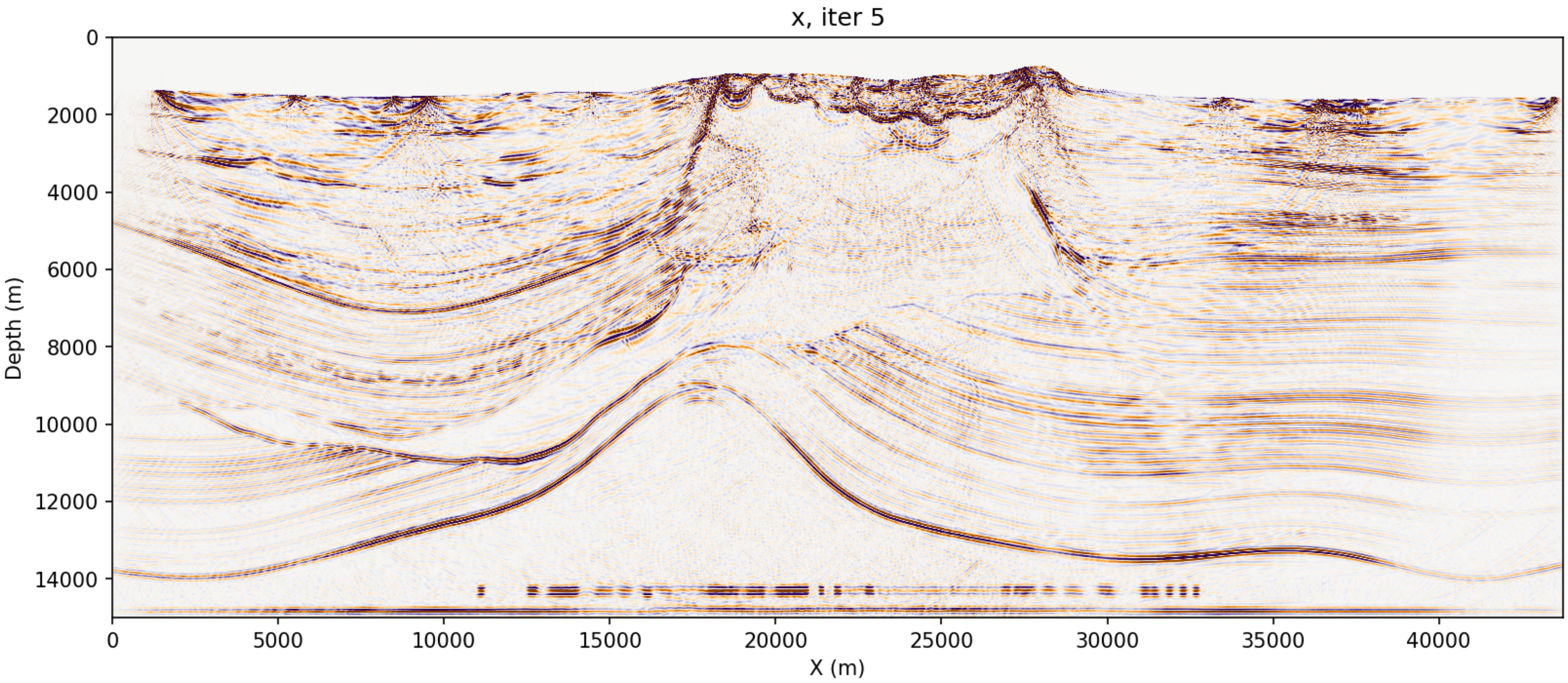
from this “mess”



into this…"ideally"

w/ 5 iterations w/ 4 shots sampled at ($\Delta x_s = 1\mathrm{km}$) w/ replacement

cost of ≈ 1.5 RTM



Challenges

– deal w/ free surface
– deal w/ blended data
– all in one go imaging…

A tale of multiples…

RTM based single scattering assumption:

- estimate inverse wavelet, remove surface-related multiples & ghost (SRME)
- **pro:** industry standard; **con:** expensive (app. cost 1 extra RTM)

Inversion w/ Linearized Born + EPSI = LS-RTM w/ upgoing wavefield

- estimate wavelet & deghost, invert primaries & multiples
- **pro:** images multiples; **con:** expensive & complex, not industry standard

Inversion w/ linearized Born + free surface BC

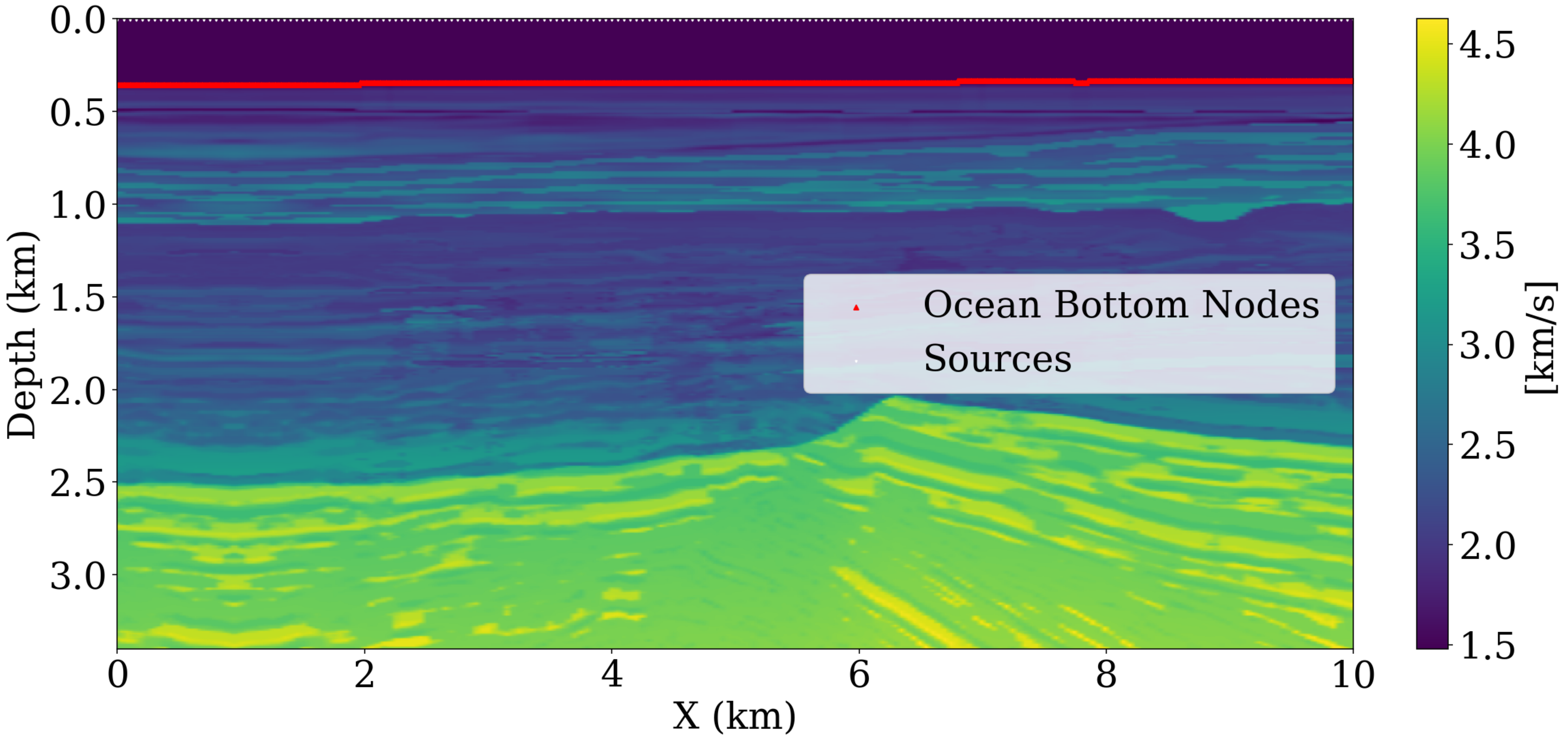
- include free surface BC in linearized Born operator
- **pro:** simple & easy combined w/ FWI; **con:** correct ocean bottom

Imaging w/ multiples

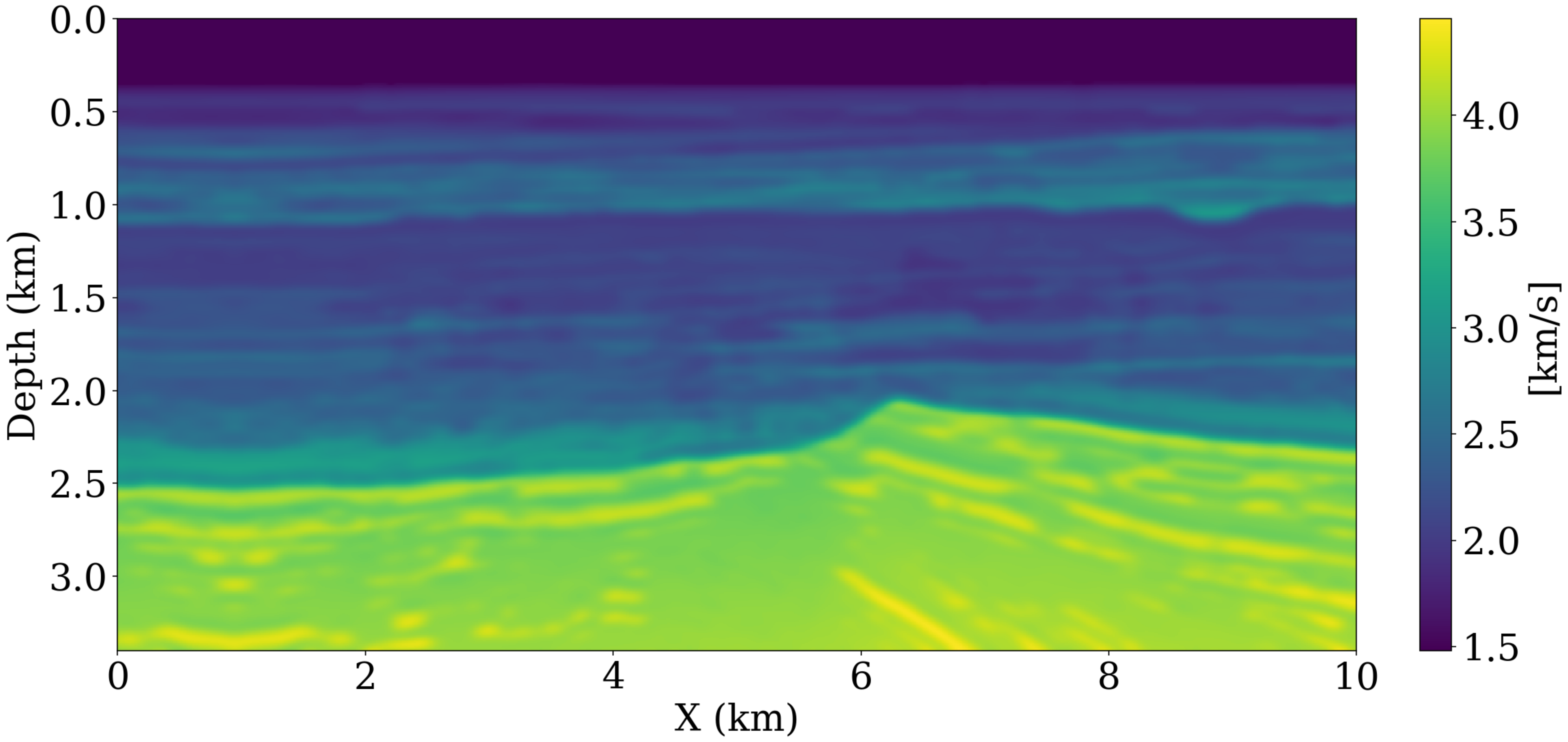
BG Compass model:

- 1001 OBNs (10m spacing)
- 201 sources (50m spacing, 6m depth)
- 15Hz Ricker wavelet
- 3.5 seconds recording
- inverse-scattering imaging condition
- 4 data passes (6 X cost single RTM)

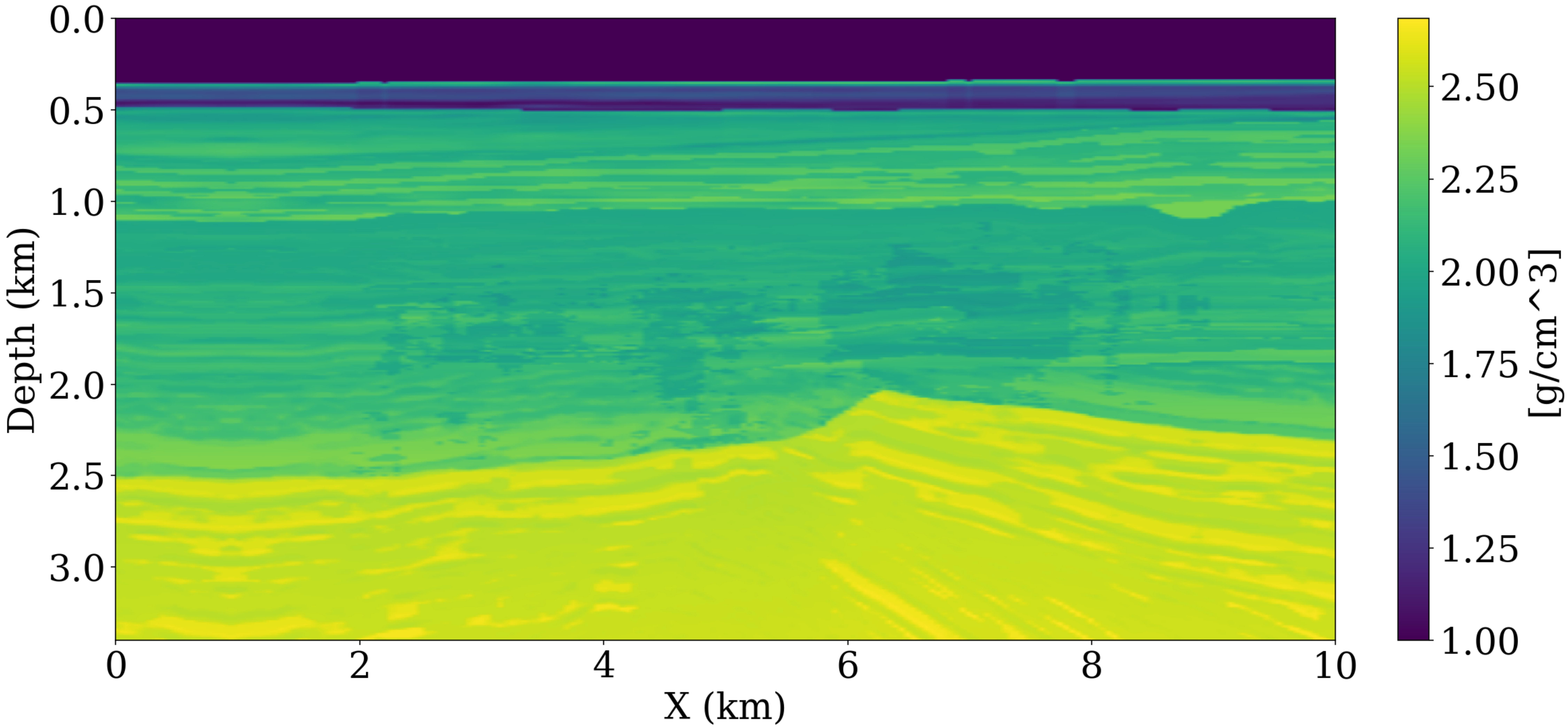
True Velocity



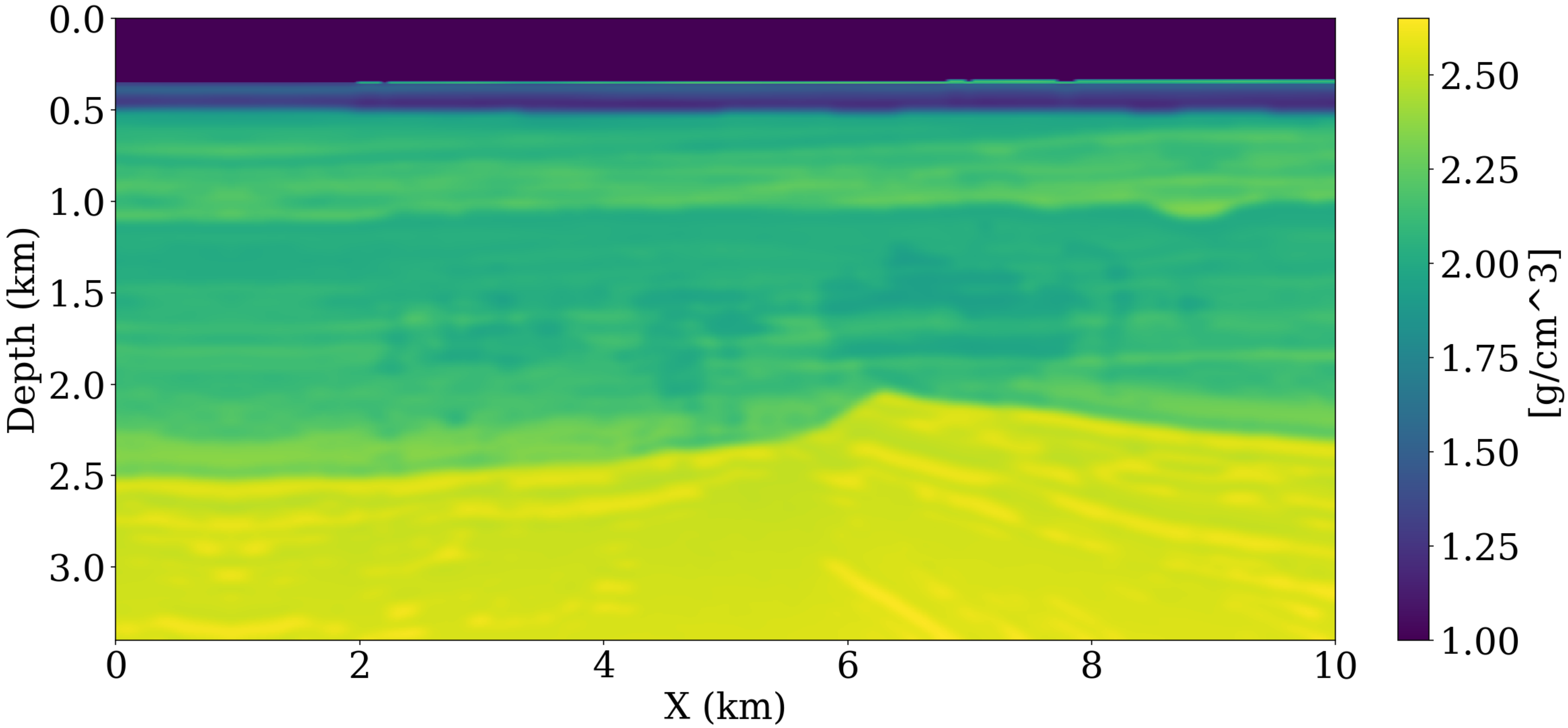
Background Velocity



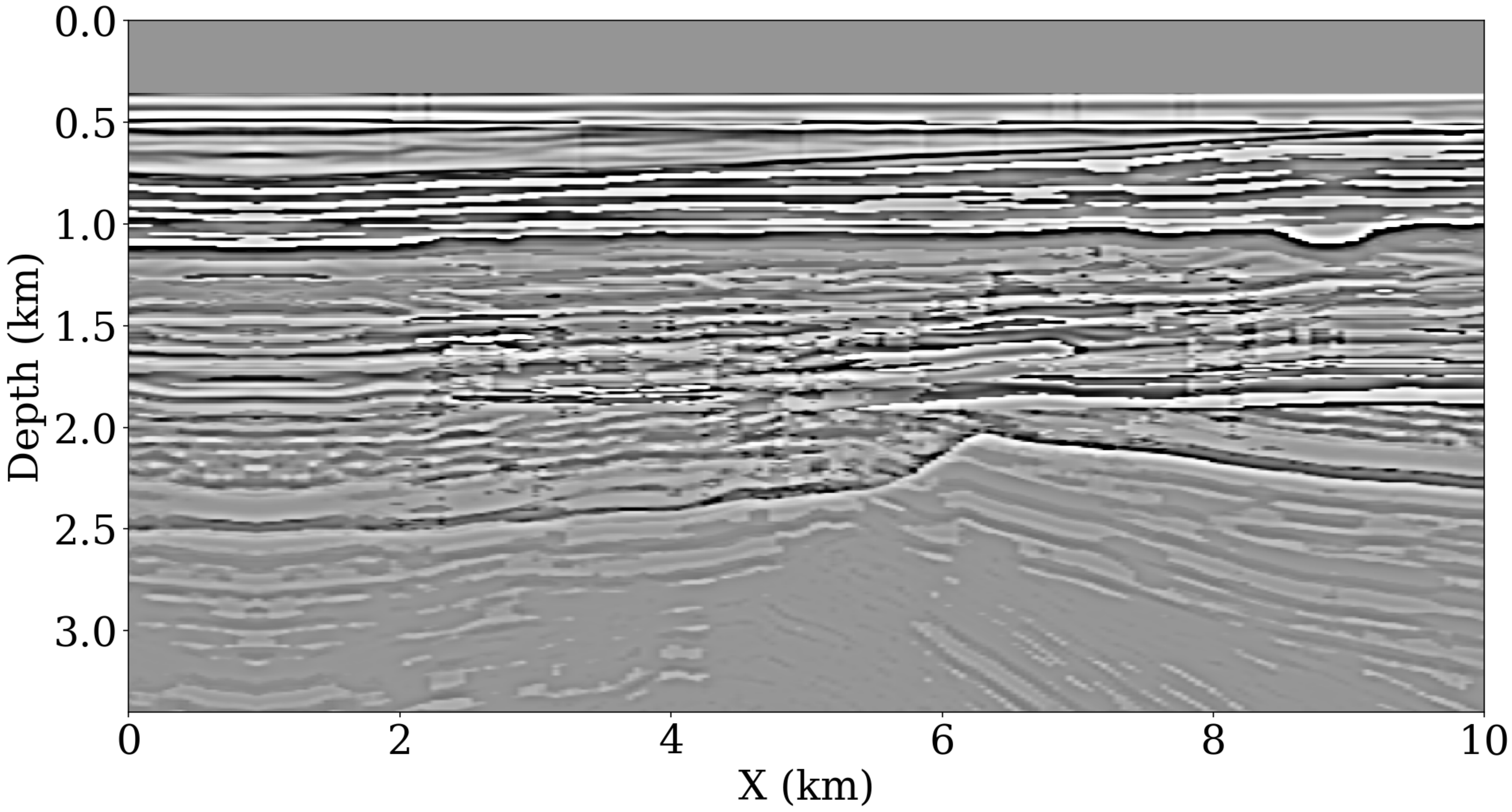
True Density



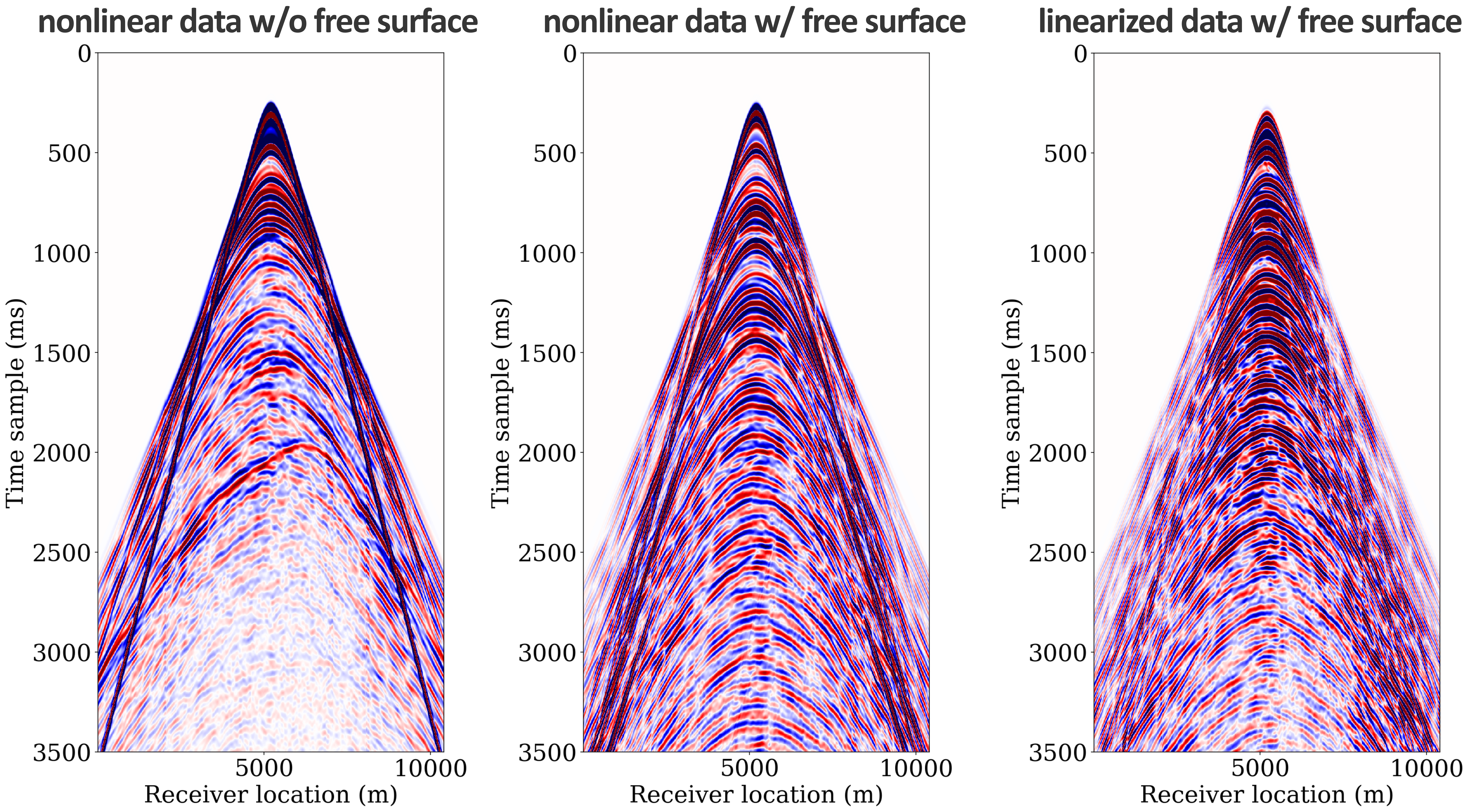
Background density



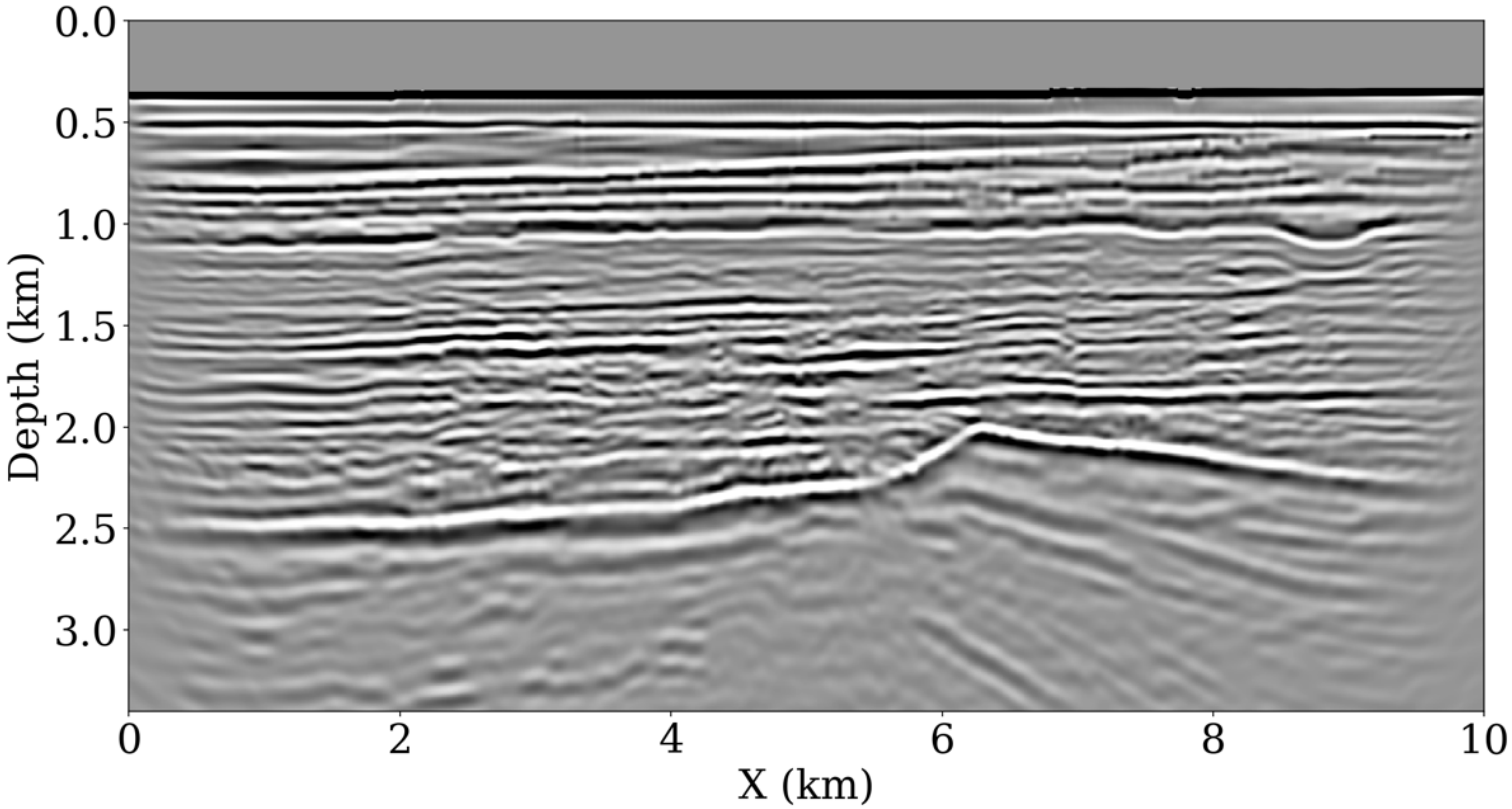
Velocity Perturbation



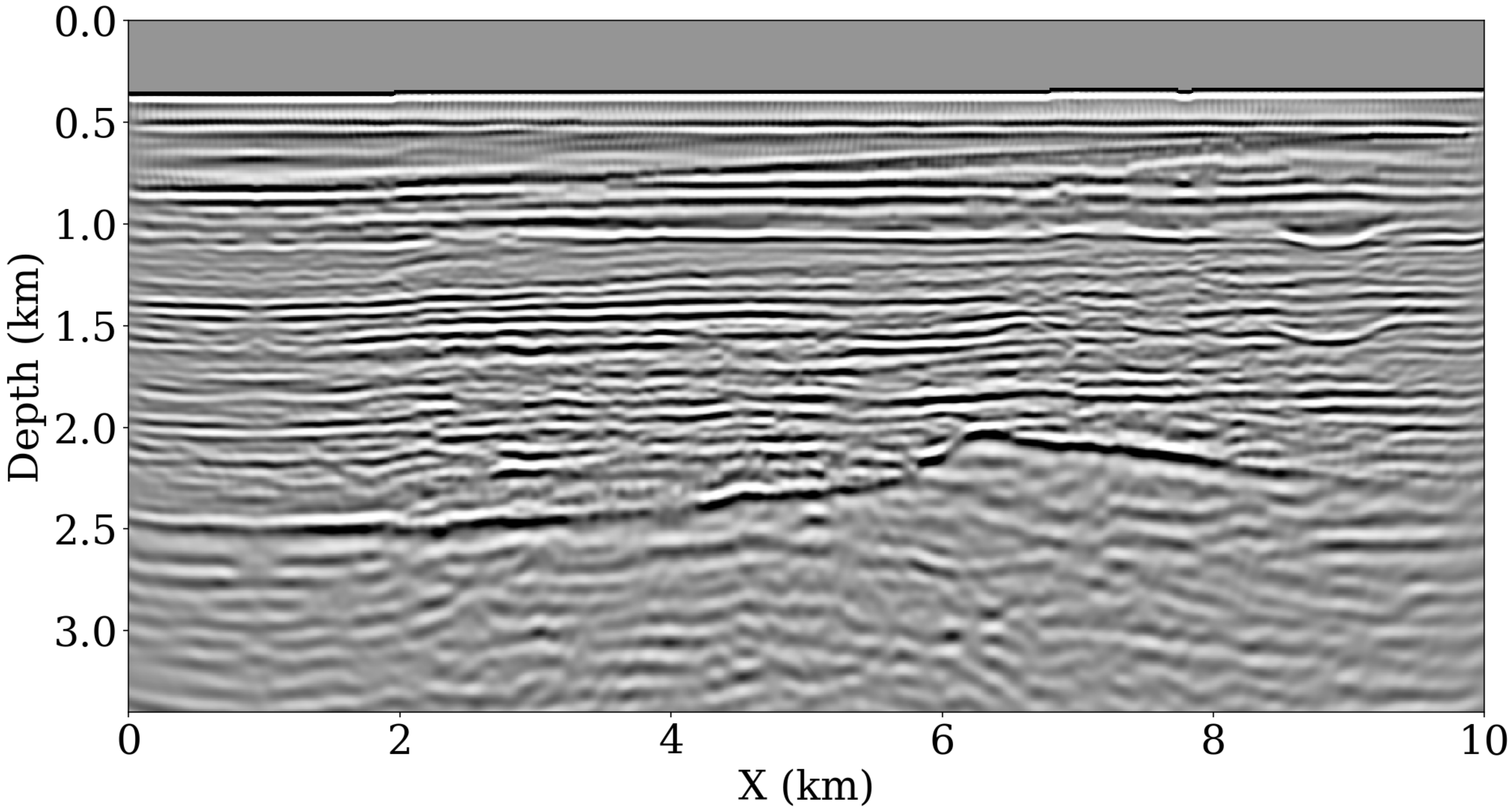
Dense OBN – BG compass model



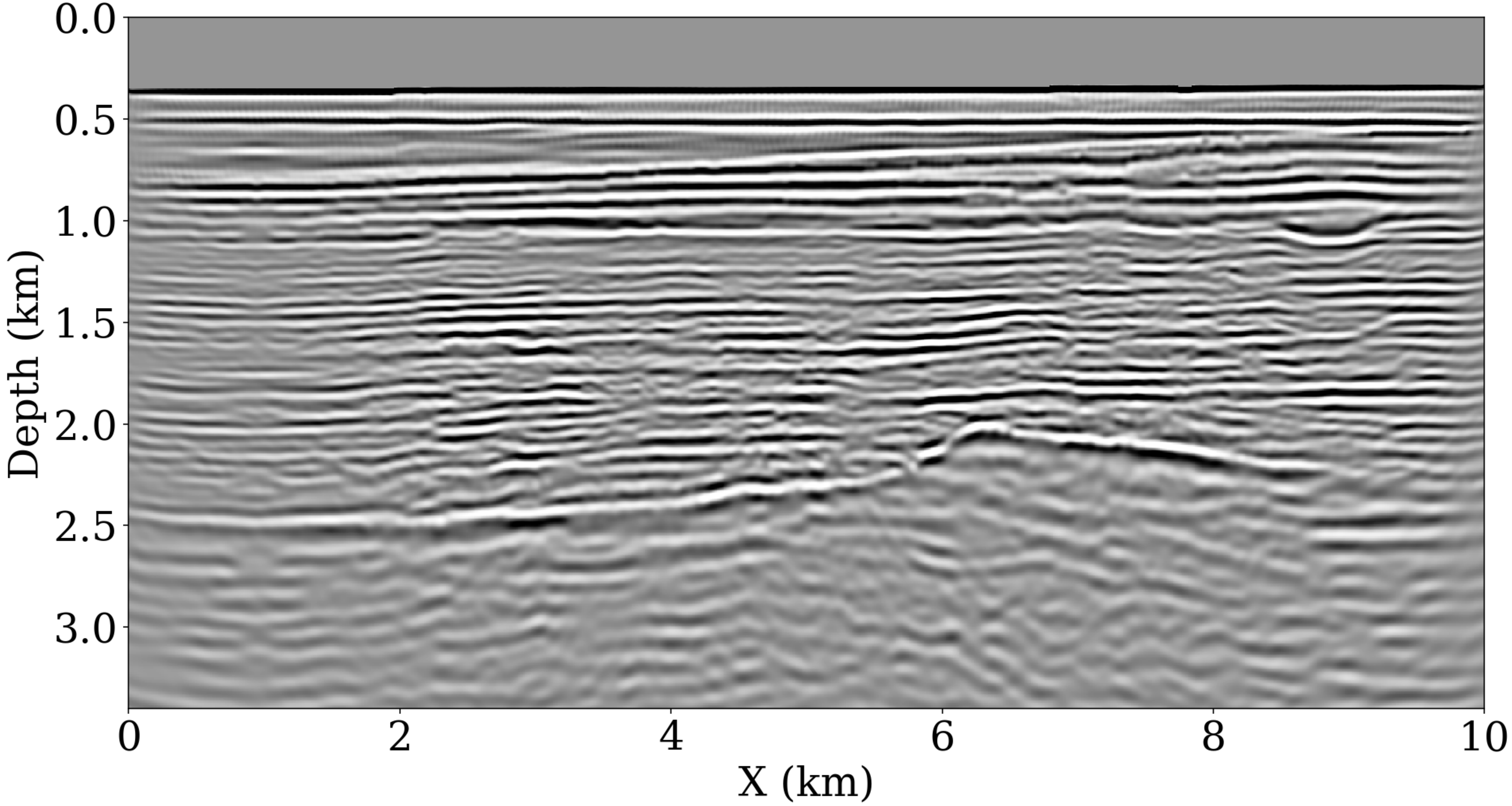
RTM – no free surface



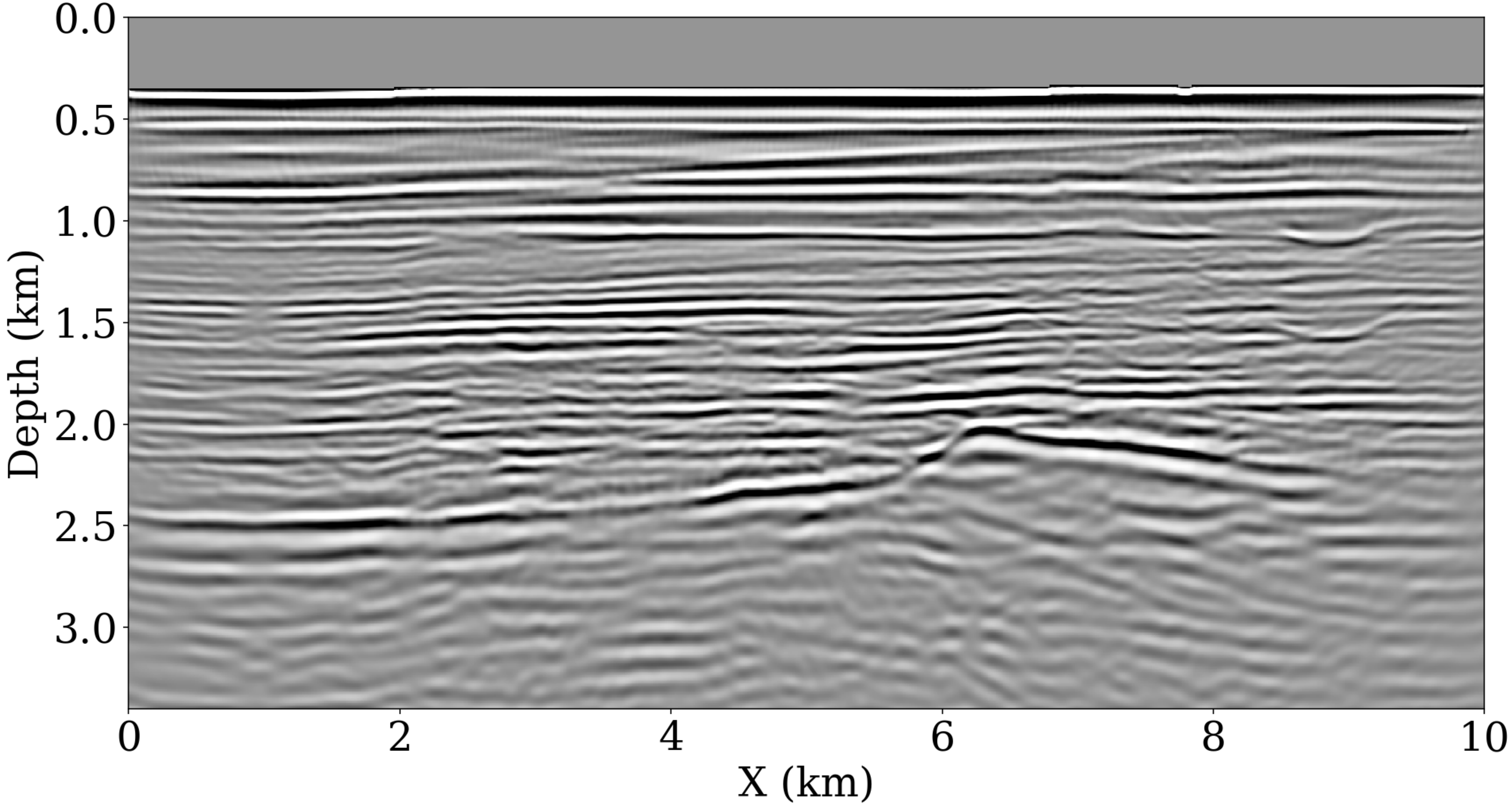
RTM – data w/ free surface



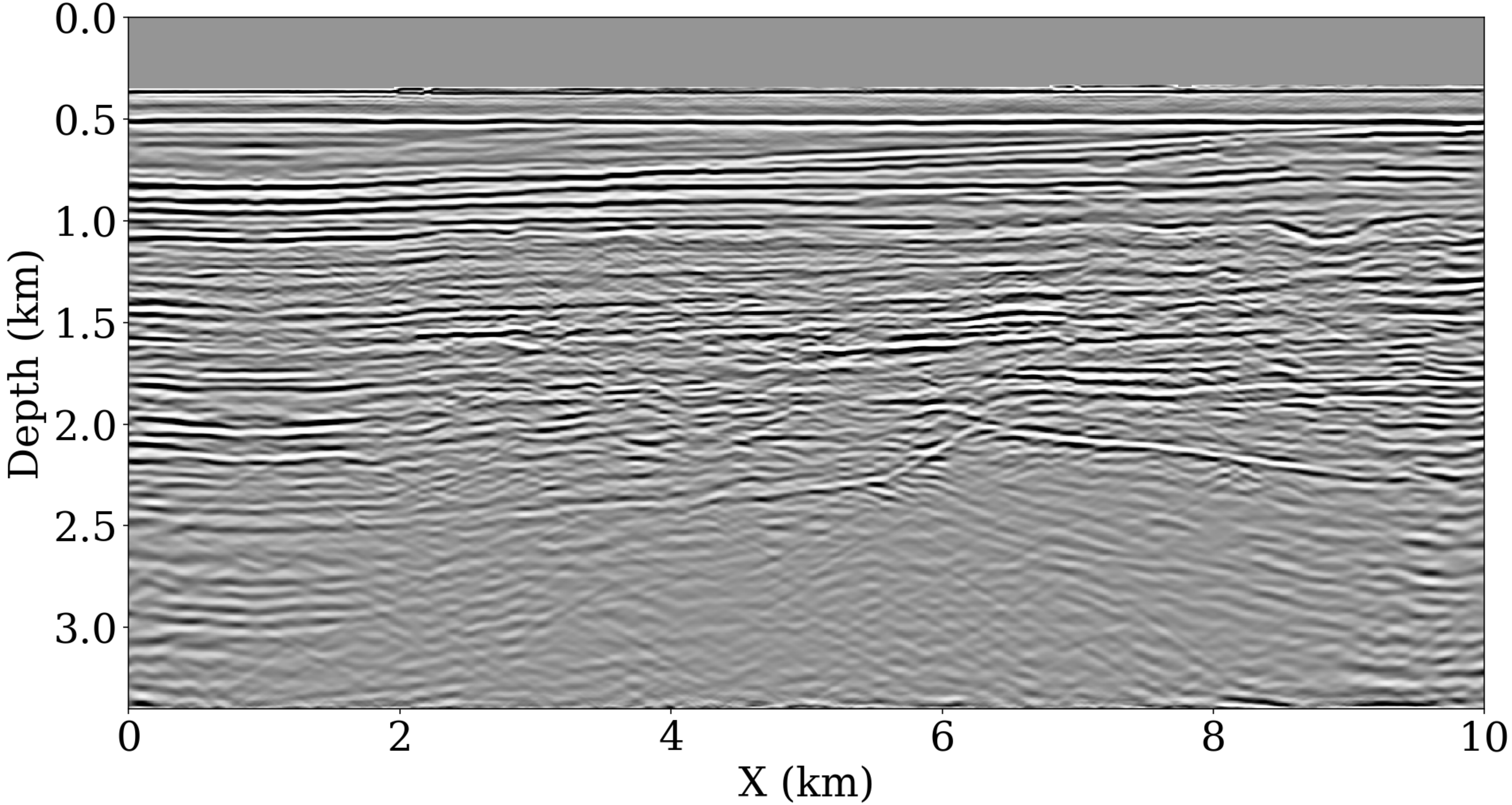
RTM – data w/ free surface + free surface BC



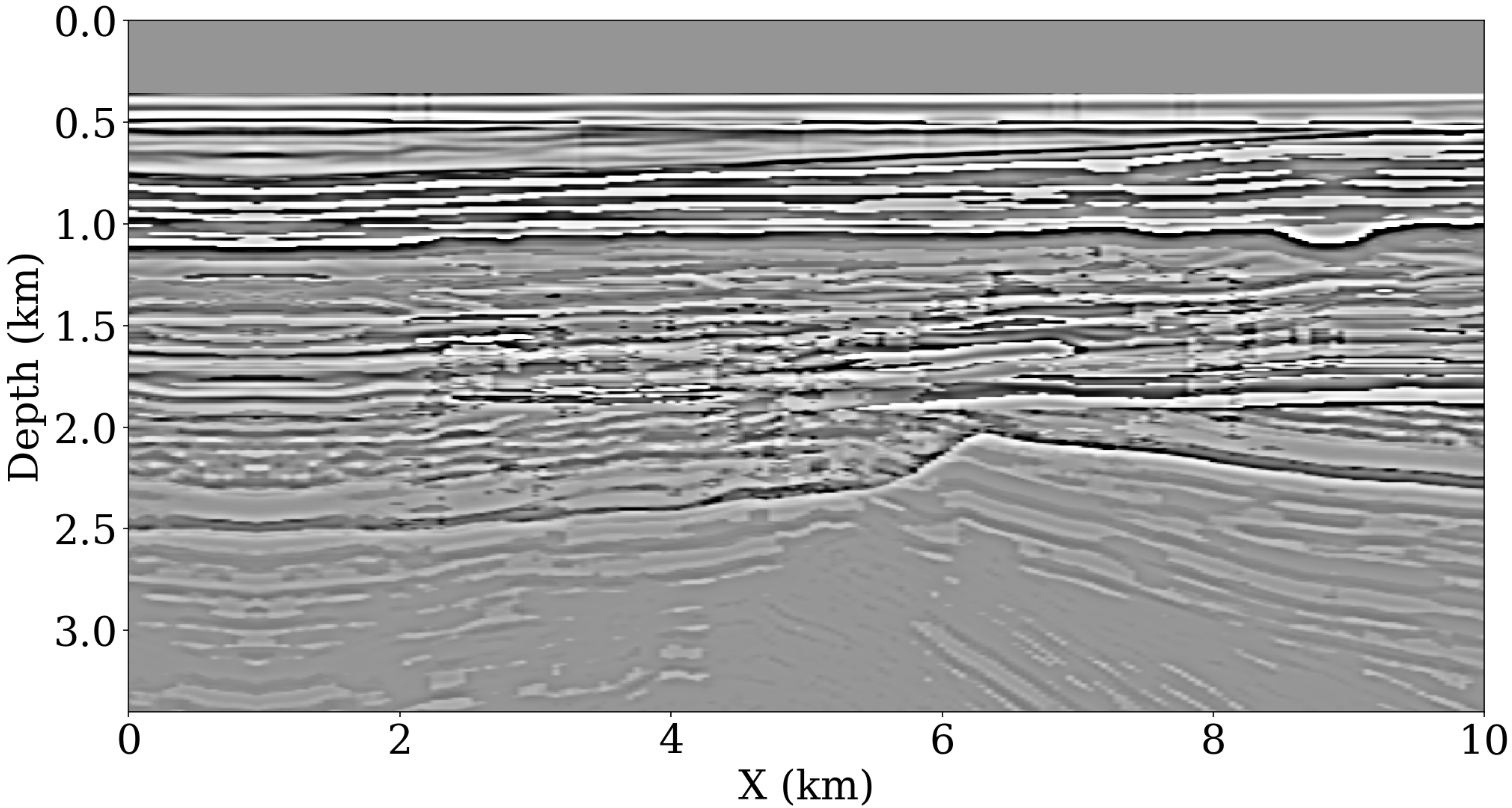
LS-RTM – data w/ free surface



LS-RTM – data w/ free surface + free surface BC



Velocity Perturbation



Observations

Surface-related multiples can be mitigated

- adding free surface boundary condition
- image multiples gives better aperture etc.
- no extra cost or data handling
- relies on sparse SP-LSRTM
- reduce costs of inversion via randomized subsampling

A tale of deblending…

linearized Born data $\neq$ blended data

- deblend & deghost data
- RTM or LS-RTM
- **pro:** industry standard; **con:** expensive (app. cost 1 extra RTM)

image blended data directly

- deblend, deghost during SP-LSRTM
- **pro:** avoids extra step & cheap, **con:** no QC on data

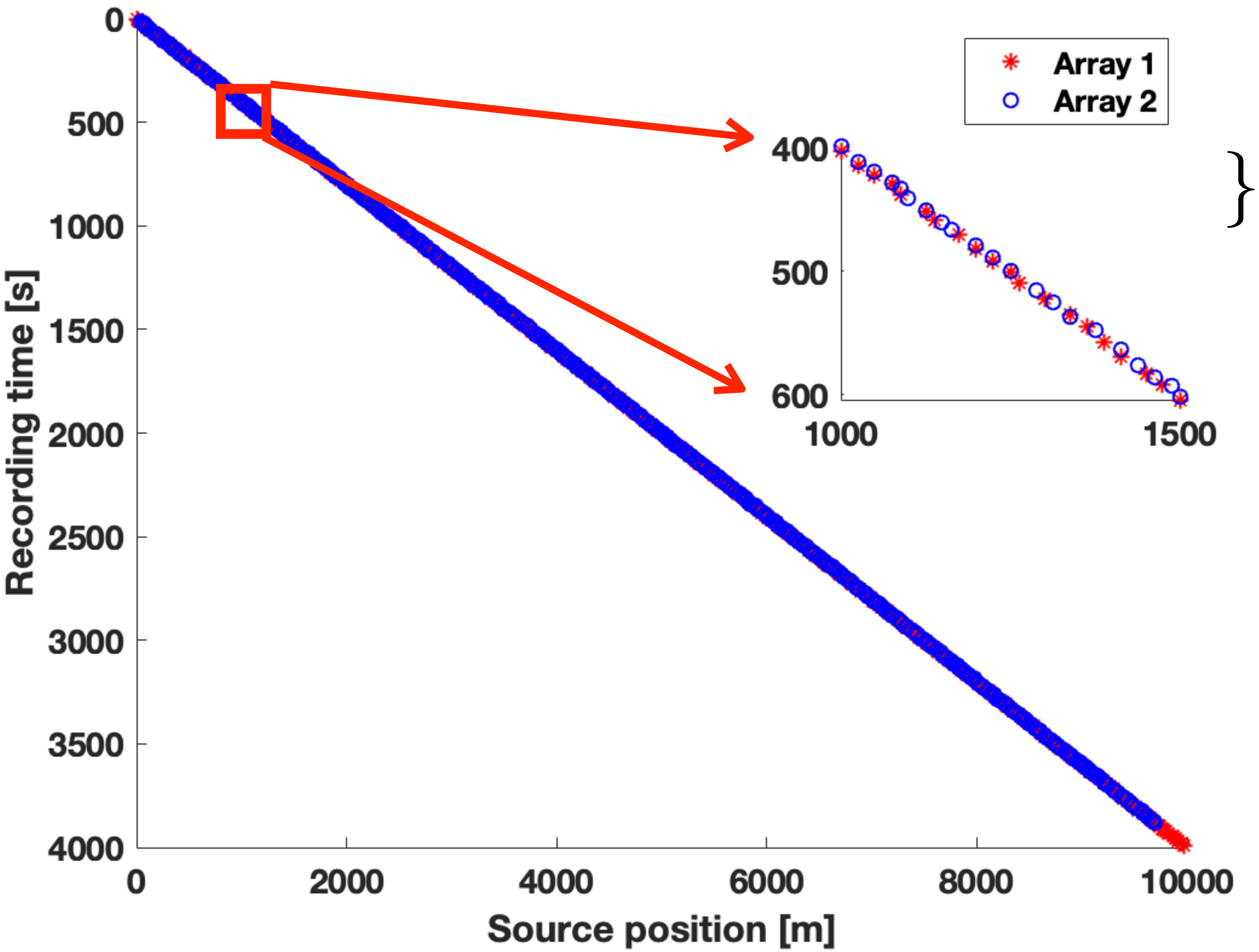
validate by comparing

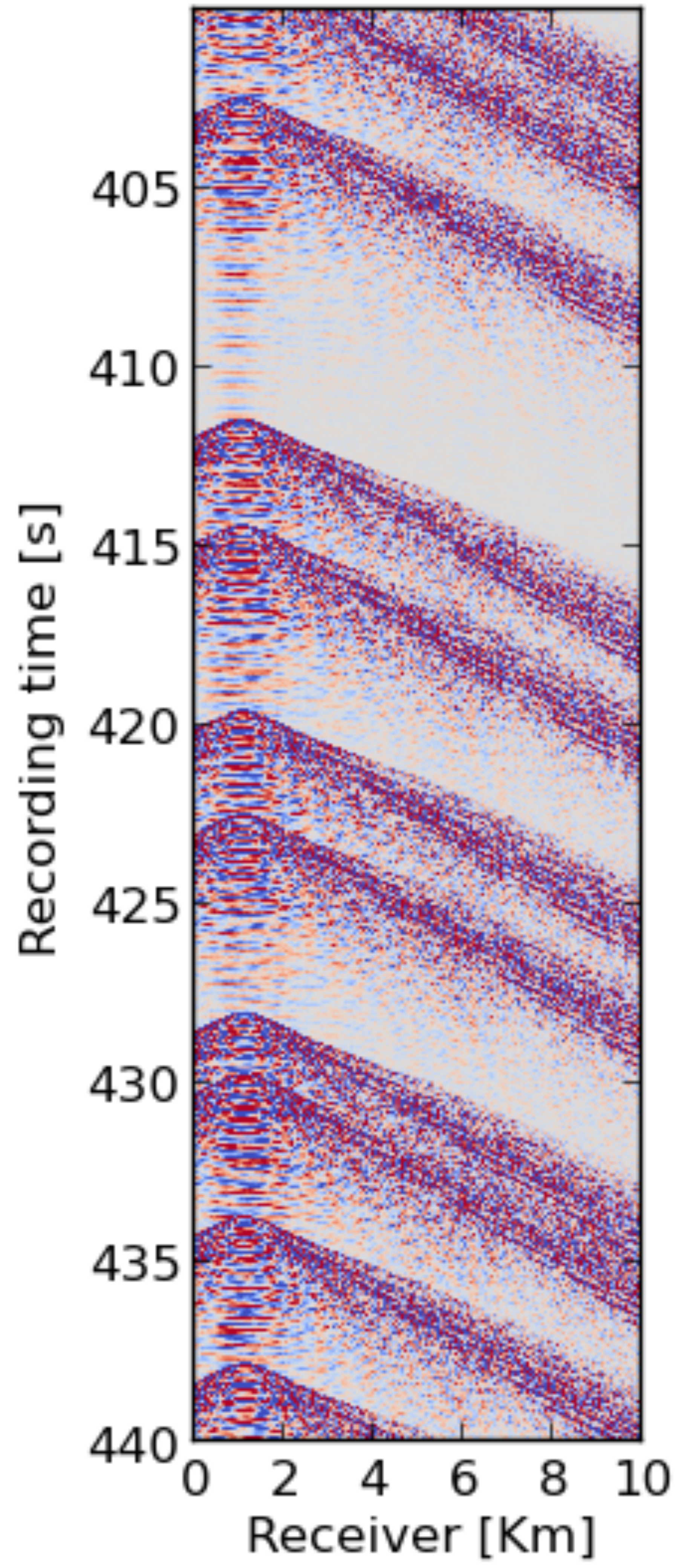
- **deblend first and then image $\Longleftrightarrow$ deblend while image**

Wason, Haneet, and Felix J. Herrmann. "Time-jittered ocean bottom seismic acquisition." *SEG Technical Program Expanded Abstracts 2013*. Society of Exploration Geophysicists, 2013. 1-6.

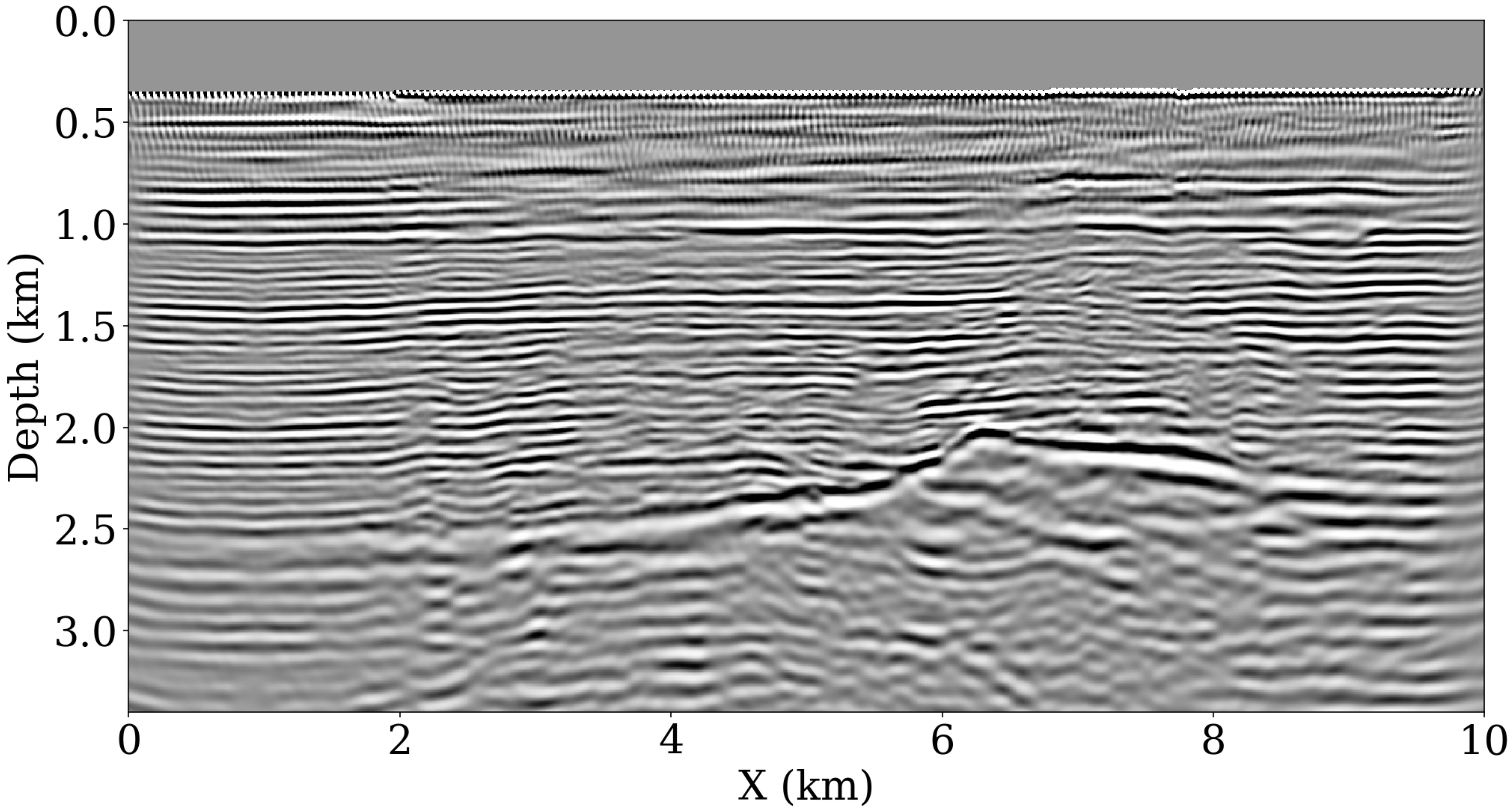
Time-jittered OBC acquisition

[1 source vessels, two airguns, speed = 5 knots, underlying grid: 25 m]
[no. of jittered source location is half the number of sources in ideal periodic survey w/o overlap]

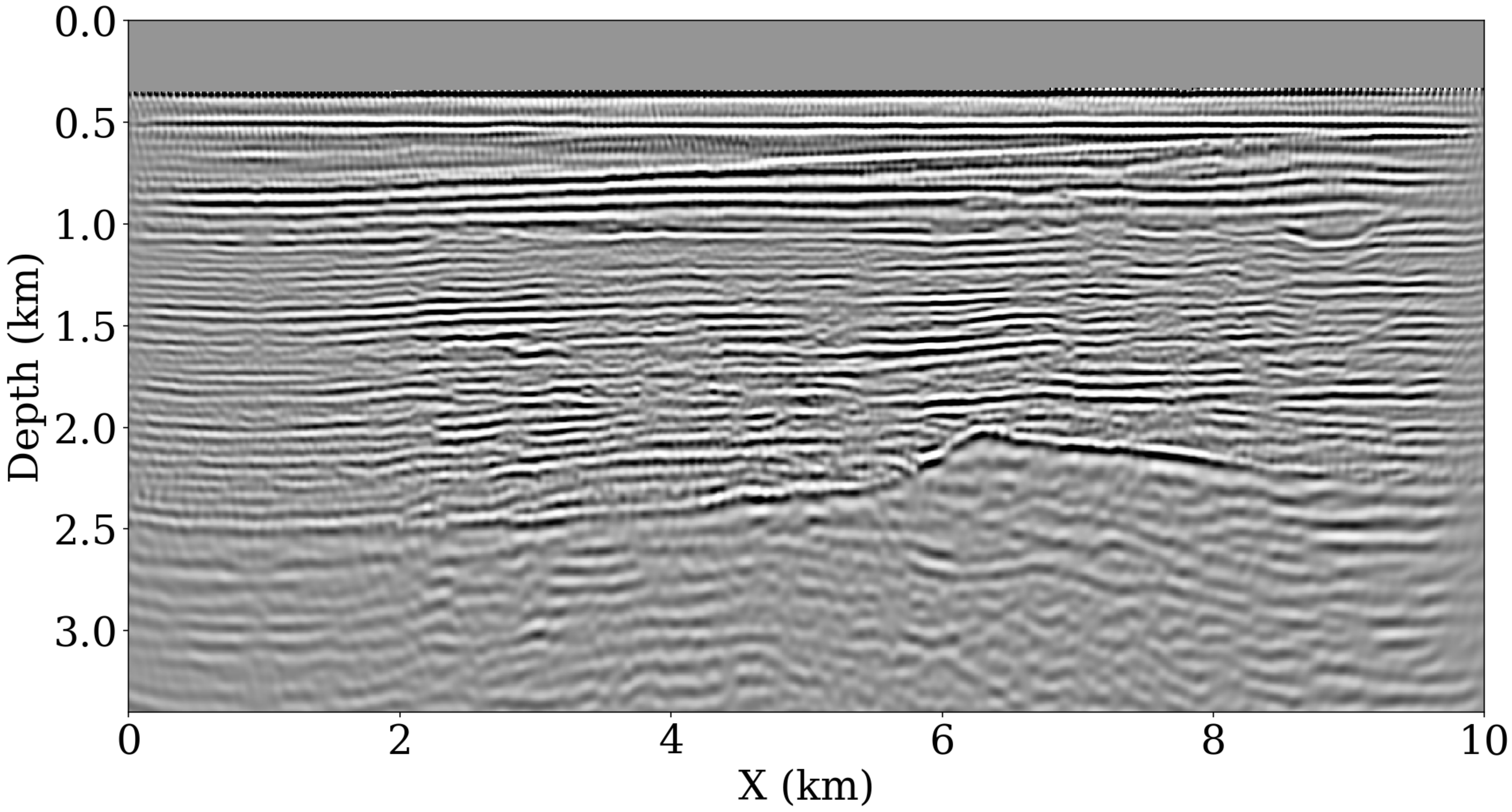




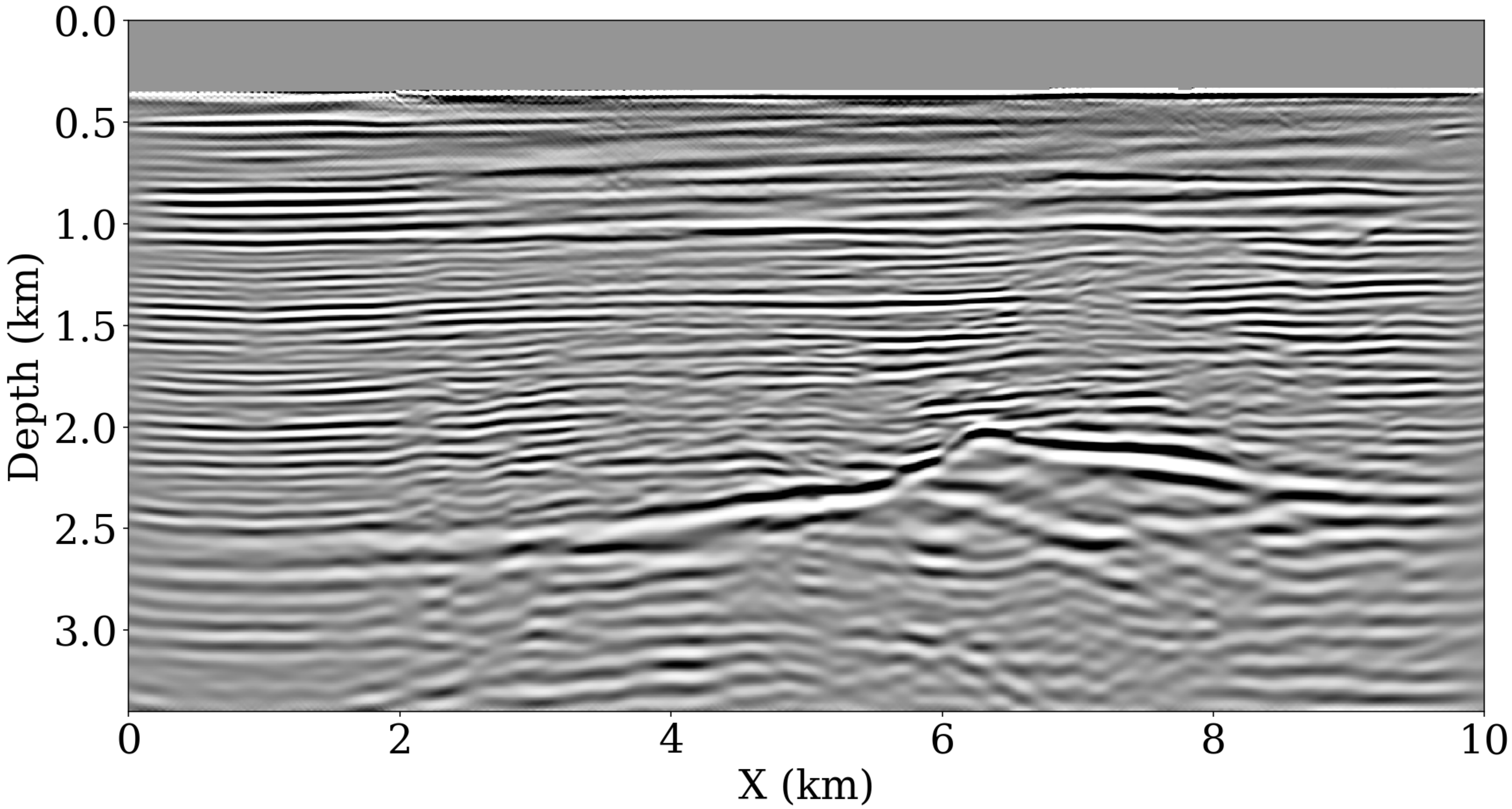
RTM – deblend first and then image – w/ free surface



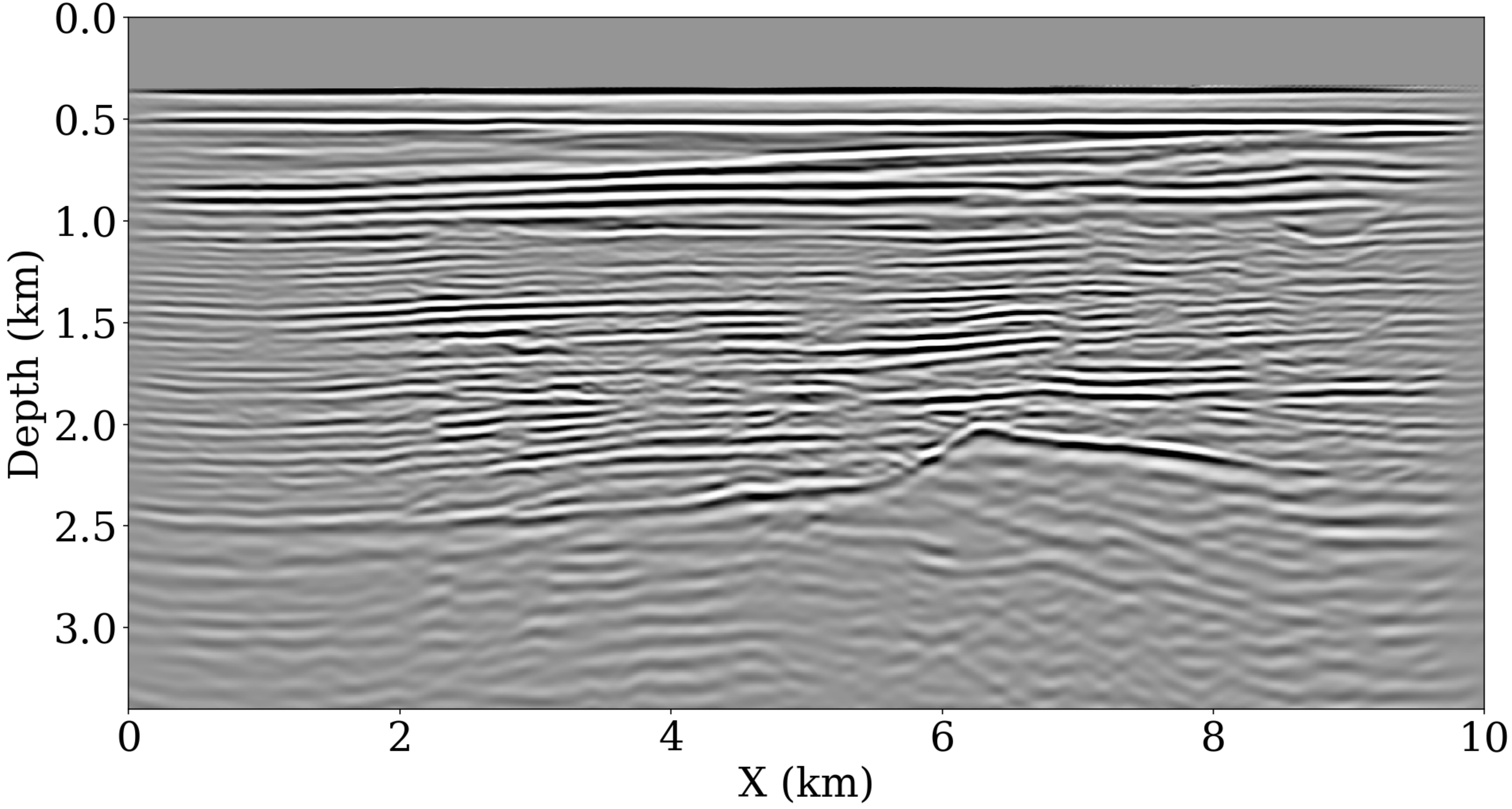
RTM w/ poor man's deblending



SP-LSRTM deblend first then image



LS-RTM w/ poor man's deblending

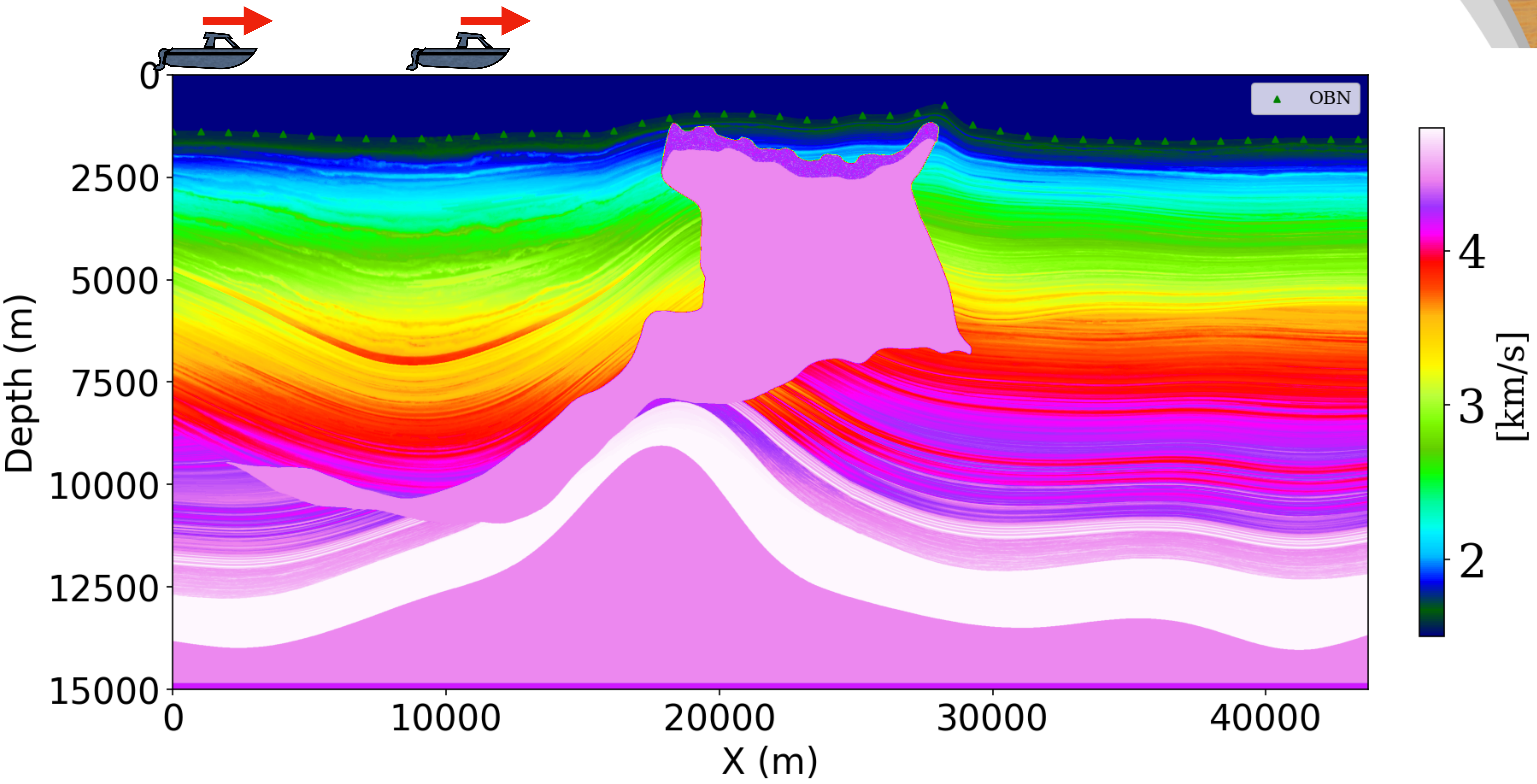


Observations

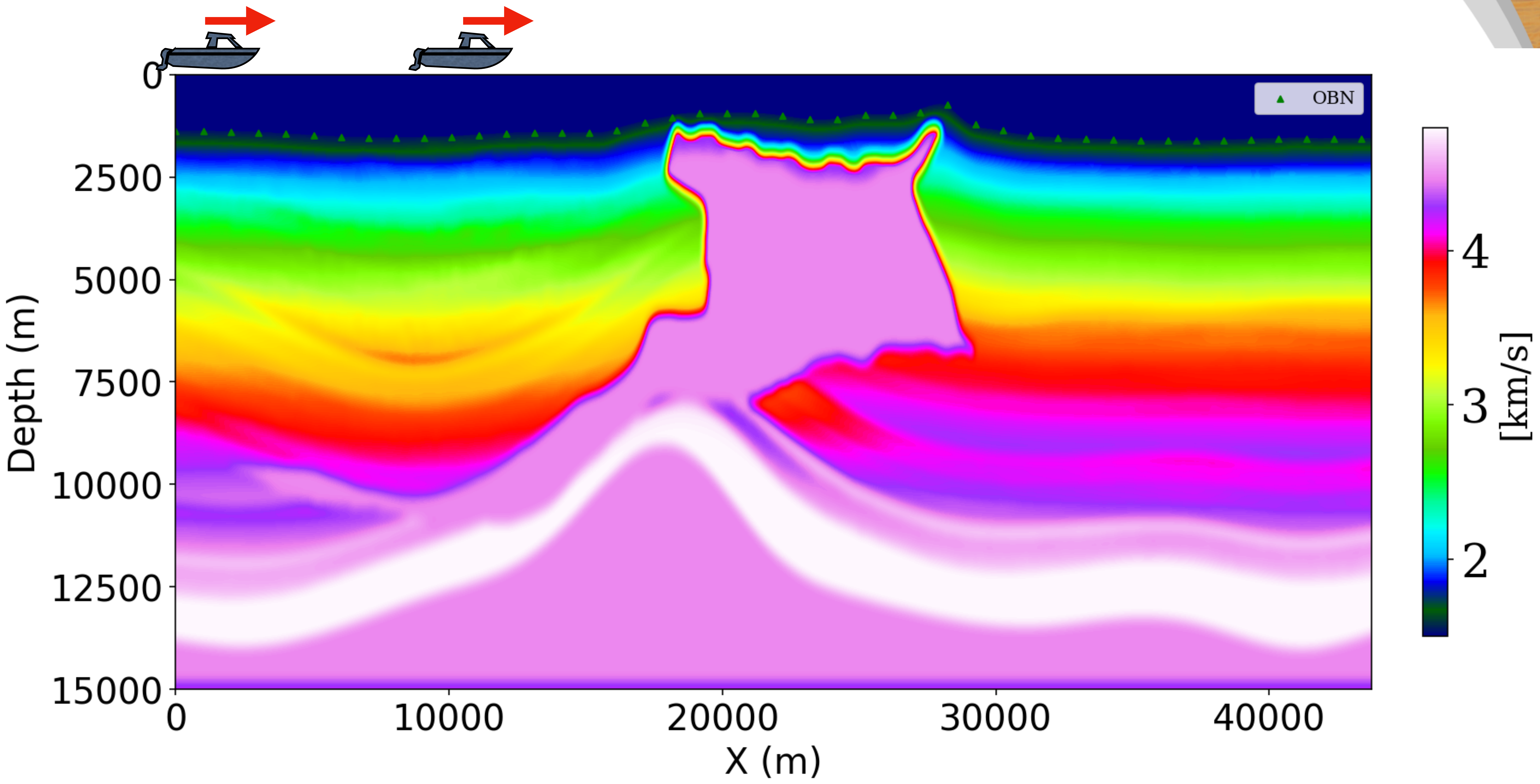
Deblending can be handled during imaging

- during sparse SP-LSRTM
- no extra cost
- reduce costs inversion via randomized subsampling

Stretched SEAM – max offset 43.75 km ($\Delta x \rightarrow 12.5\text{m}$)



Stretched SEAM – background velocity

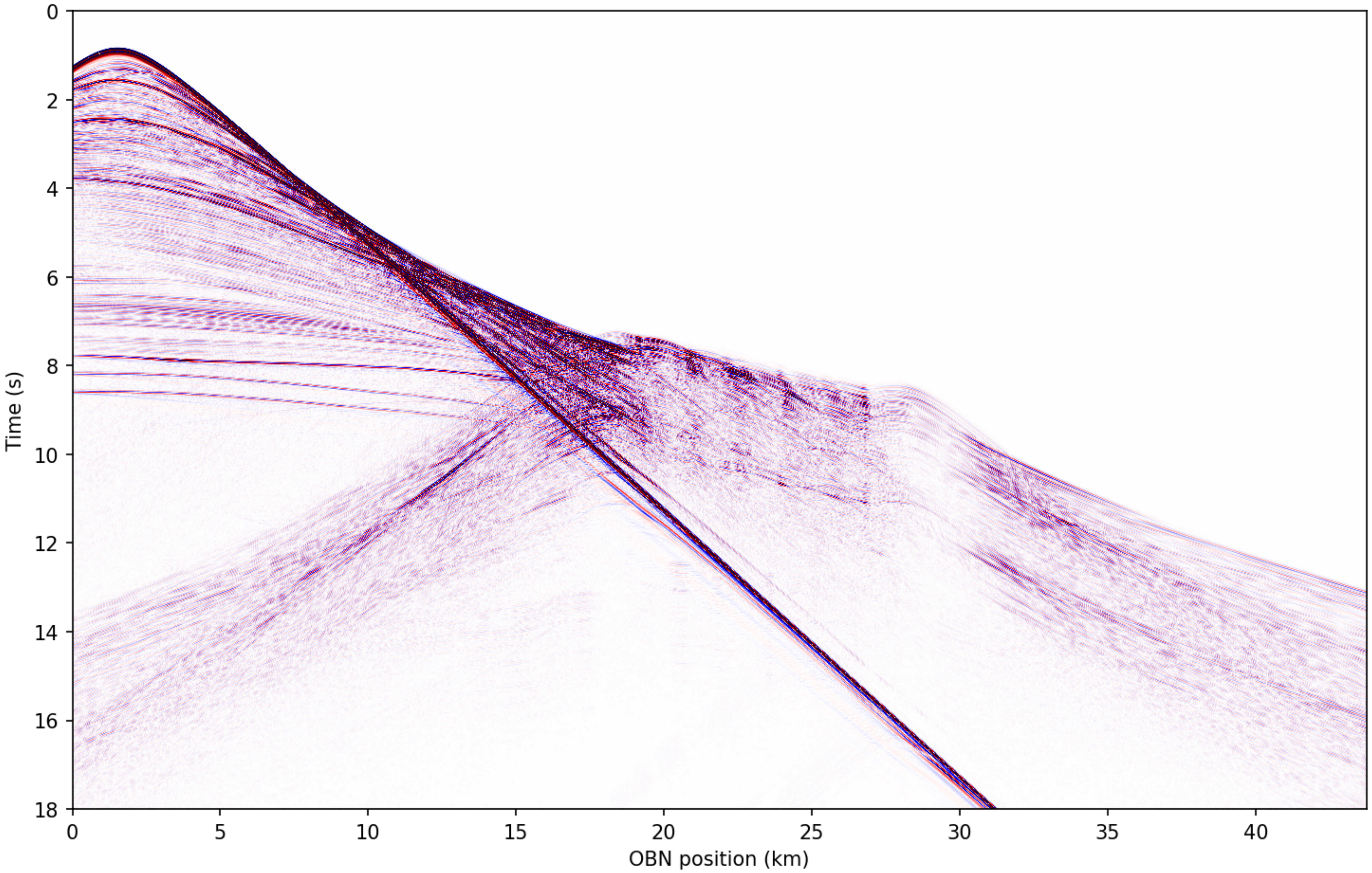


Imaging w/ sparse OBN – no free surface

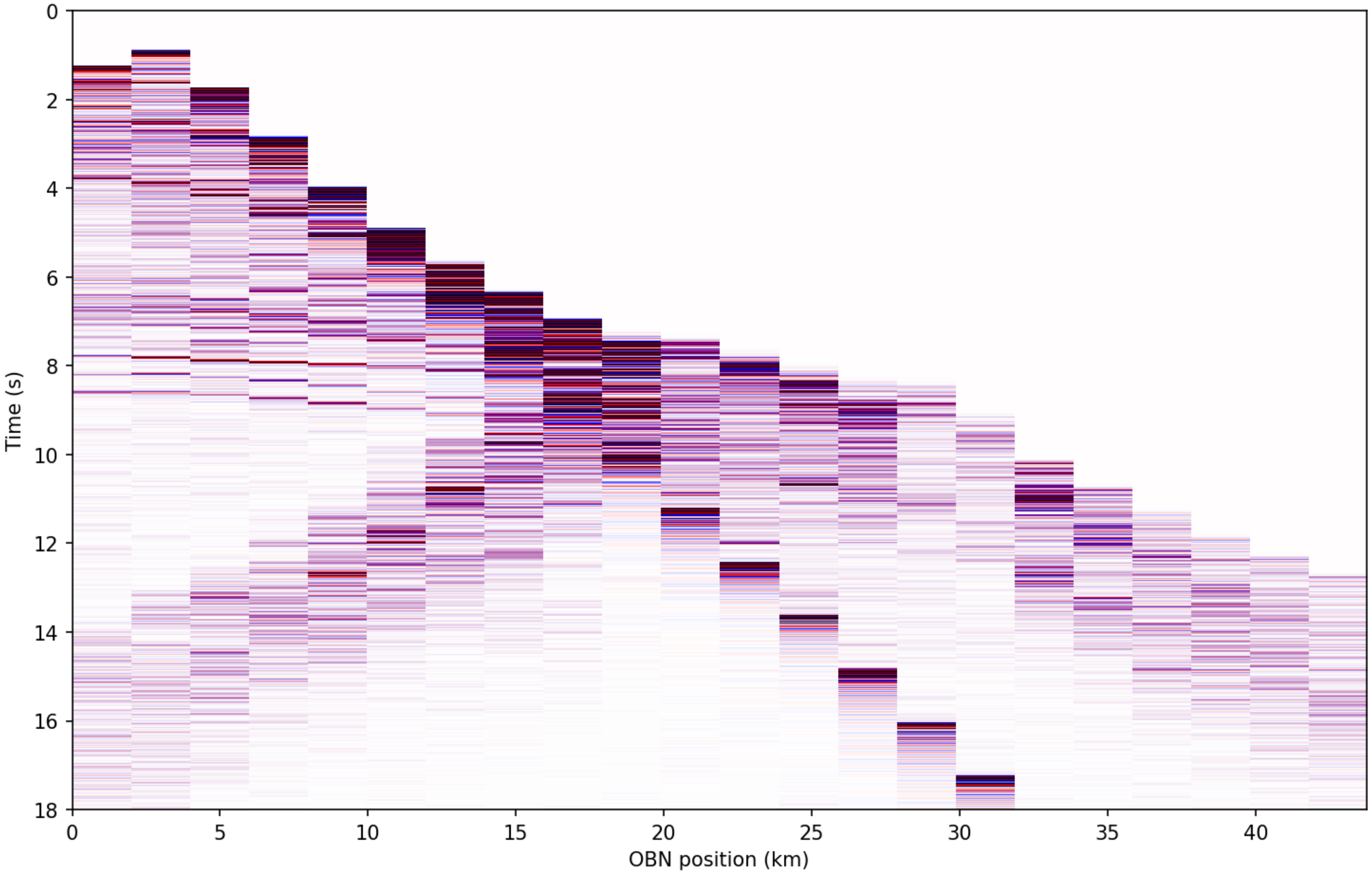
2D SEAM model:

- 44 OBN (1000m spacing)
- 1751 sources (25m spacing, 10m depth)
- source wavelet w/ 14.5Hz peak effectively 3–41 Hz
- 18 seconds recording w/ 45000 timesteps
- imaging & modeling w/ density
- inverse-scattering imaging condition
- 11 iterations w/ 4 source experiments each
- 1 data pass (1.5 X cost single RTM)
- 20 min per gradient

Densely sampled common receiver gather OBN
– no free surface – no blending

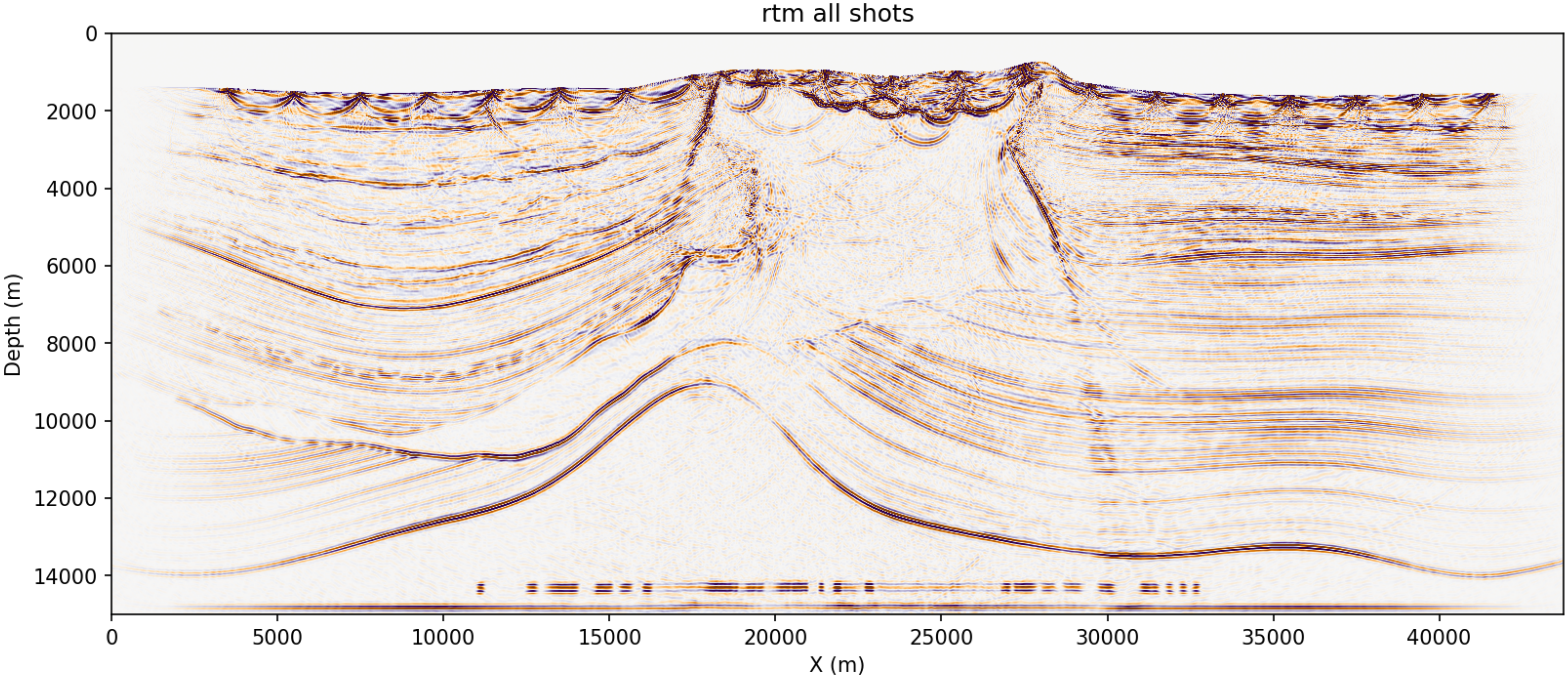


Sparse OBN common shot gather 1km spacing
– no free surface – no blending

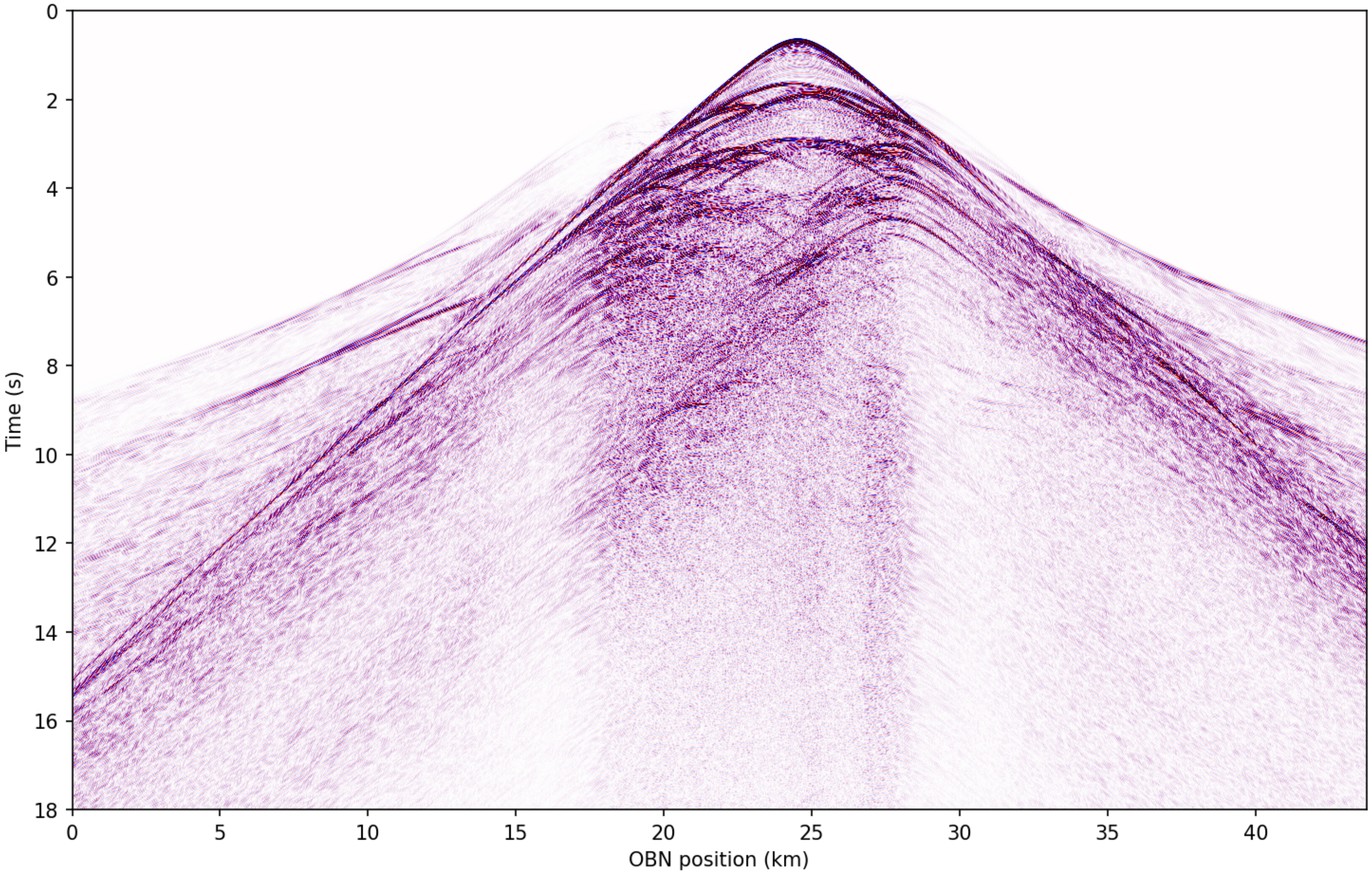


RTM – no free surface – no blending

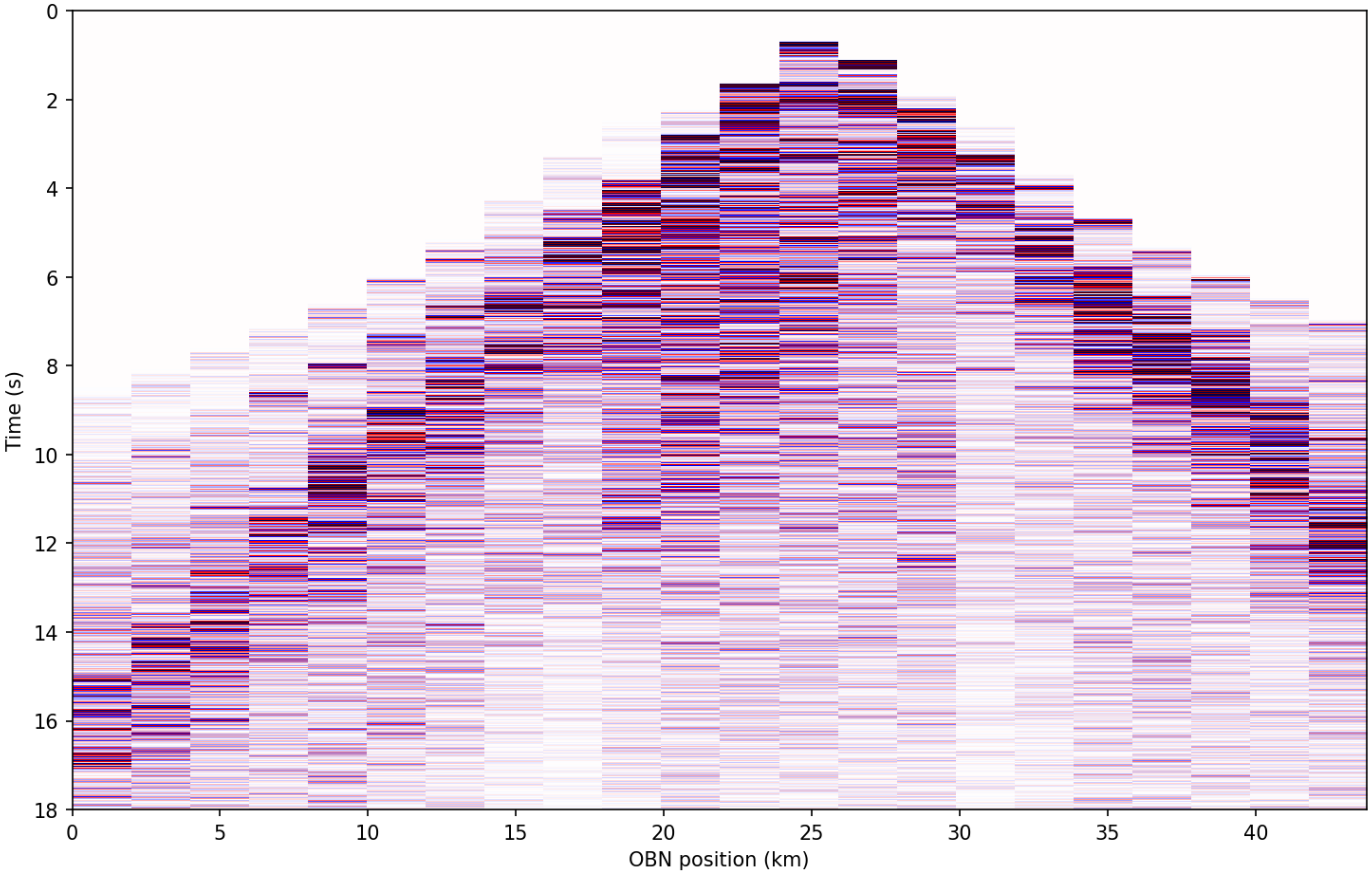
no processing



Dense common shot gather
– free surface – no blending

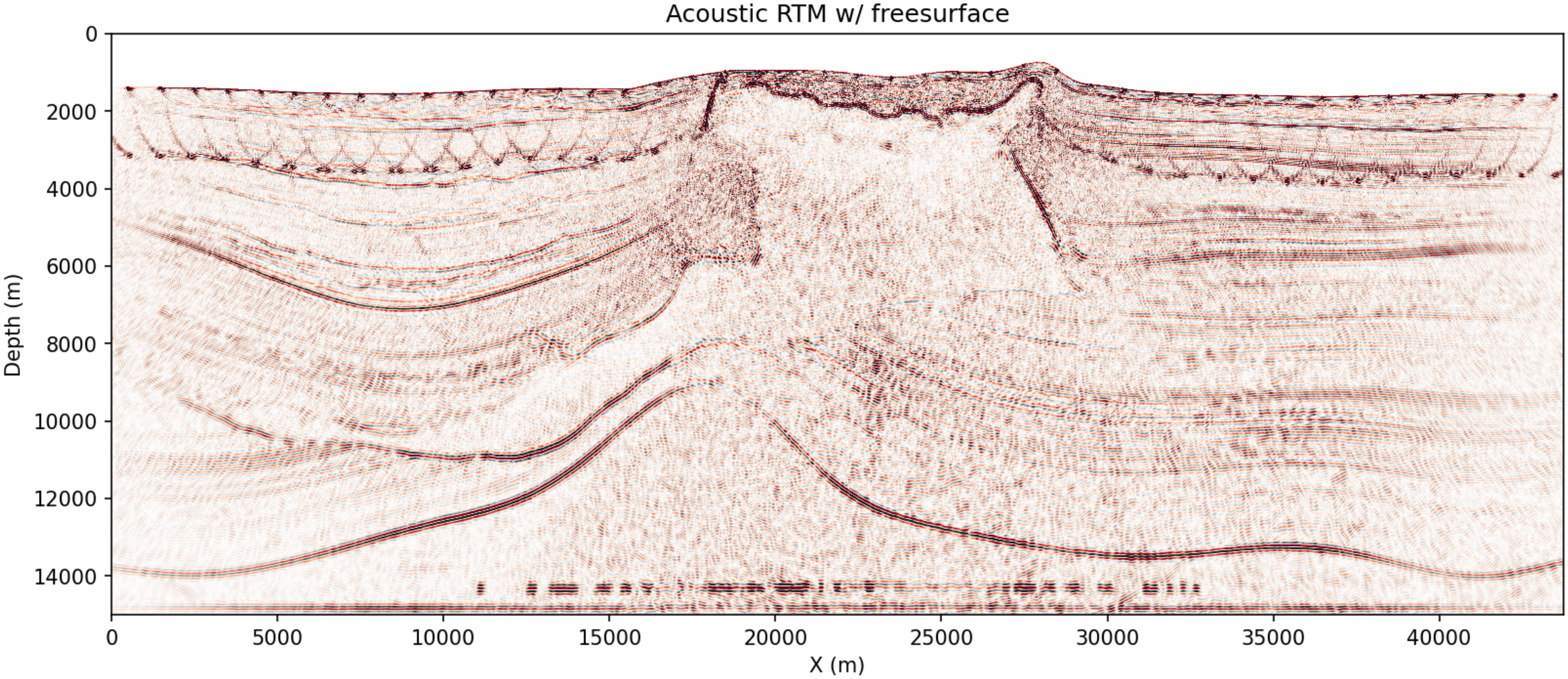


Sparse common shot gather
– free surface – no blending

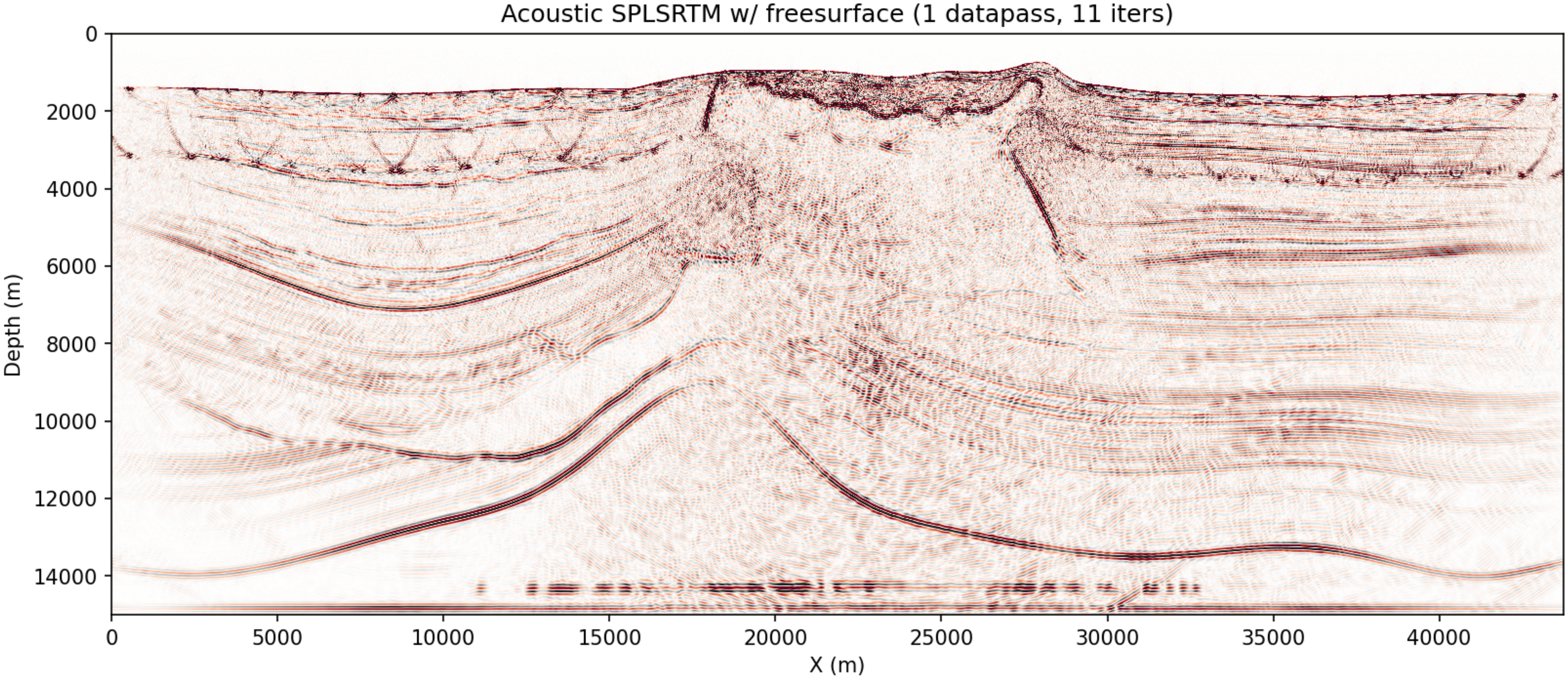


RTM – free surface – no blending - free surface BC

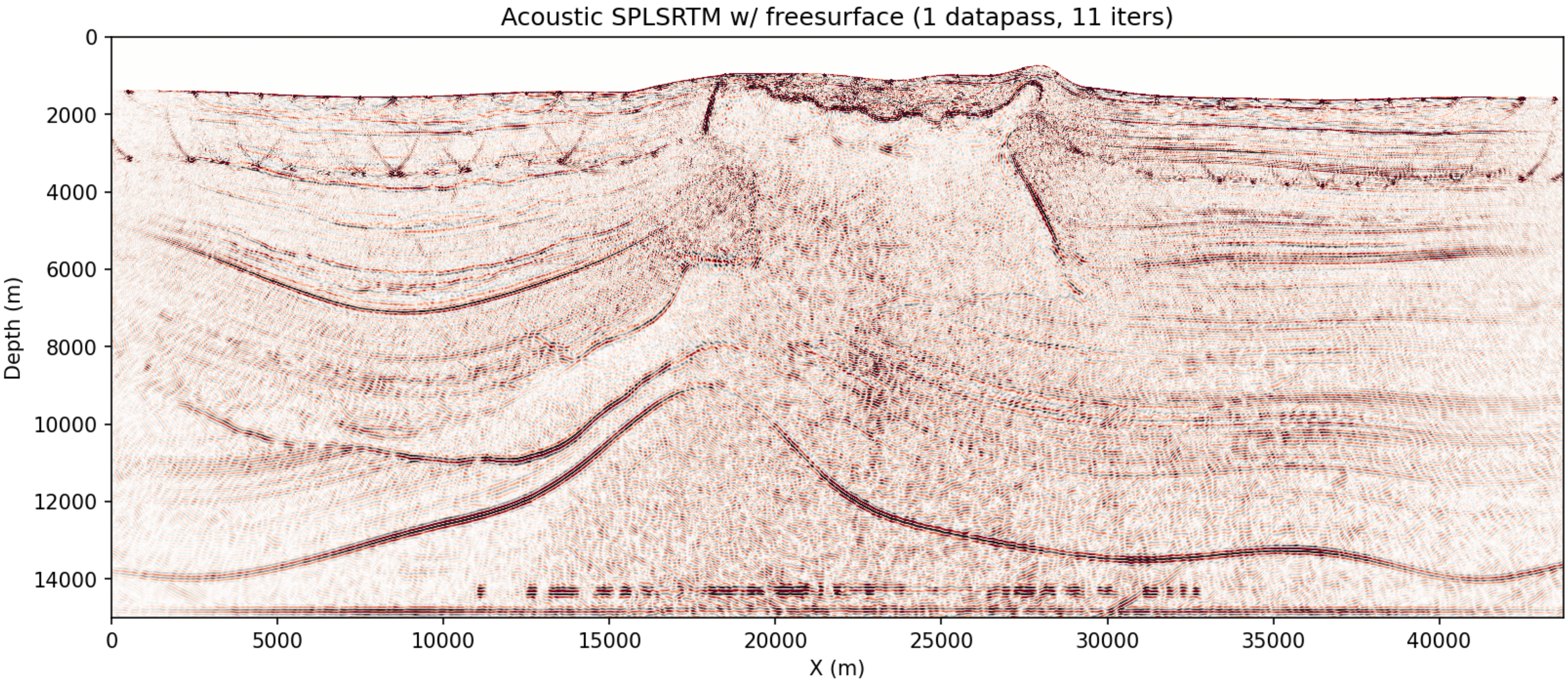
no processing



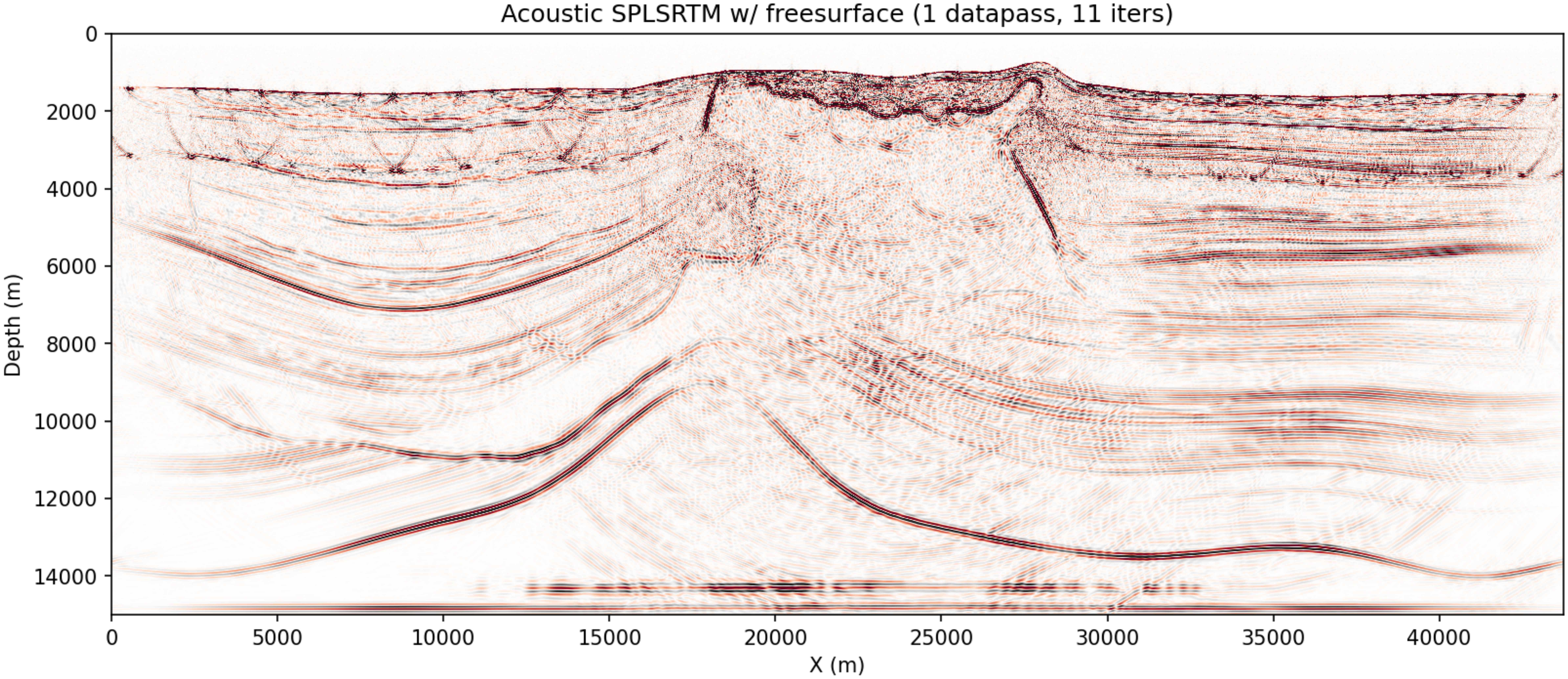
SP-LSRTM – free surface – no blending - free surface BC



SP-LSRTM – free surface – no blending - free surface BC – lower threshold



SP-LSRTM – free surface – no blending - free surface BC – higher threshold

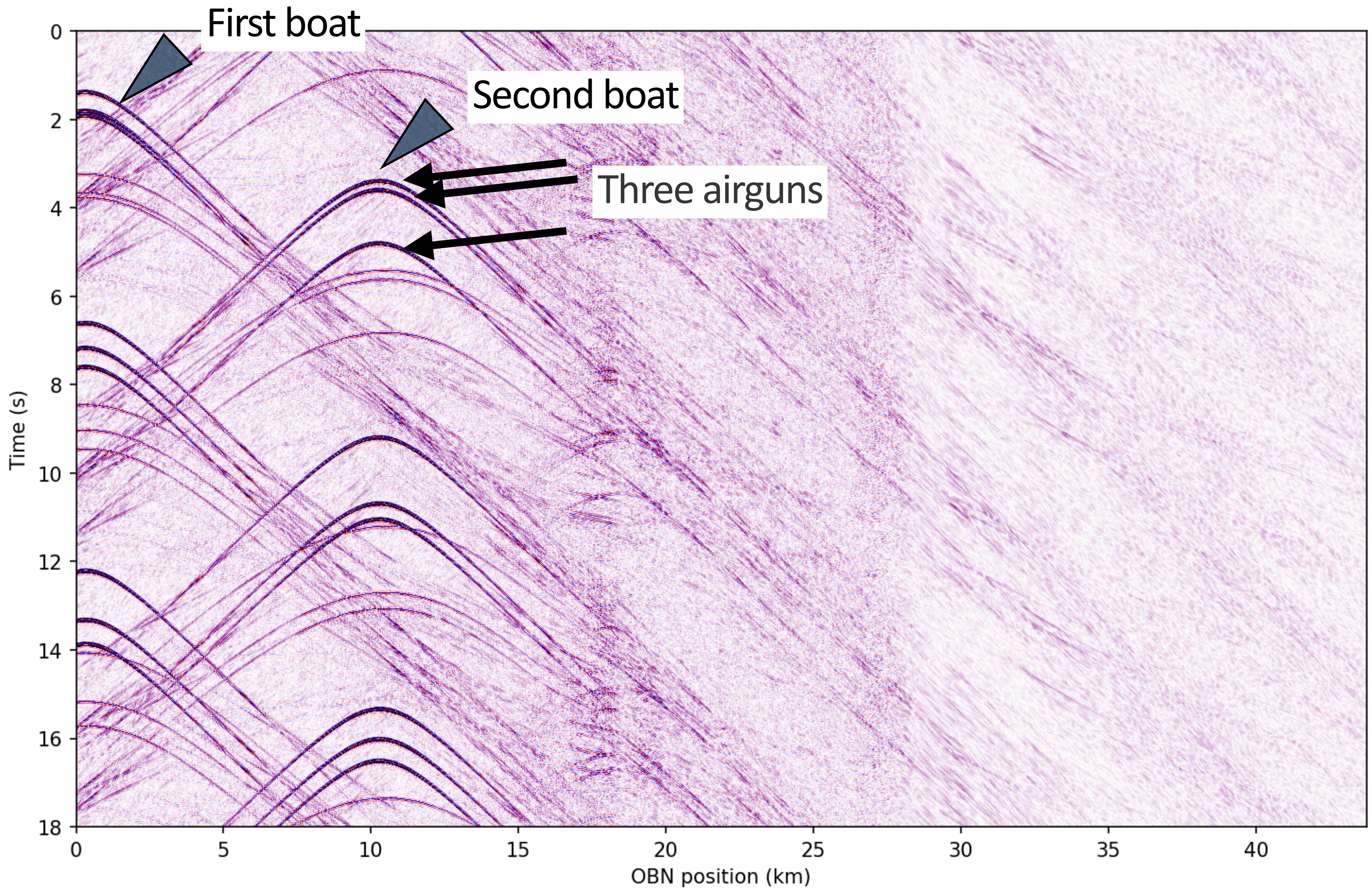


Imaging w/ continuous sim. source recording

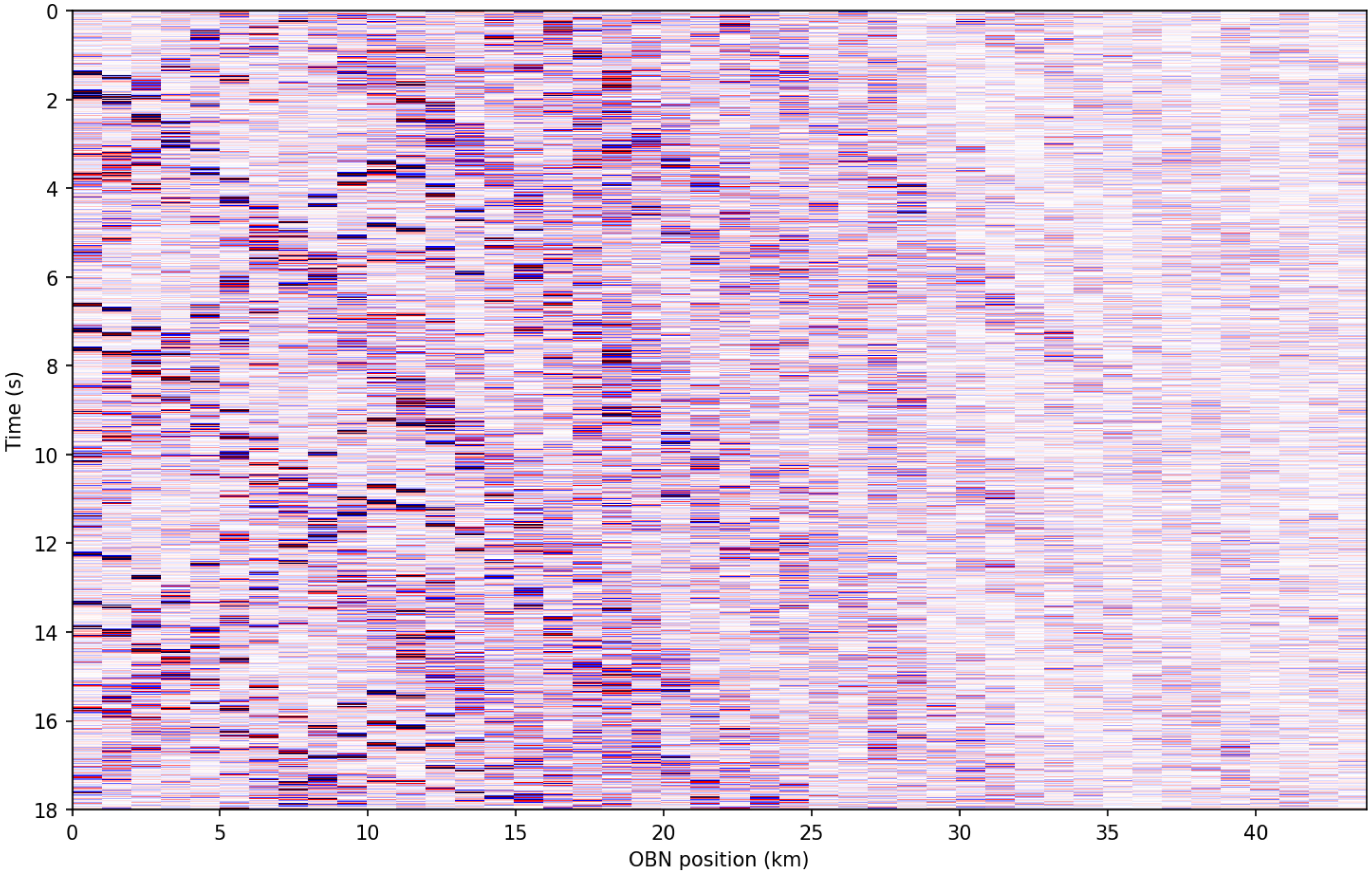
2D SEAM model:

- 2 boats 10km apart, 2.5m/s moving left to right
- 3 airguns per boat
- airgun 1 fires every 6 sec
- airgun 2,3 fires within ± 1 sec of airgun1
- 15m source spacing at the surface (2898 unique source positions)
- 3h45min recording (2200 x 6s)
- 14.5Hz peak source wavelet
- 1km OBN spacing (44 positions)

Dense continuous recording
– free surface – blended



Sparse continuous recording
– free surface – blended



RTM blended data w/ multiples – first iteration

RTM blended data w/ multiples – after thresholding

Conclusions

All in-one go imaging w/o processing may be feasible

- adding free surface boundary condition
- deblending on the fly
- no extra cost or data handling
- relies on sparse SP-LSRTM
- reduce costs inversion via randomized subsampling

Next steps

- improved denoising, e.g. weighted thresholding or ML
- time-lapse & extension to elastic
- inclusion of common image gathers