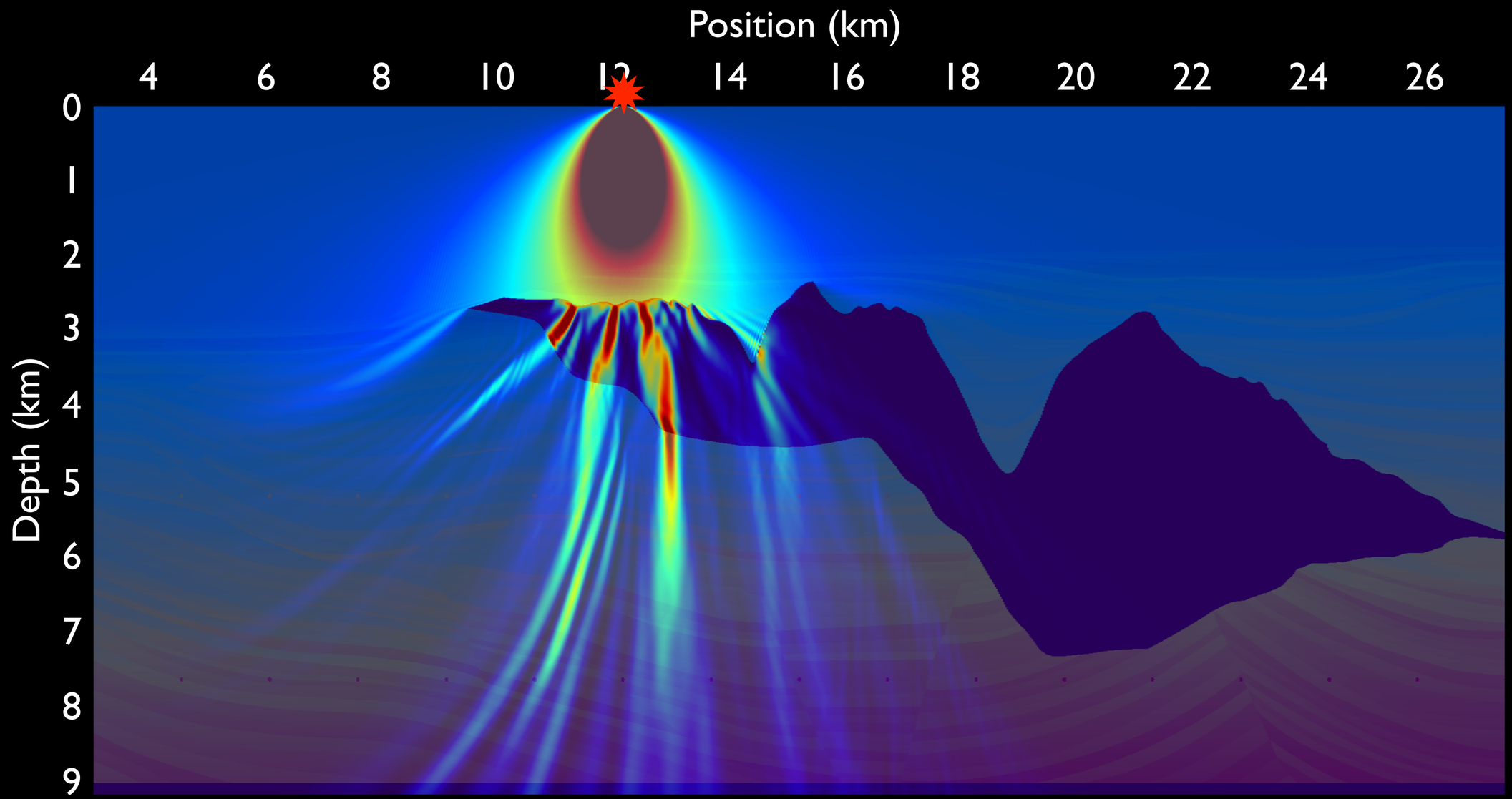
Subsalt imaging by target-oriented wavefield least-squares migration

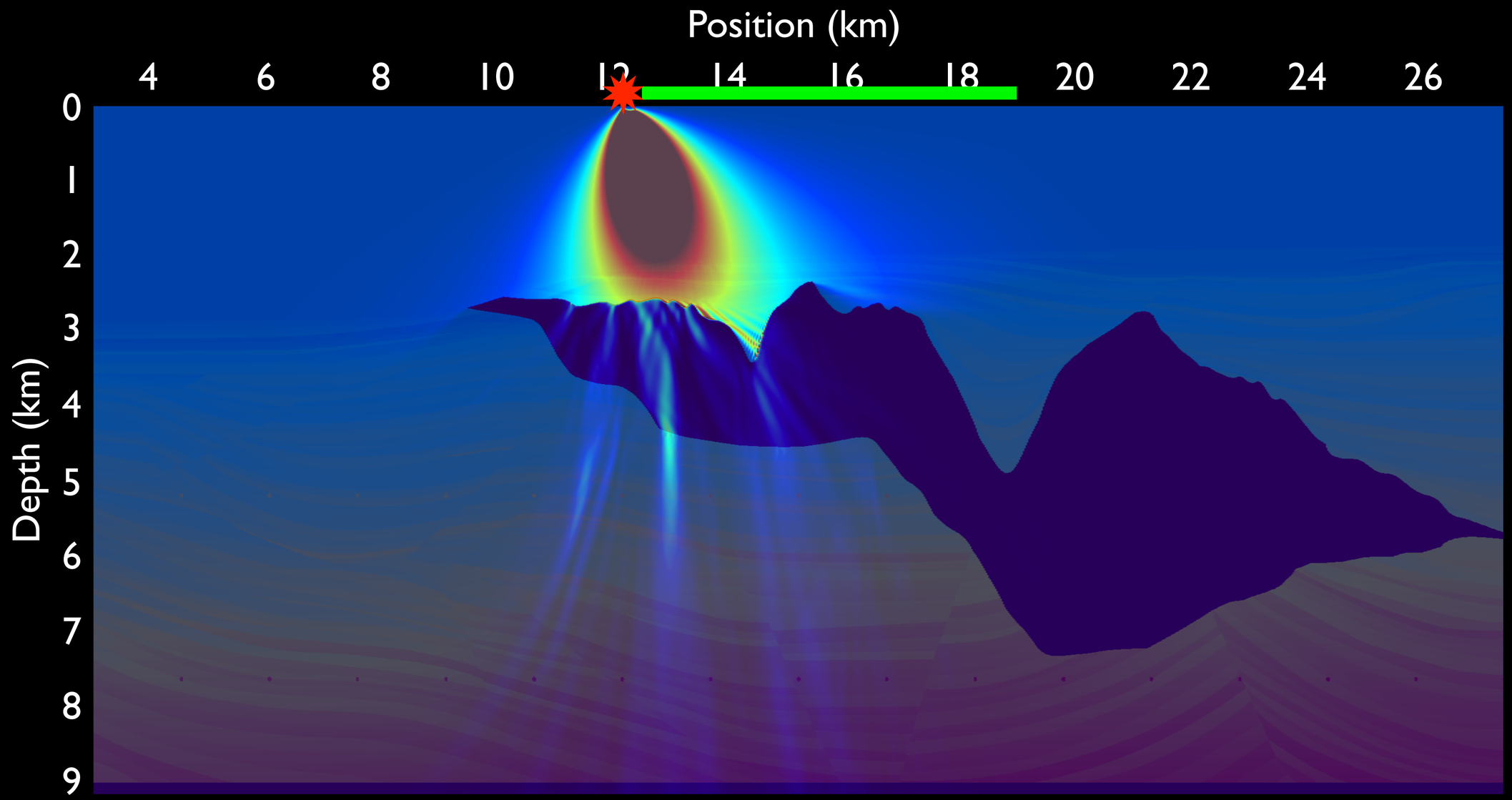
Yaxun Tang and Biondo Biondi

SEP-143, pp. 93

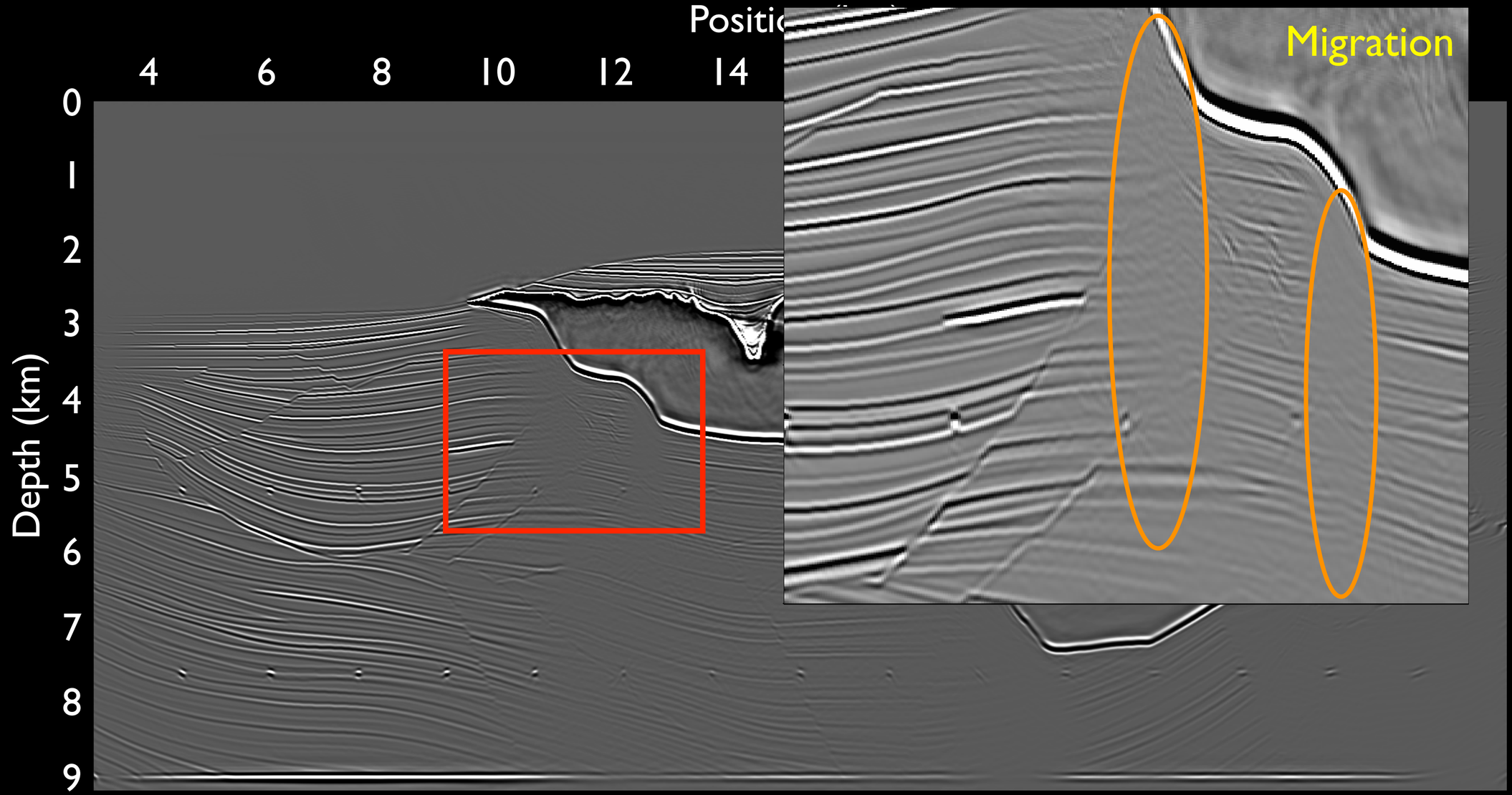
Source + complete receiver



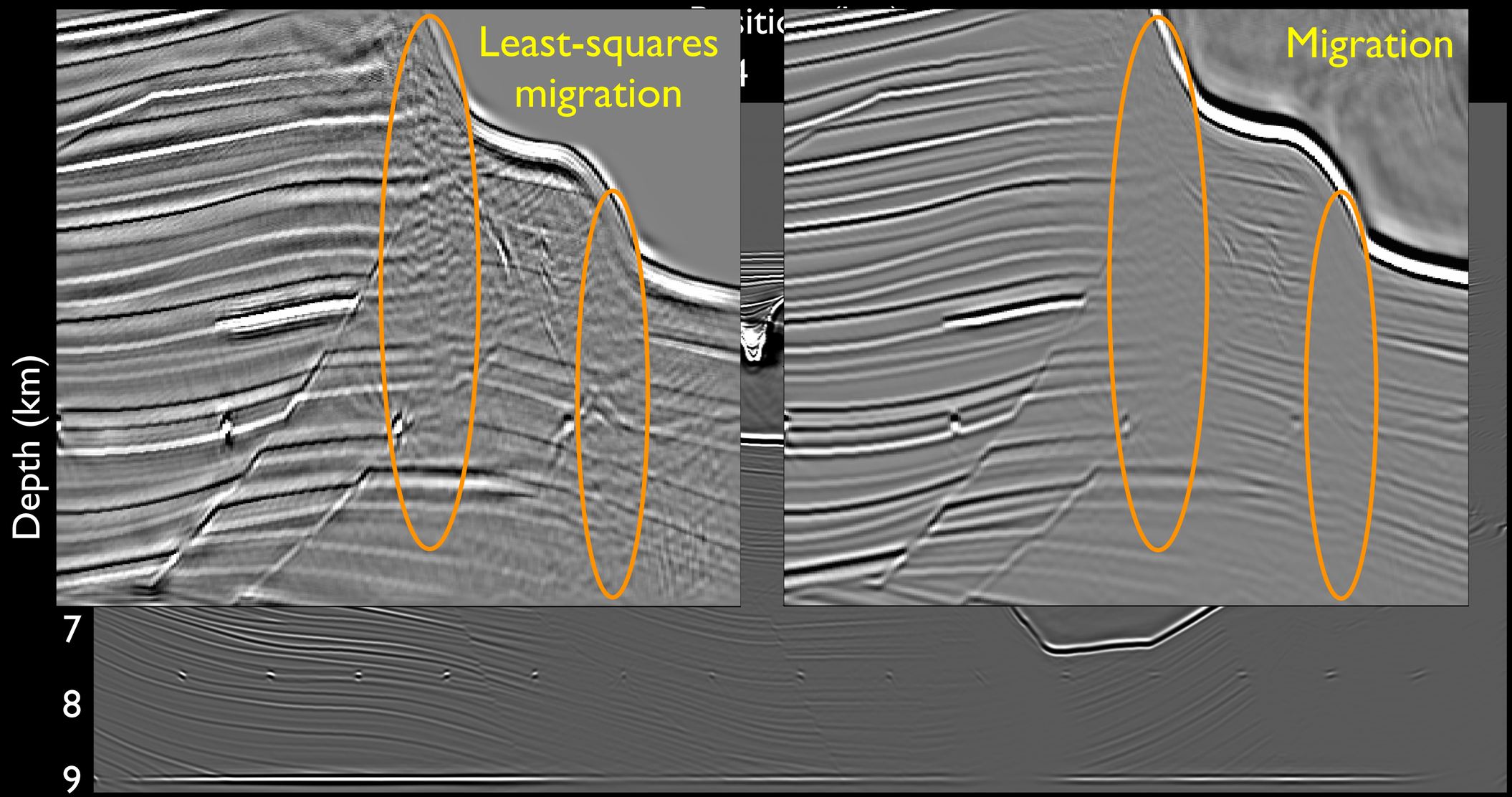
Source + limited receiver



Reflectivity image obtained by conventional method



Reflectivity image obtained by least-squares migration



Decomposing the earth model

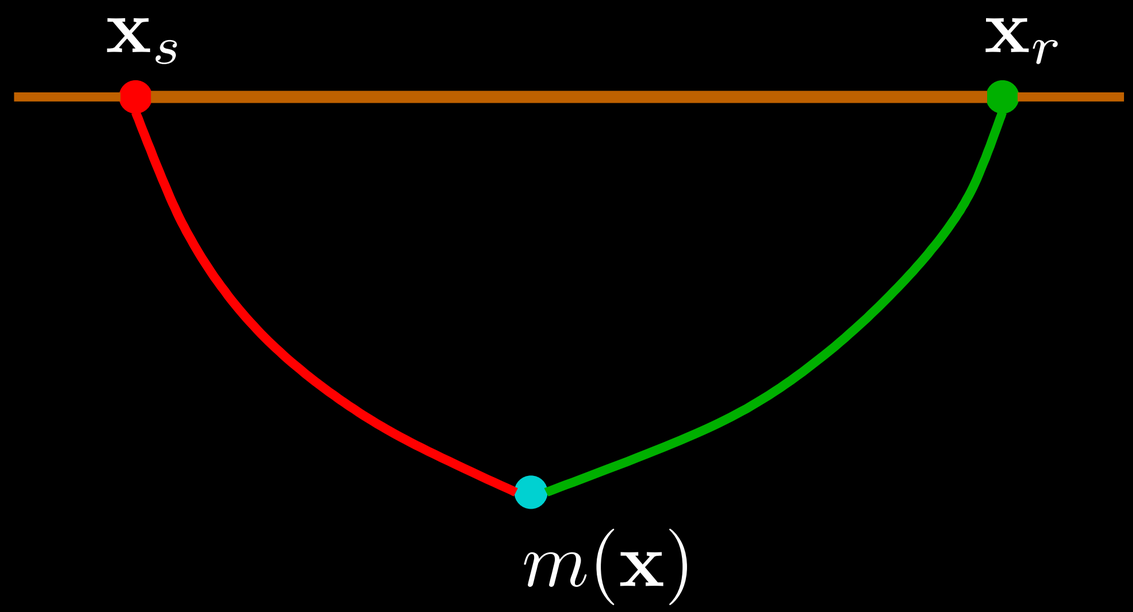
Earth model = rapidly varying component + slowly varying component

Reflectivity	Background velocity
Dynamic information	Kinematic information

The fundamental equation

To the first order, seismic data can be modeled using

$$\mathbf{d} = \mathbf{L}[\mathbf{v}]\mathbf{m}$$



$\mathbf{L}$: Born modeling operator

$\mathbf{m}$: Reflectivity

$\mathbf{d}$: Born modeled data

$\mathbf{v}$: Background velocity

Reconstruction of the reflectivity

Our recorded data (after some preprocessing) satisfy this linear equation:

$$\mathbf{d}_{\mathrm{obs}} = \mathbf{L}\mathbf{m}$$

The question becomes: Given $\mathbf{d}_{\mathrm{obs}}$ and $\mathbf{L}$, how to reconstruct $\mathbf{m}$

Ideally: $\mathbf{m} = \mathbf{L}^{-1}\mathbf{d}_{\mathrm{obs}}$

$\mathbf{L}$: Born modeling operator

$\mathbf{d}_{\mathrm{obs}}$: Observed data

$\mathbf{m}$: Reflectivity

The adjoint Born modeling operator

Migration = adjoint of the Born modeling operator: Conventional method

$$\mathbf{m}_{\mathrm{mig}} = \mathbf{L}^{*}\mathbf{d}_{\mathrm{obs}}$$

$\mathbf{L}$: Born modeling operator

$\mathbf{m}$: Reflectivity

$\mathbf{d}_{\mathrm{obs}}$: Observed data

The adjoint Born modeling operator

Migration = adjoint of the Born modeling operator: Conventional method

$$\mathbf{m}_{\text{mig}} = \mathbf{L}^* \mathbf{d}_{\text{obs}}$$

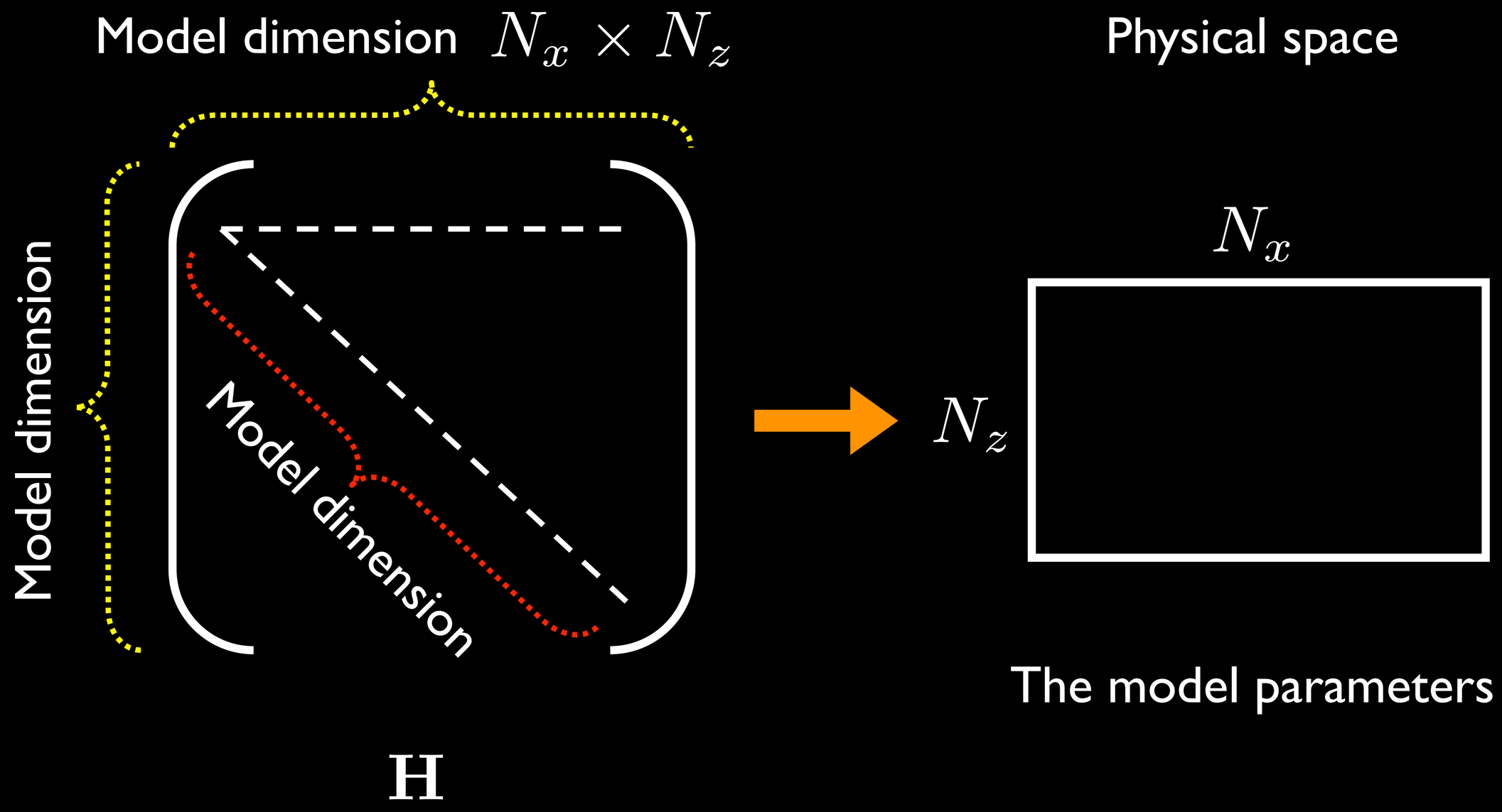
$$\mathbf{H} = \mathbf{L}^* \mathbf{L} \neq \mathbf{I}$$

$\mathbf{L}$: Born modeling operator

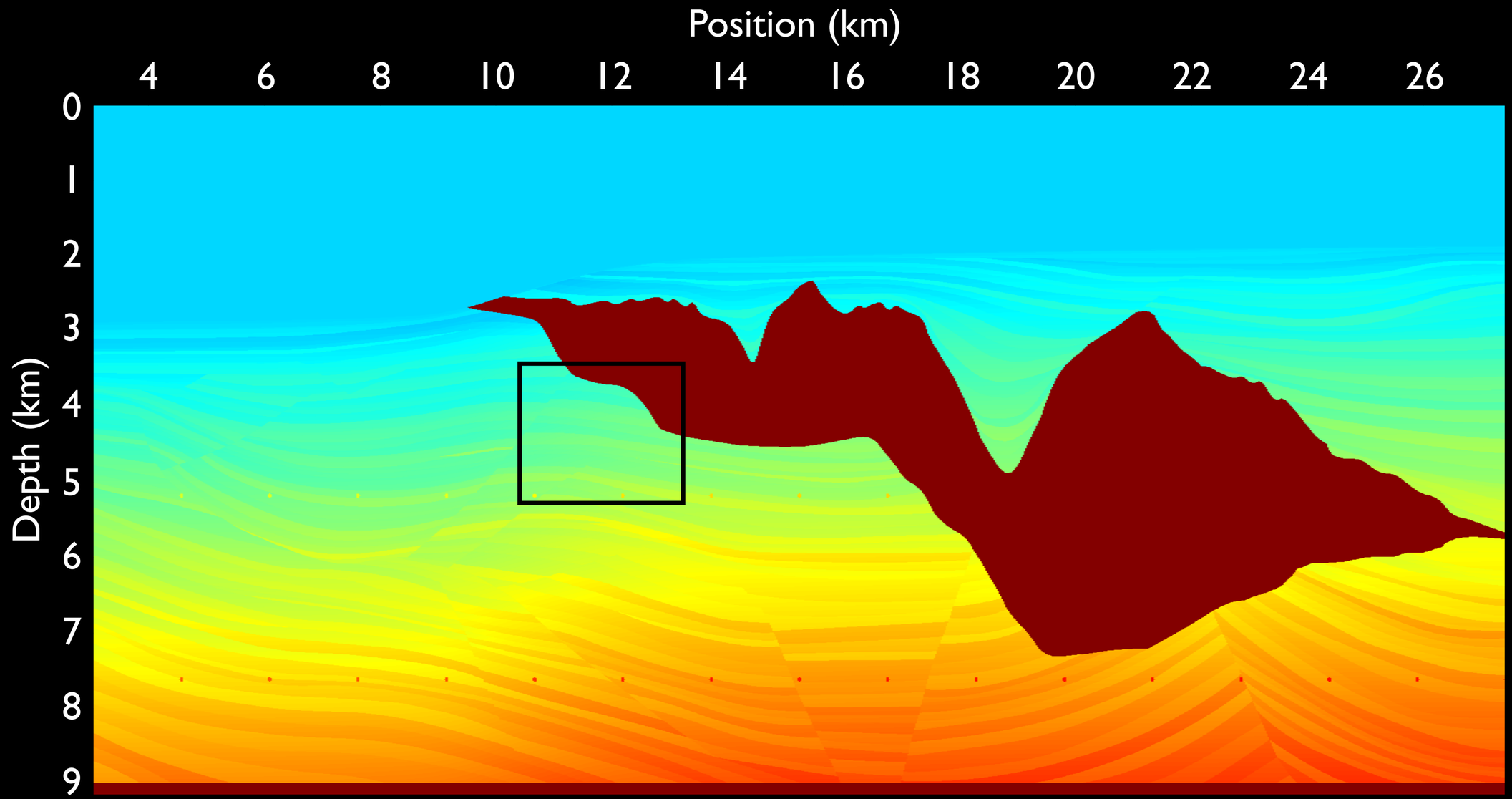
$\mathbf{d}_{\text{obs}}$: Observed data

$\mathbf{m}$: Reflectivity

The Hessian matrix

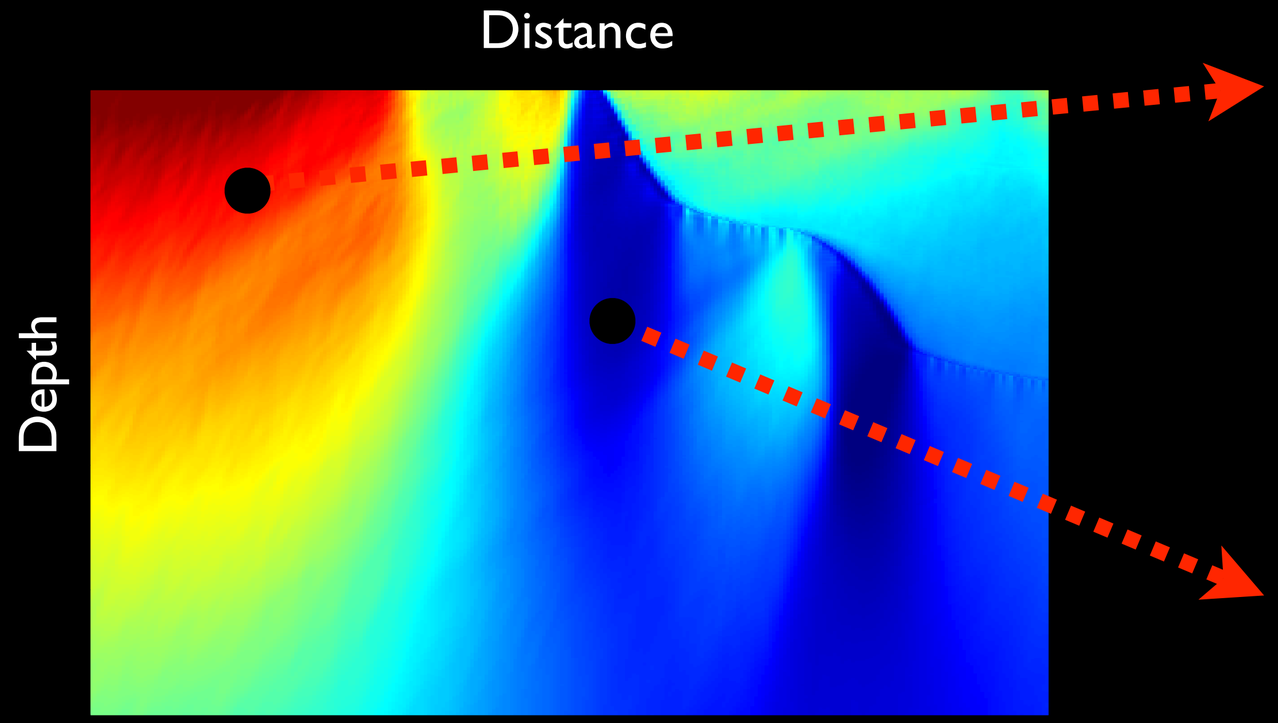


Sigsbee2A velocity model

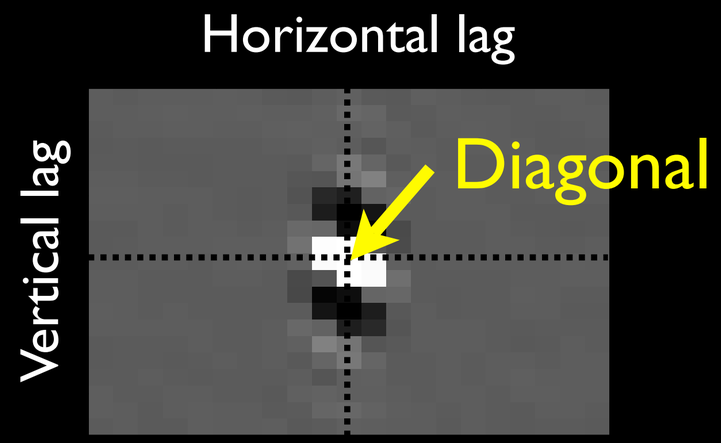


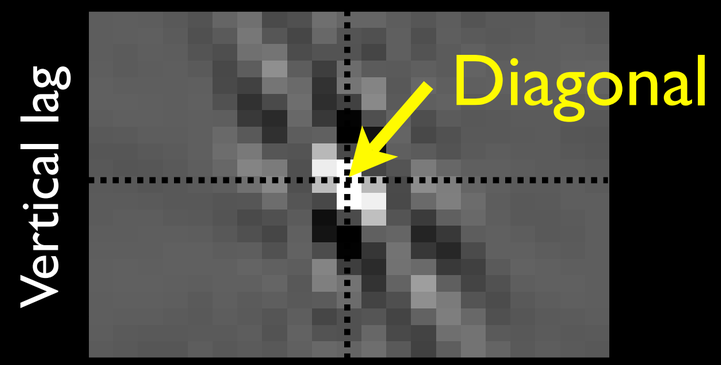
Born modeling operator is non-unitary

The normal operator (Hessian): $\mathbf{H} = \mathbf{L}^*\mathbf{L} \neq \mathbf{I}$



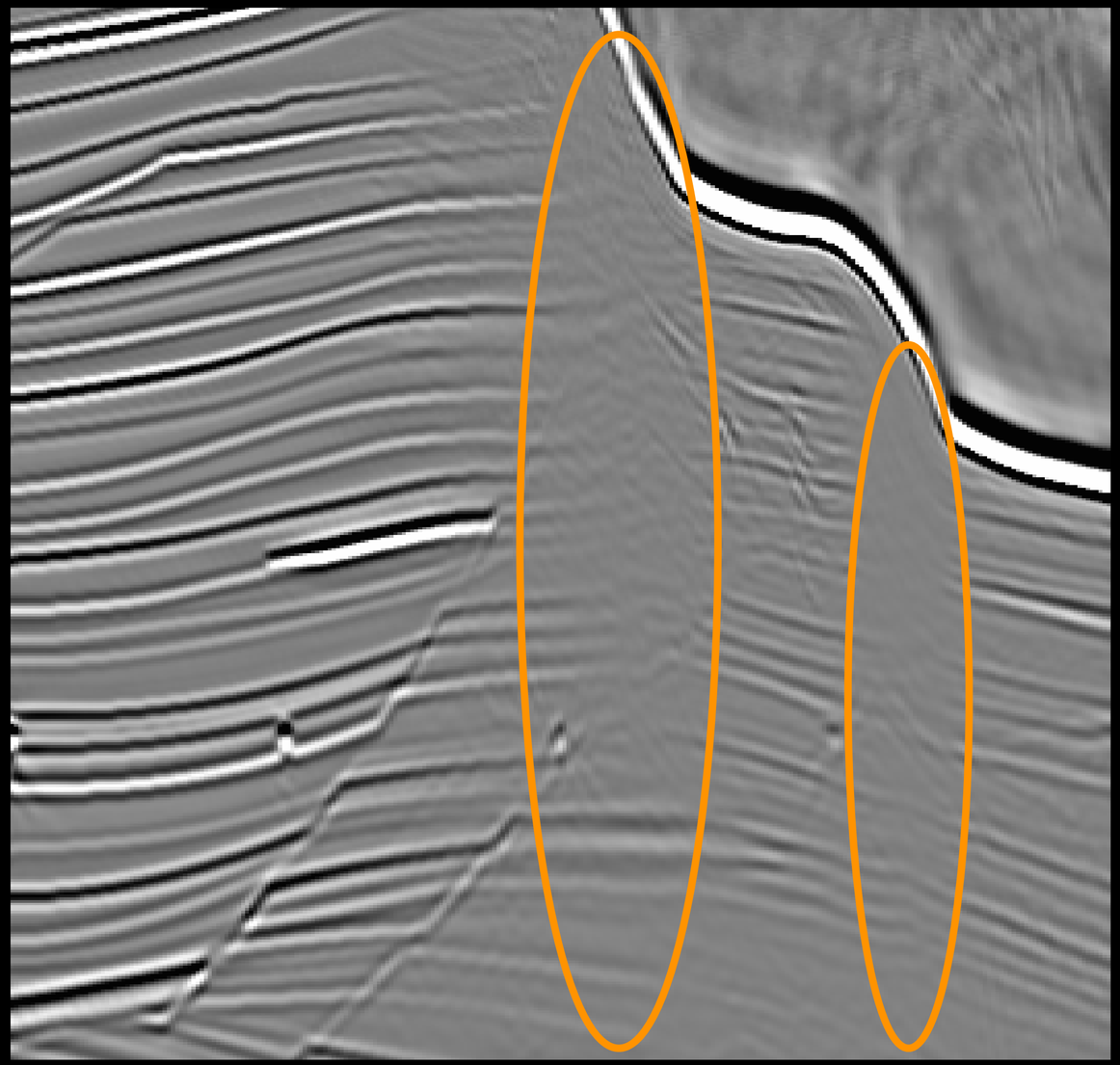
Sigsbee2A illumination map





Migration produces shadow zones

Migration



Data-domain inversion

$$F(\mathbf{m}) = \frac{1}{2}||\mathbf{Lm} - \mathbf{d}_{\text{obs}}||^2 + \mathcal{R}(\mathbf{m})$$

- Conjugate-gradient methods (M. Clapp, 2005)
- Explicit Hessian is not required
- It is costly and converges slowly for large-scale problems

Image-domain inversion

$$J(\mathbf{m}) = \frac{1}{2}||\mathbf{L}^*\mathbf{L}\mathbf{m} - \mathbf{L}^*\mathbf{d}_{\text{obs}}||^2 + \mathcal{R}(\mathbf{m})$$

Hessian

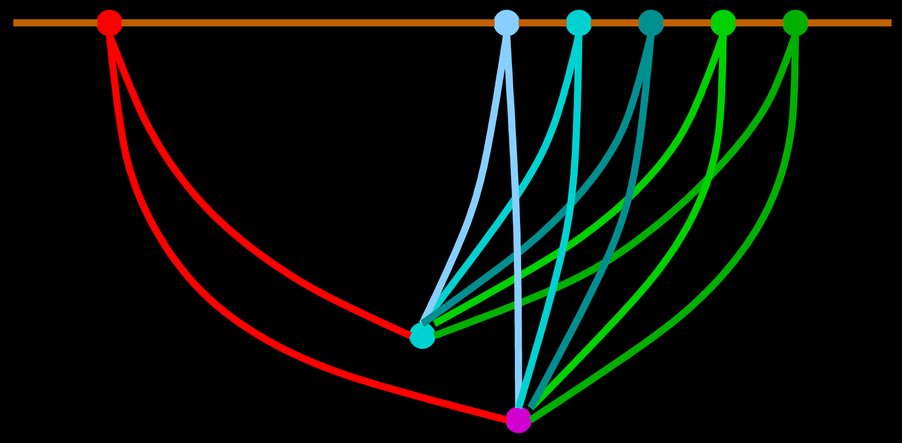
Migration image

Image-domain inversion

$$J(\mathbf{m}) = \frac{1}{2}||\mathbf{H}\mathbf{m} - \mathbf{m}_{\text{mig}}||^2 + \mathcal{R}(\mathbf{m})$$

- Target-oriented inversion (A. Valenciano, 2008)
- Different regularizations can be easily tested without extra cost
- Computing the Hessian presents a big challenge

The exact Hessian



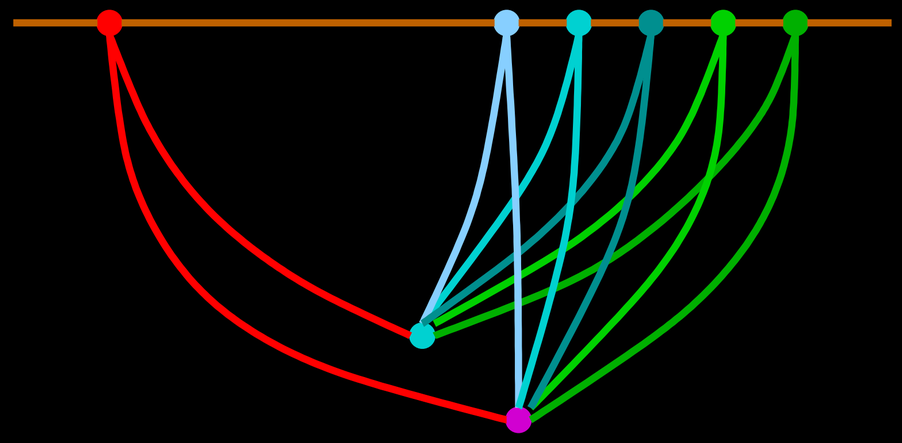
Src. Greens's function

Rec. Greens's functions

Receiver-side Green's functions are computed sequentially

$$\text{cost} \propto N_s N_r N_\omega$$

The approximate Hessian

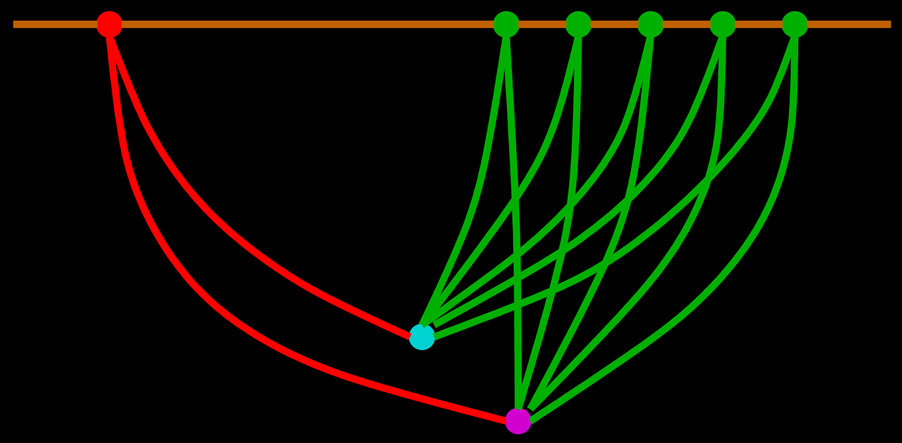


Src. Greens's function

Rec. Greens's functions

Receiver-side Green's functions are computed sequentially

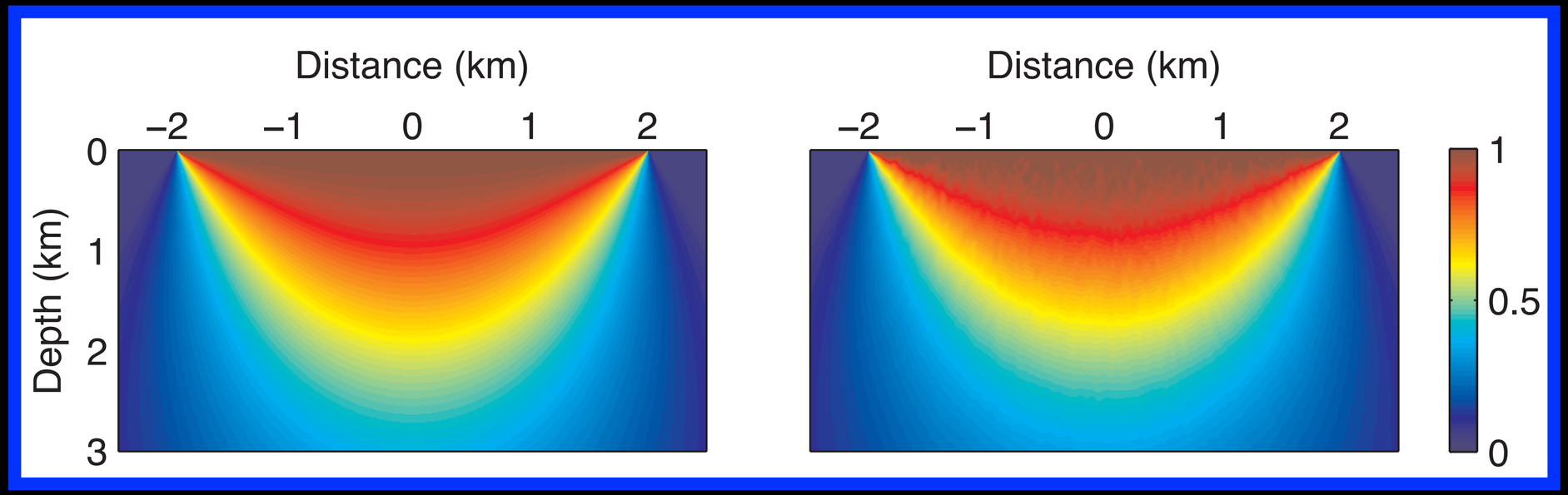
$$\text{cost} \propto N_s N_r N_\omega$$



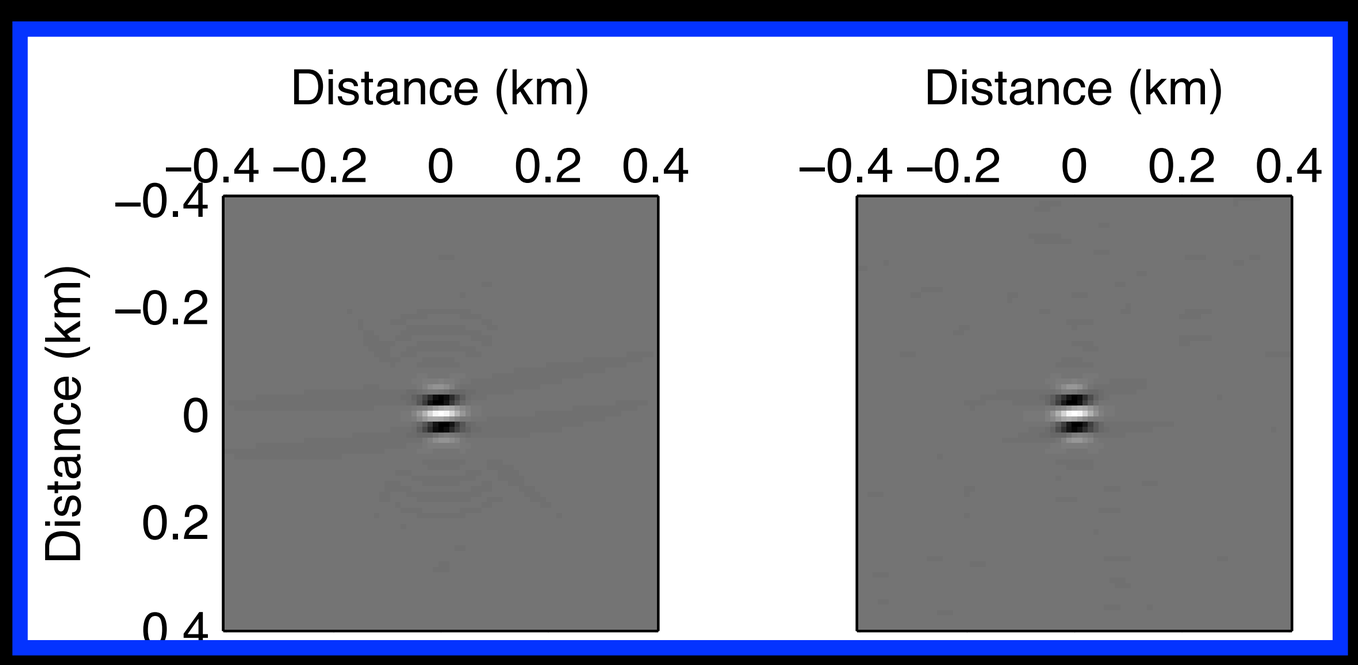
Receiver-side Green's functions are computed simultaneously

$$\text{cost} \propto N_s N_\omega$$

Hessian with 401 sources and 401 receivers



Exact Hessian

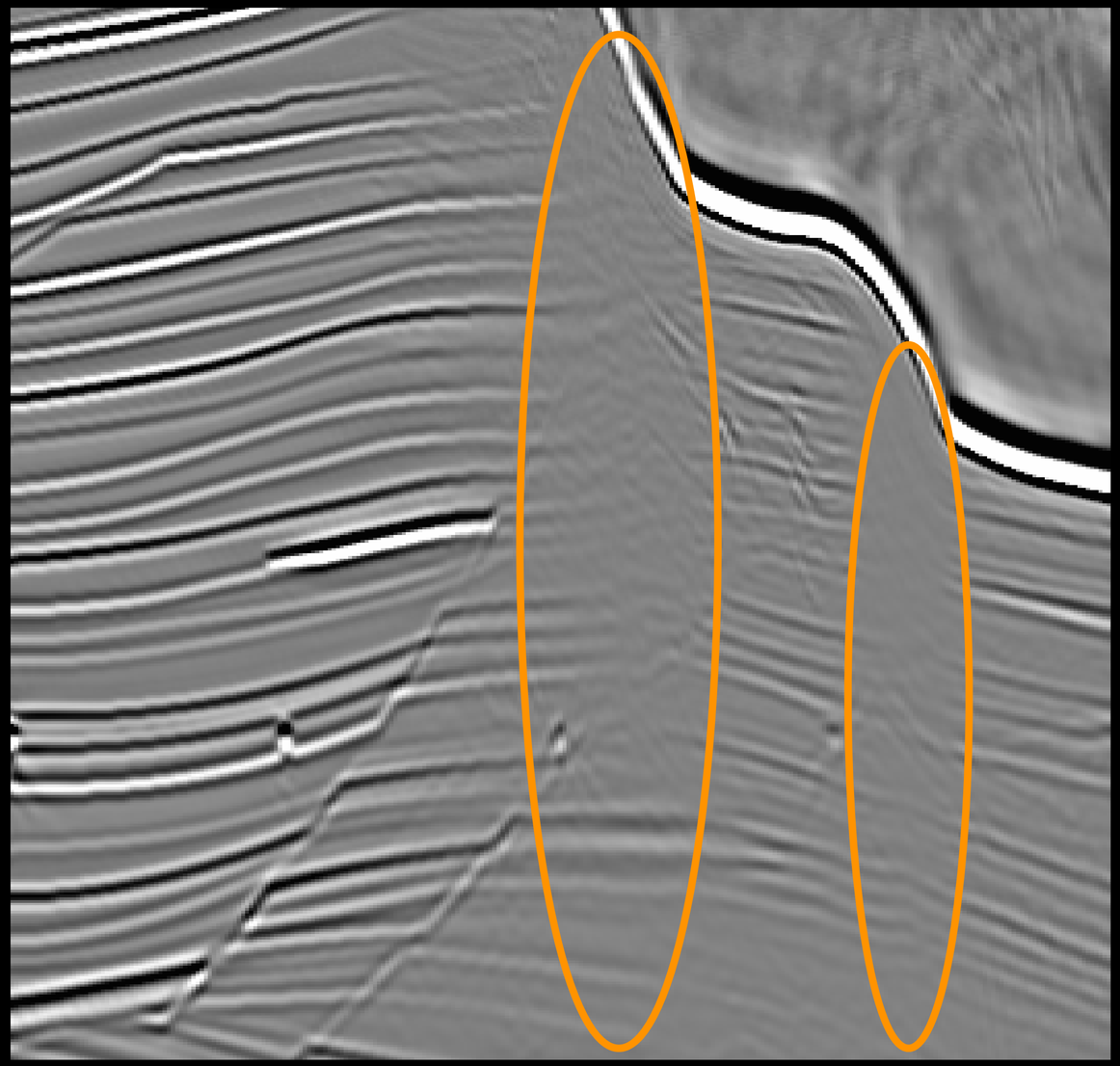


Random encoding

$N_{\text{realize}} = 1$

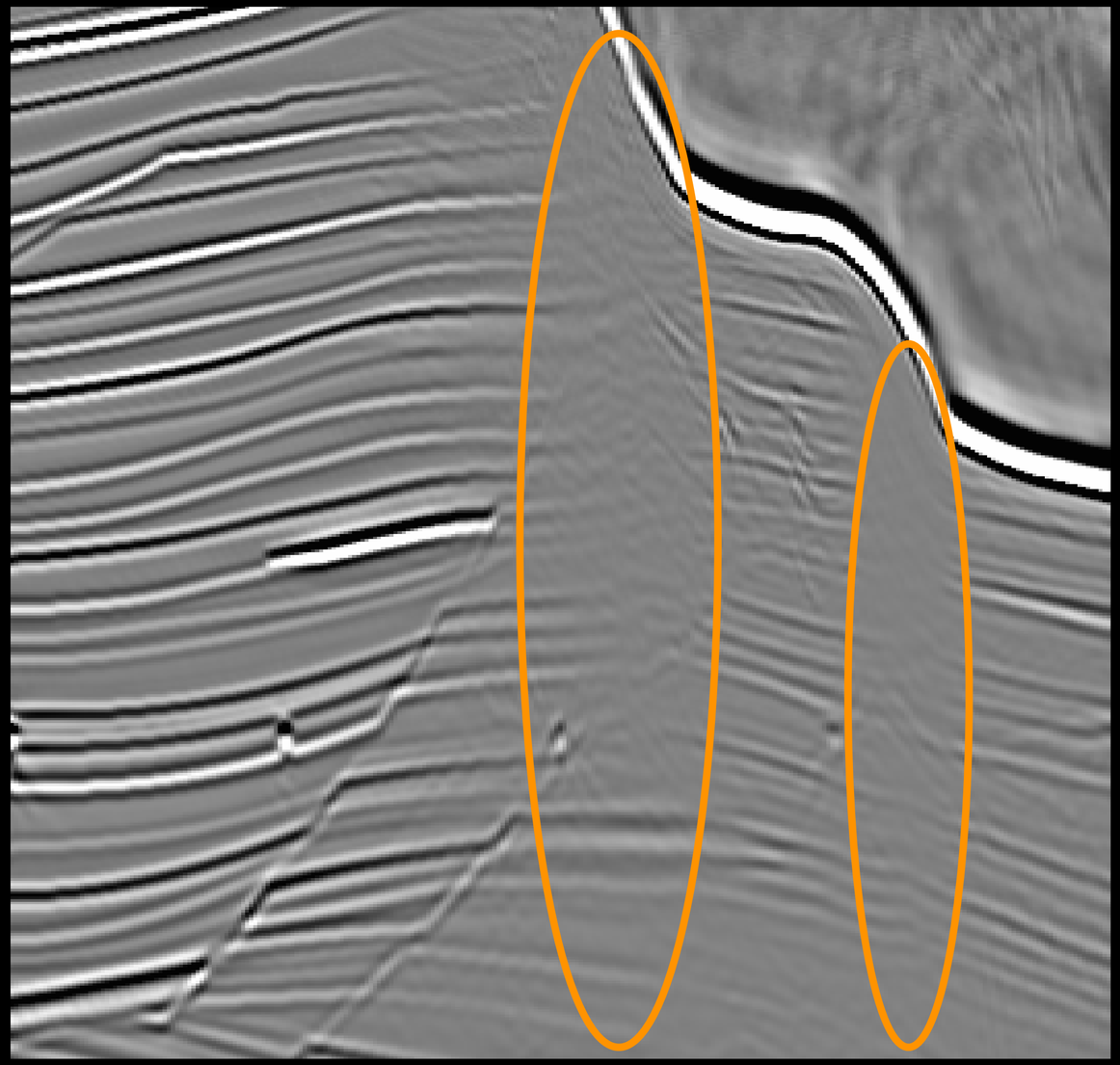
Migration produces shadow zones

Migration

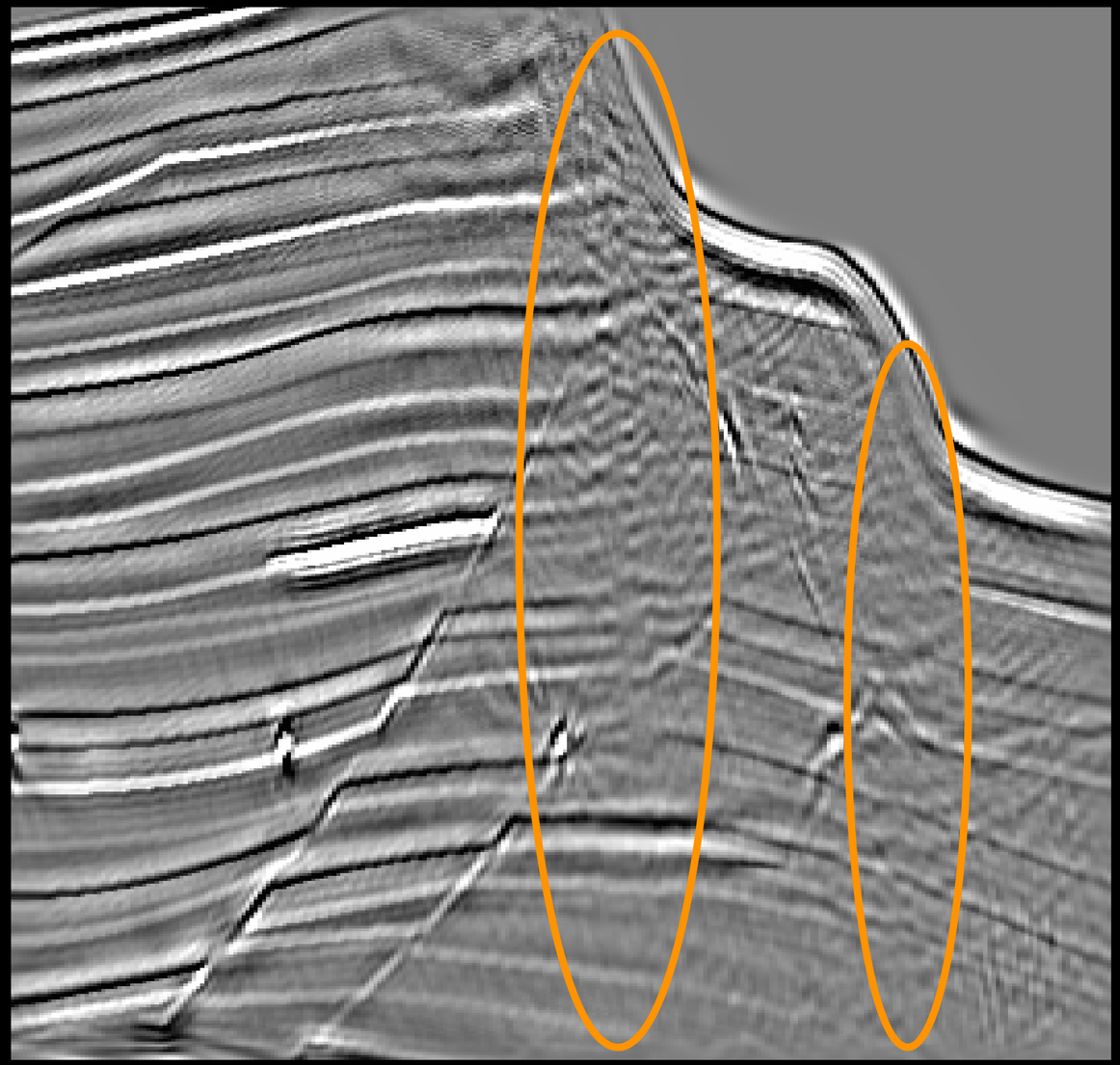


Least-squares migration fills shadow zones

Migration



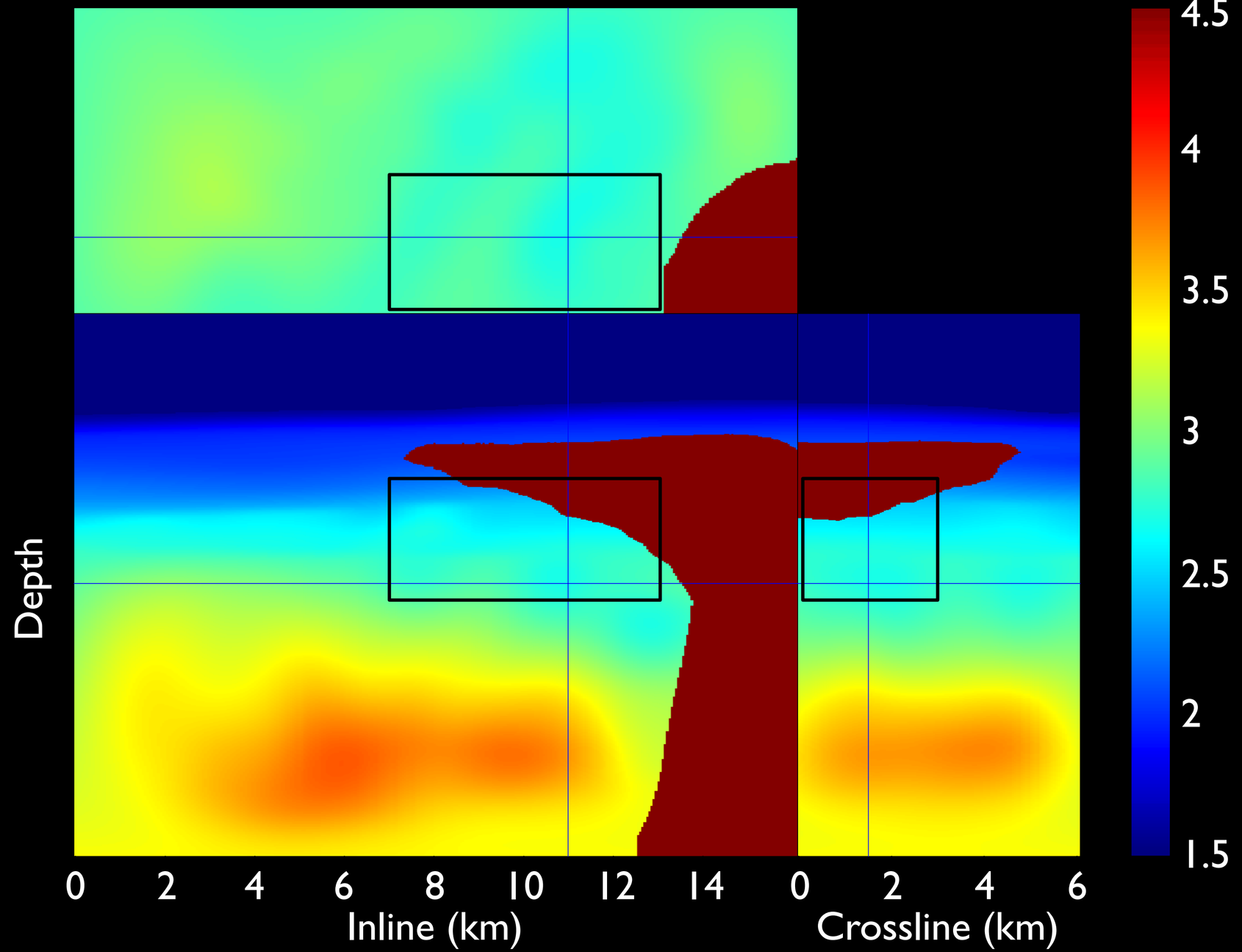
Least-squares migration



3-D Gulf of Mexico field-data example

- Narrow azimuth acquisition
- Dimensions: 20 km X 6 km
- Maximum offset: 8.2 km
- Number of shots: 100125
- Maximum frequency used for imaging: 20 Hz

Target region for reflectivity inversion



Subsalt reflectivity imaging

- **Step 1: Compute migration image**

- **Step 2: Compute the Hessian**

  - Inline plane-wave phase-encoding for sources
  - Random phase-encoding for receivers (1 realization)
  - Computed about 4 billion matrix elements
  - 3~4 days on 280 CPUs (35 nodes with 8 cores on each)

- **Step 3: Linear inversion**

  - Regularize with dip constraints to improve dip continuity
  - 6 min for 100 iterations on 34 CPUs (17 nodes with 2 cores on each)

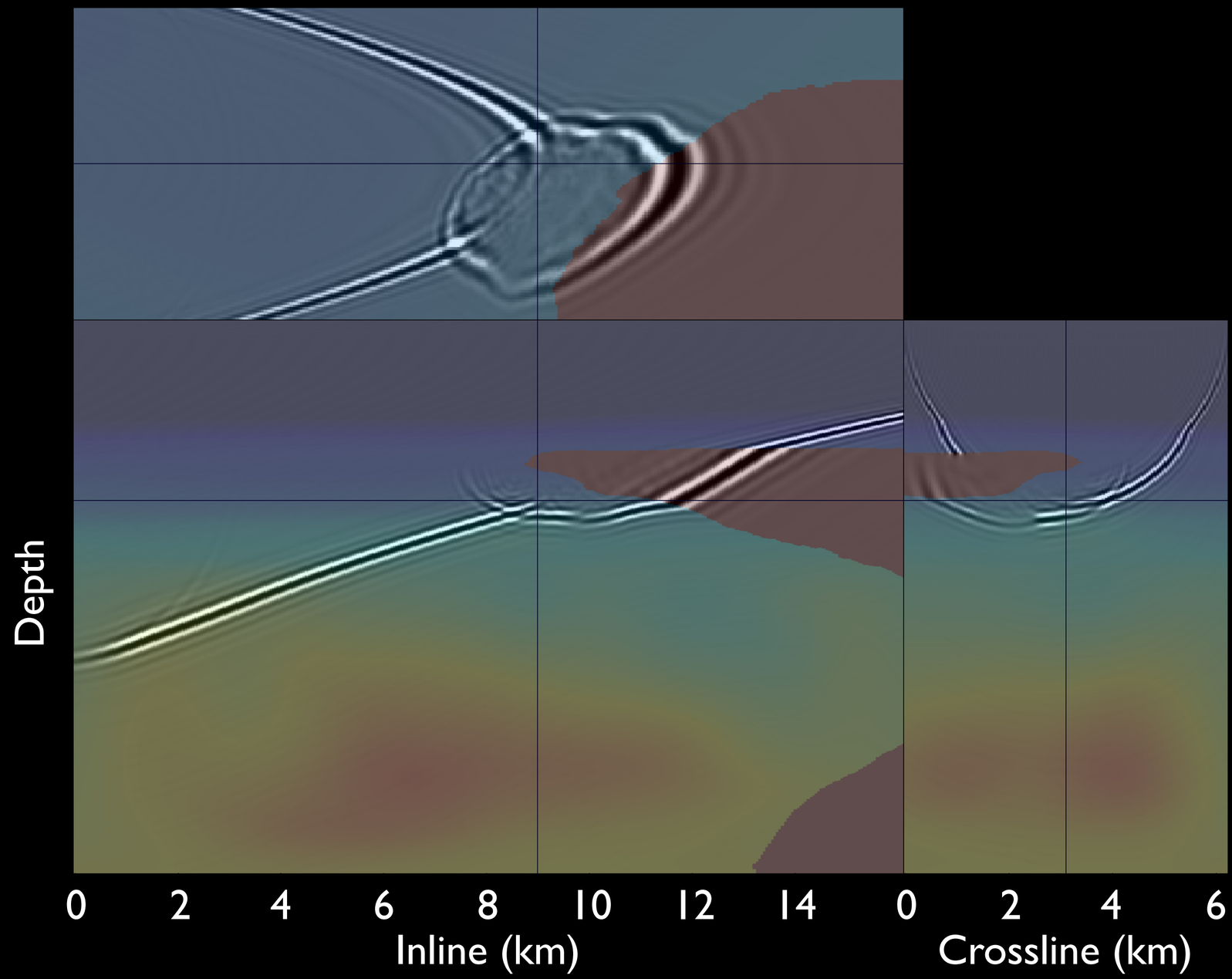
Subsalt reflectivity imaging

- **Step 1: Compute migration image**

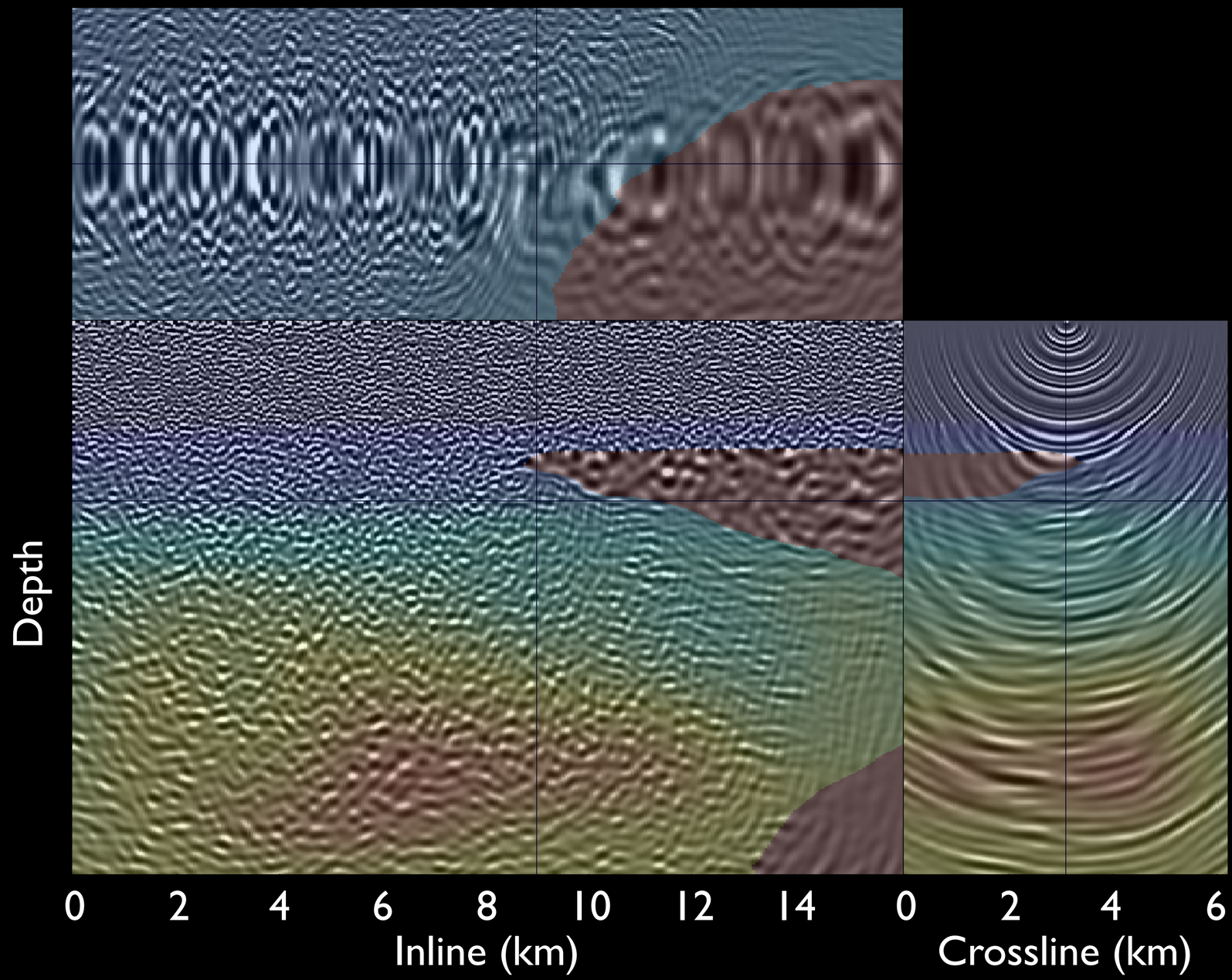
- **Step 2: Compute the Hessian**
  - Inline plane-wave phase-encoding for sources
  - Random phase-encoding for receivers (1 realization)
  - Computed about 4 billion matrix elements
  - 3~4 days on 280 CPUs (35 nodes with 8 cores on each)

- **Step 3: Linear inversion**
  - Regularize with dip constraints to improve dip continuity
  - 6 min for 100 iterations on 34 CPUs (17 nodes with 2 cores on each)

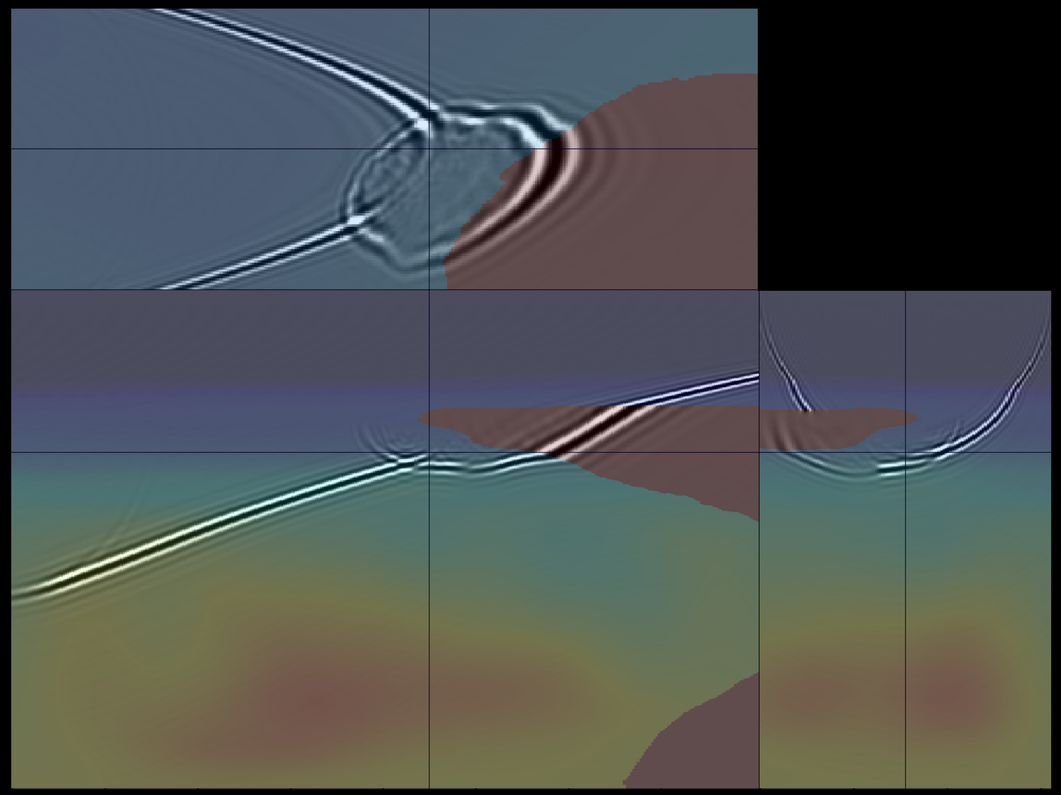
Snapshot of encoded source-side Green's function



Snapshot of encoded receiver-side Green's function

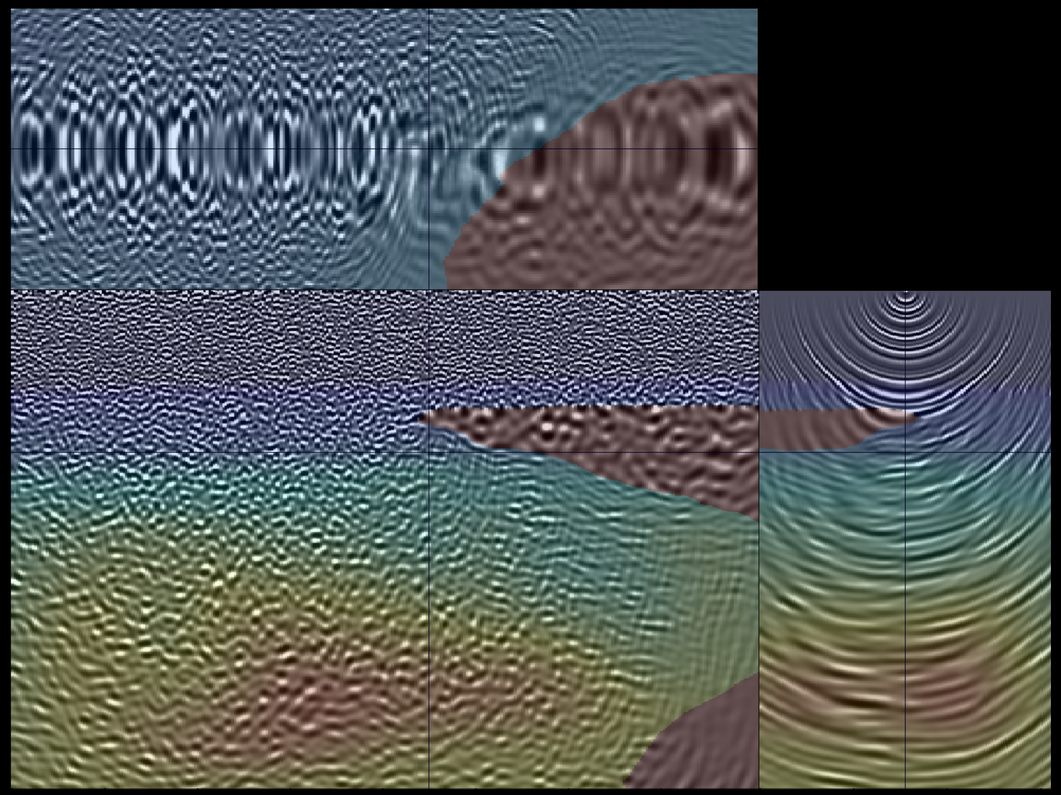


Correlate source-side Green's function



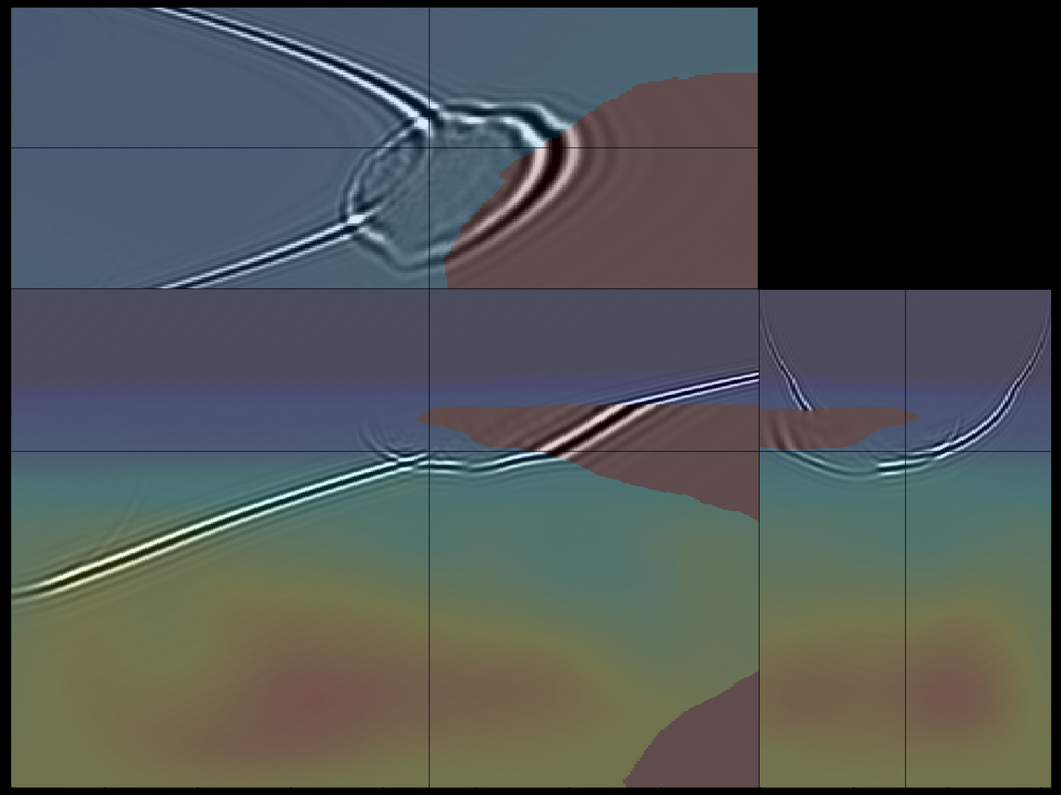
$$S(\mathbf{x},\omega)S^*(\mathbf{y},\omega)$$

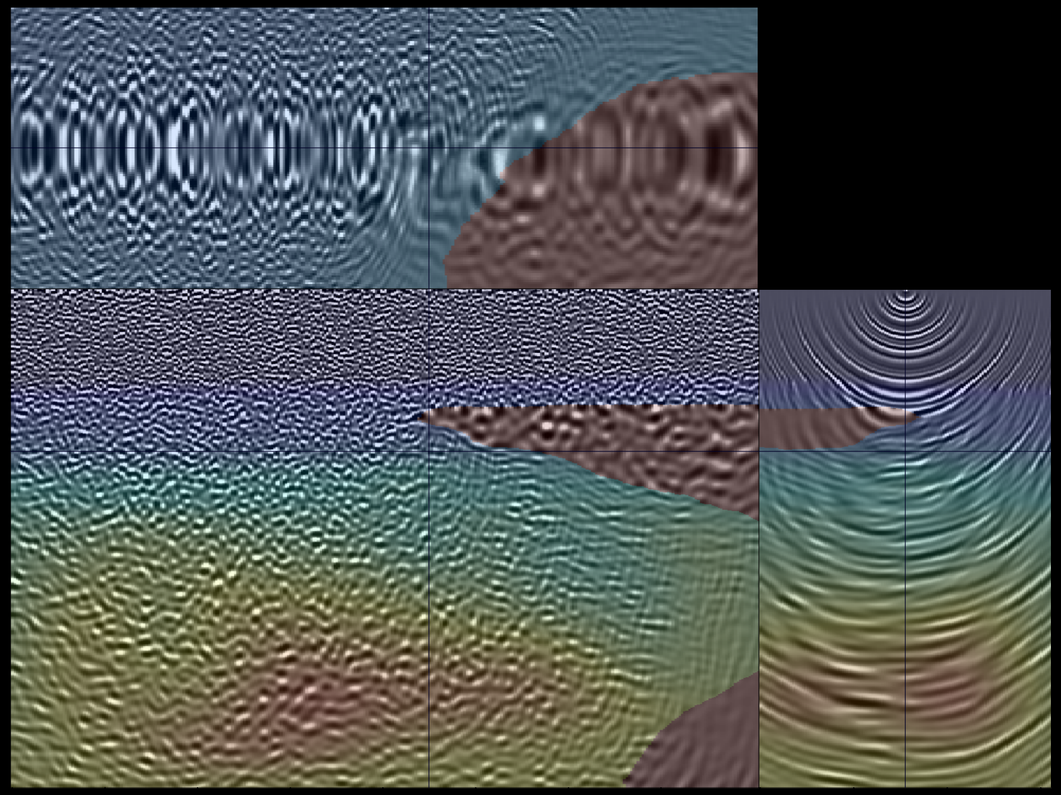
Correlate receiver-side Green's function



$$R(\mathbf{x},\omega)R^*(\mathbf{y},\omega)$$

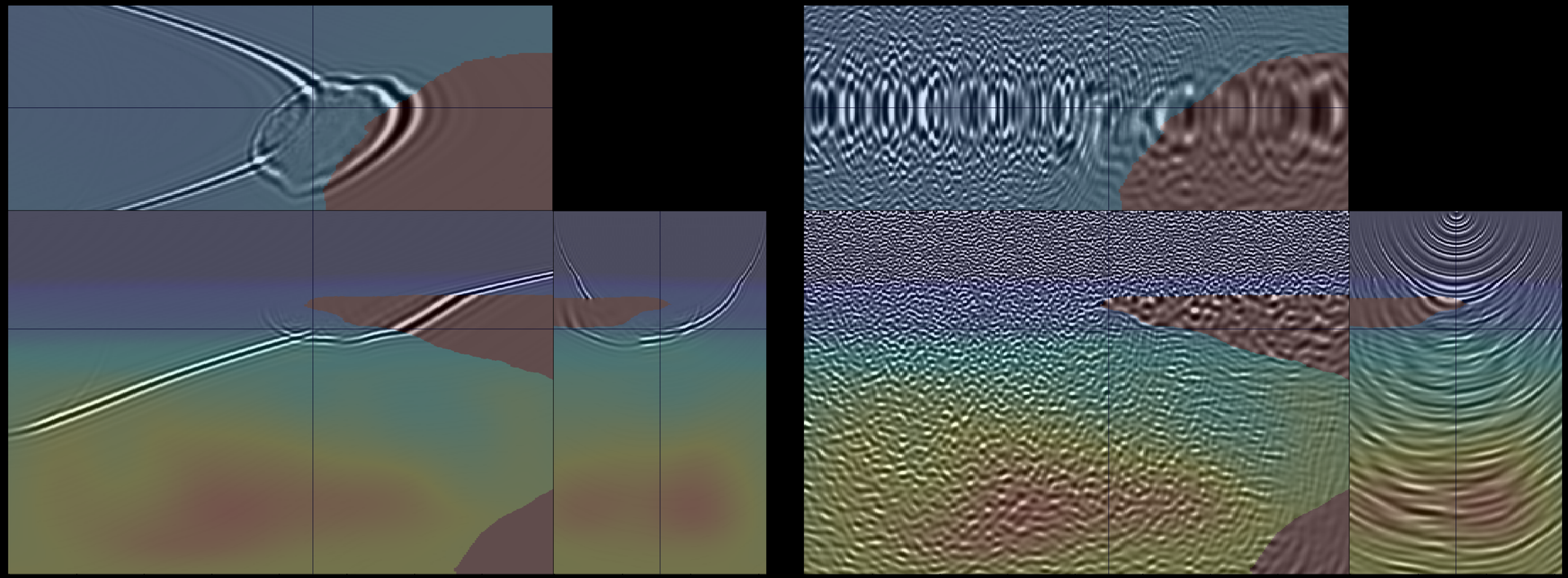
Phase-encoded Hessian for one conical wave





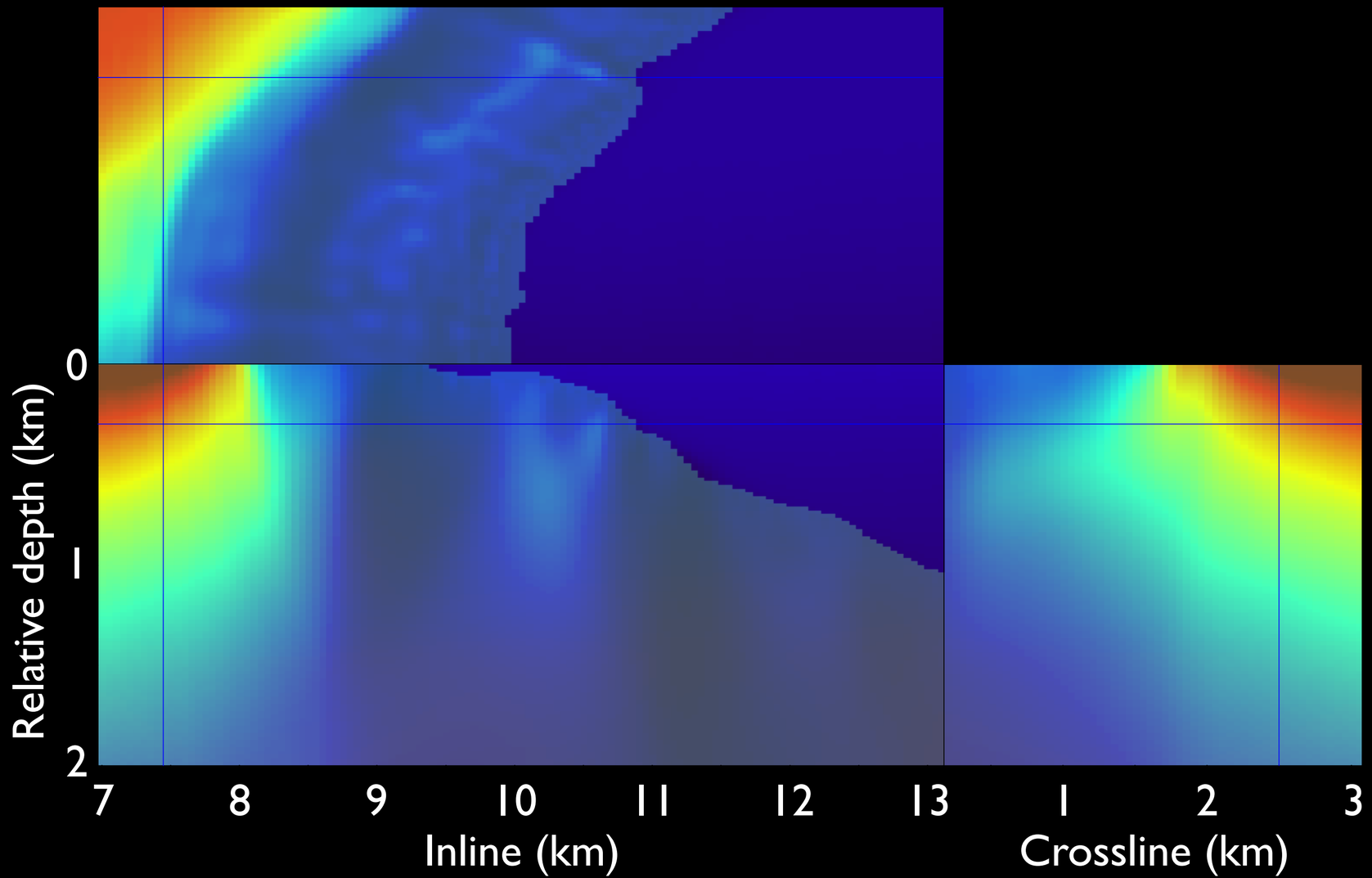
$$H(\mathbf{x},\mathbf{y}) = S(\mathbf{x},\omega)S^*(\mathbf{y},\omega)R(\mathbf{x},\omega)R^*(\mathbf{y},\omega)$$

Sum over all contributions

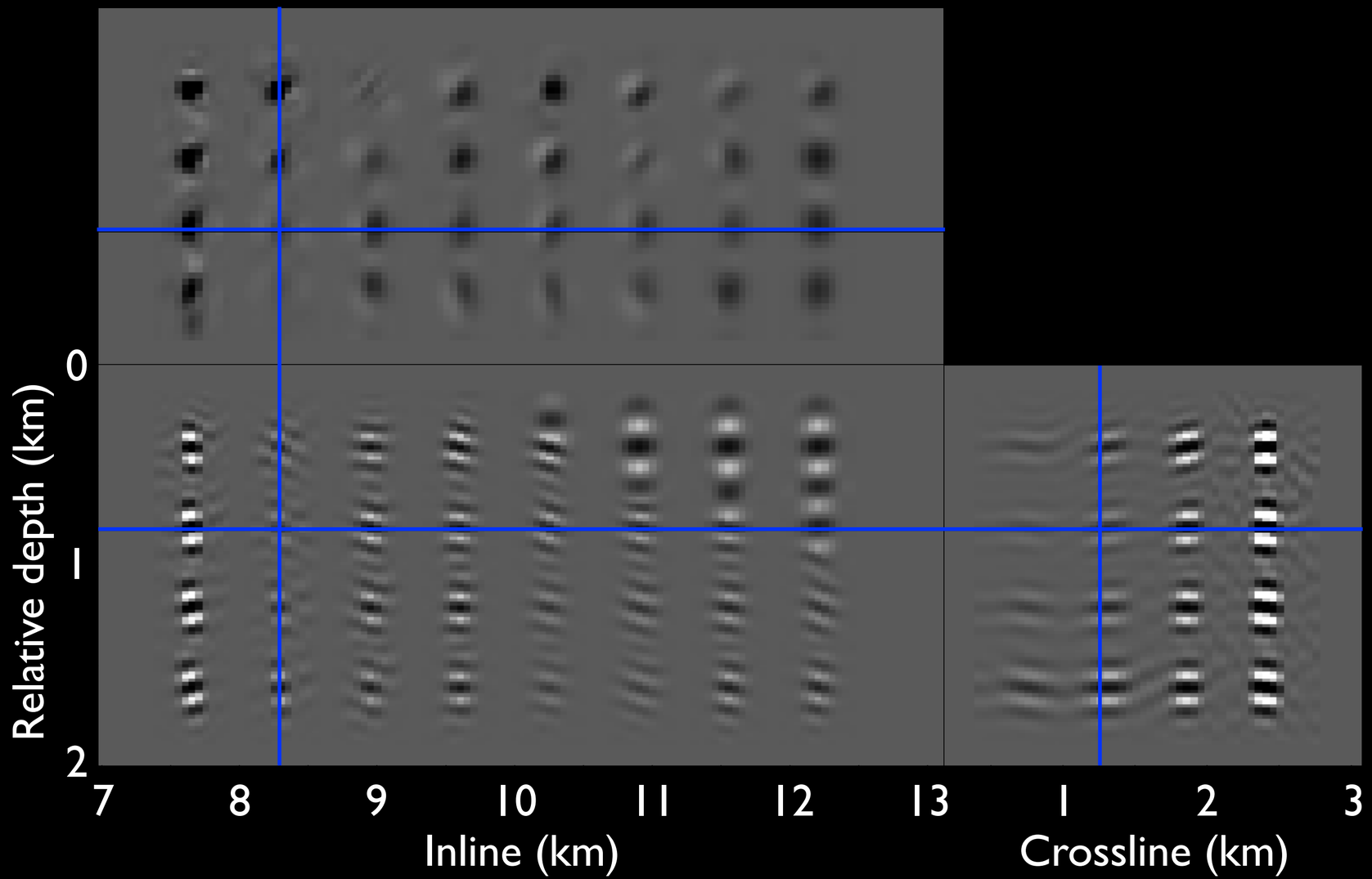


$$H(\mathbf{x},\mathbf{y}) = \sum_{\text{xline}} \sum_{\omega} \sum_{\text{plane}} S(\mathbf{x},\omega) S^*(\mathbf{y},\omega) R(\mathbf{x},\omega) R^*(\mathbf{y},\omega)$$

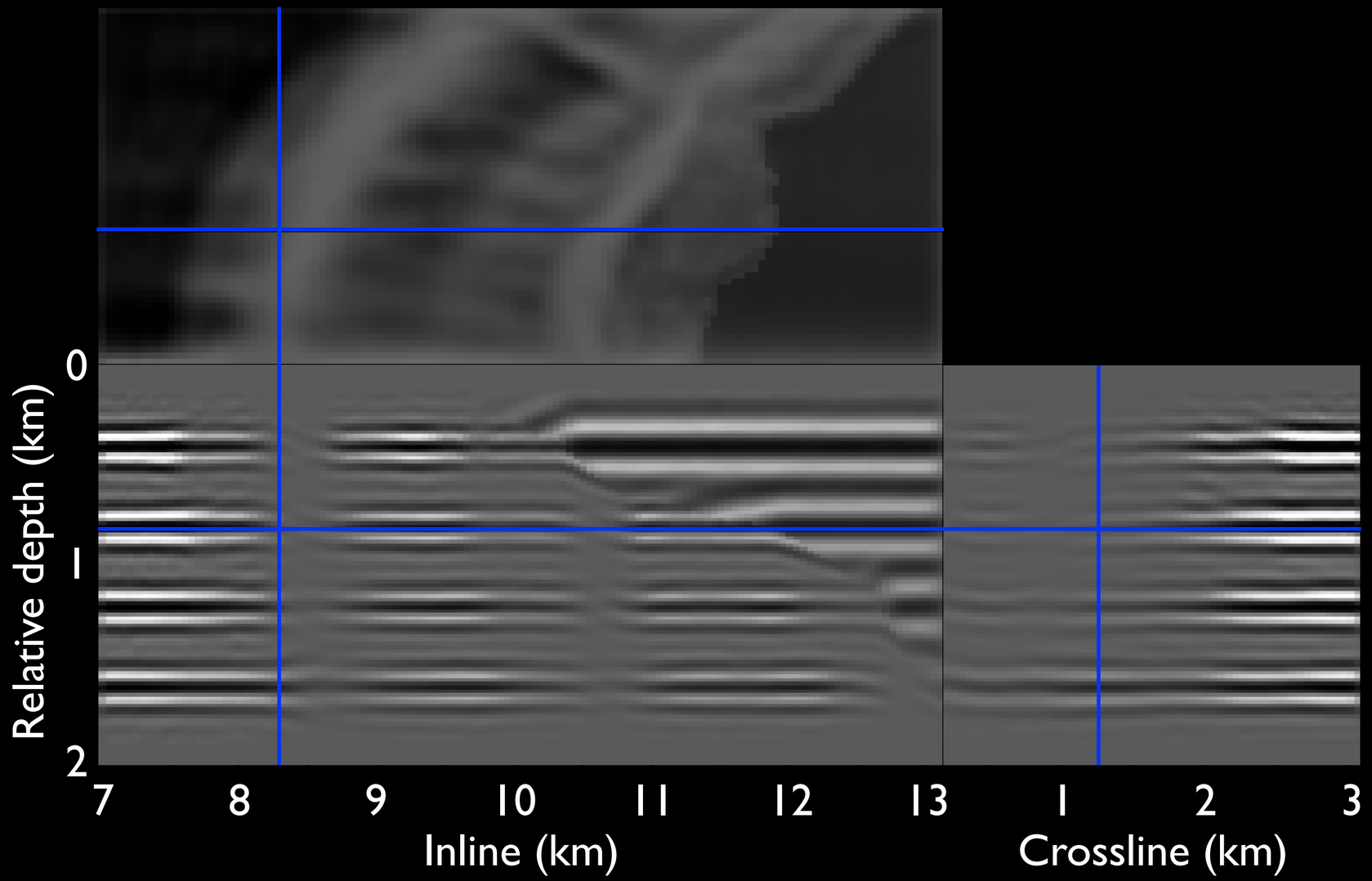
The diagonal of the 3-D phase-encoded Hessian



Hessian response to point scatterers



Hessian response to flat reflectors



Subsalt reflectivity imaging

- **Step 1: Compute migration image**

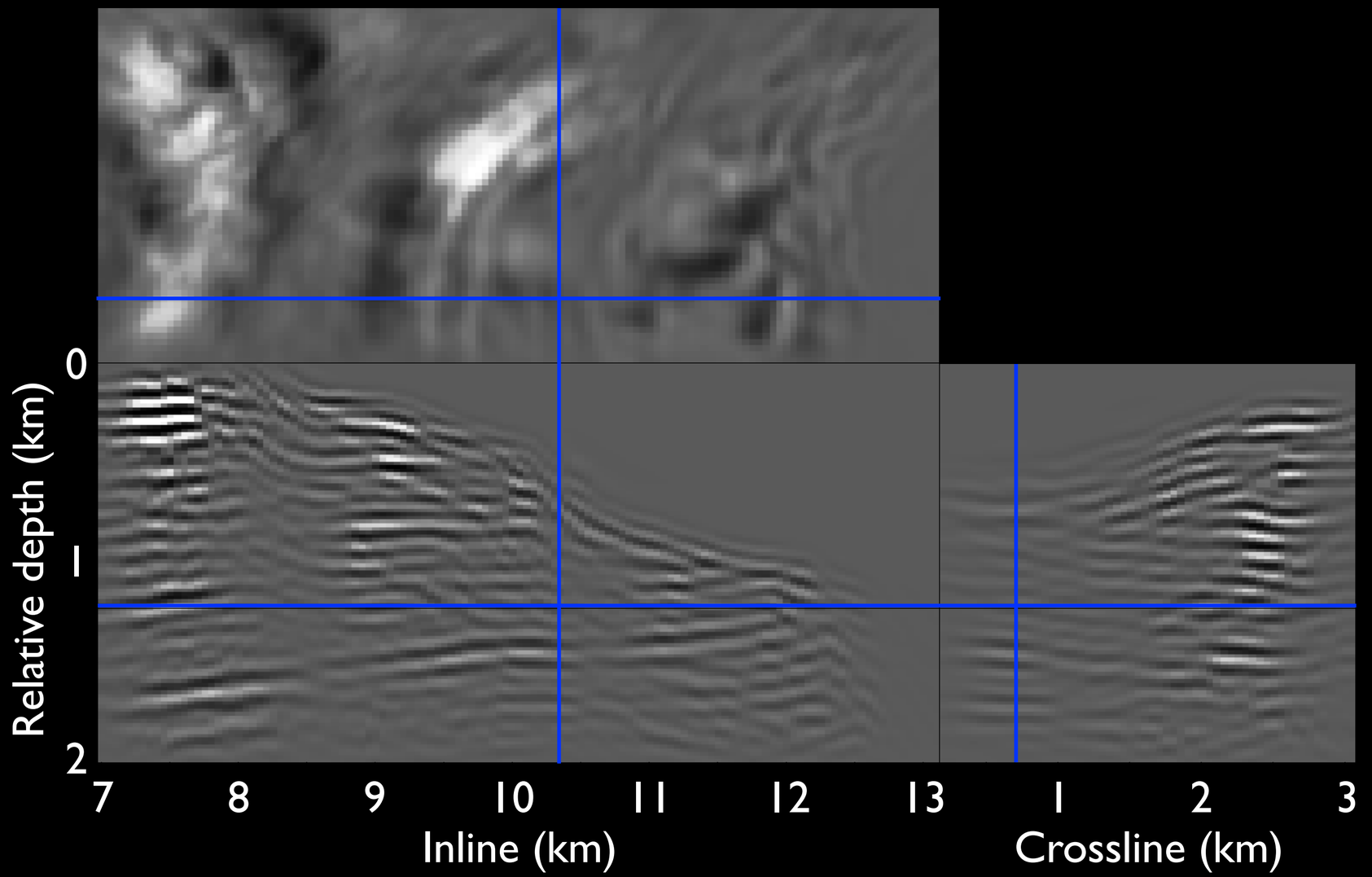
- **Step 2: Compute the Hessian**

  - Inline plane-wave phase-encoding for sources
  - Random phase-encoding for receivers (1 realization)
  - Computed about 4 billion matrix elements
  - 3~4 days on 280 CPUs (35 nodes with 8 cores on each)

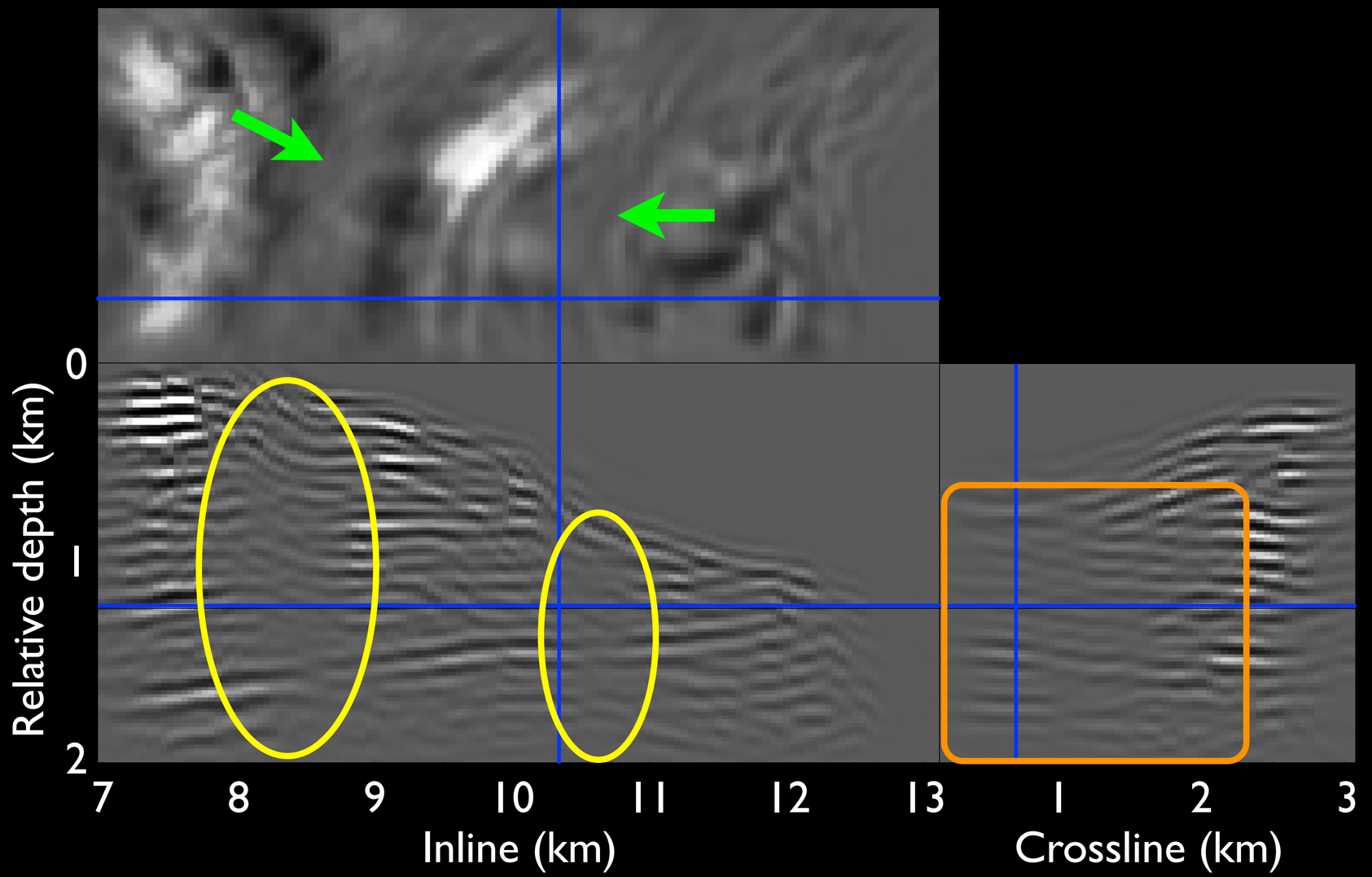
- **Step 3: Linear inversion**

  - Regularize with dip constraints to improve dip continuity
  - 6 min for 100 iterations on 34 CPUs (17 nodes with 2 cores on each)

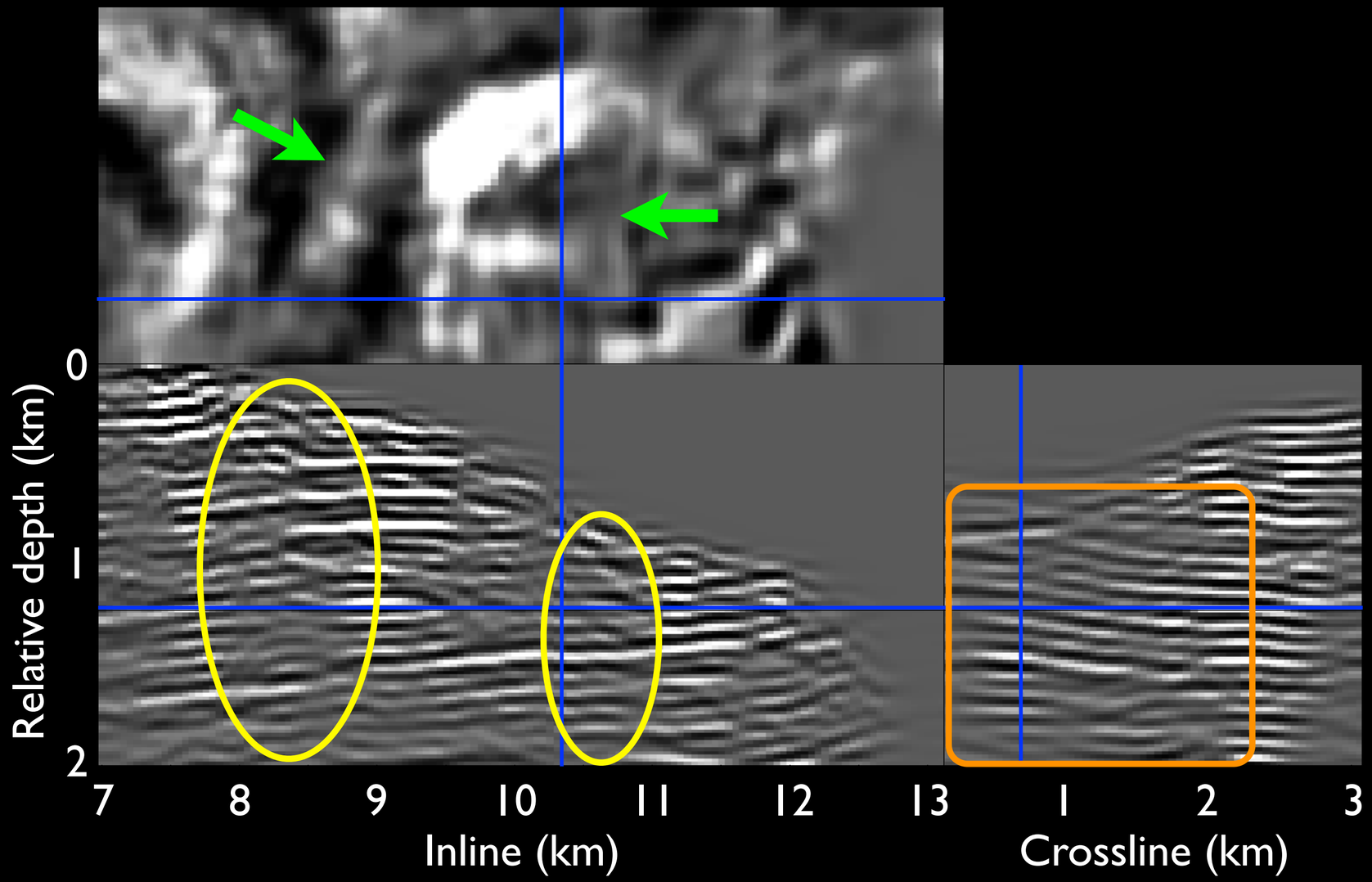
Migration


0
1
2
Relative depth (km)
7
8
9
10
11
12
13
Inline (km)
1
2
3
Crossline (km)

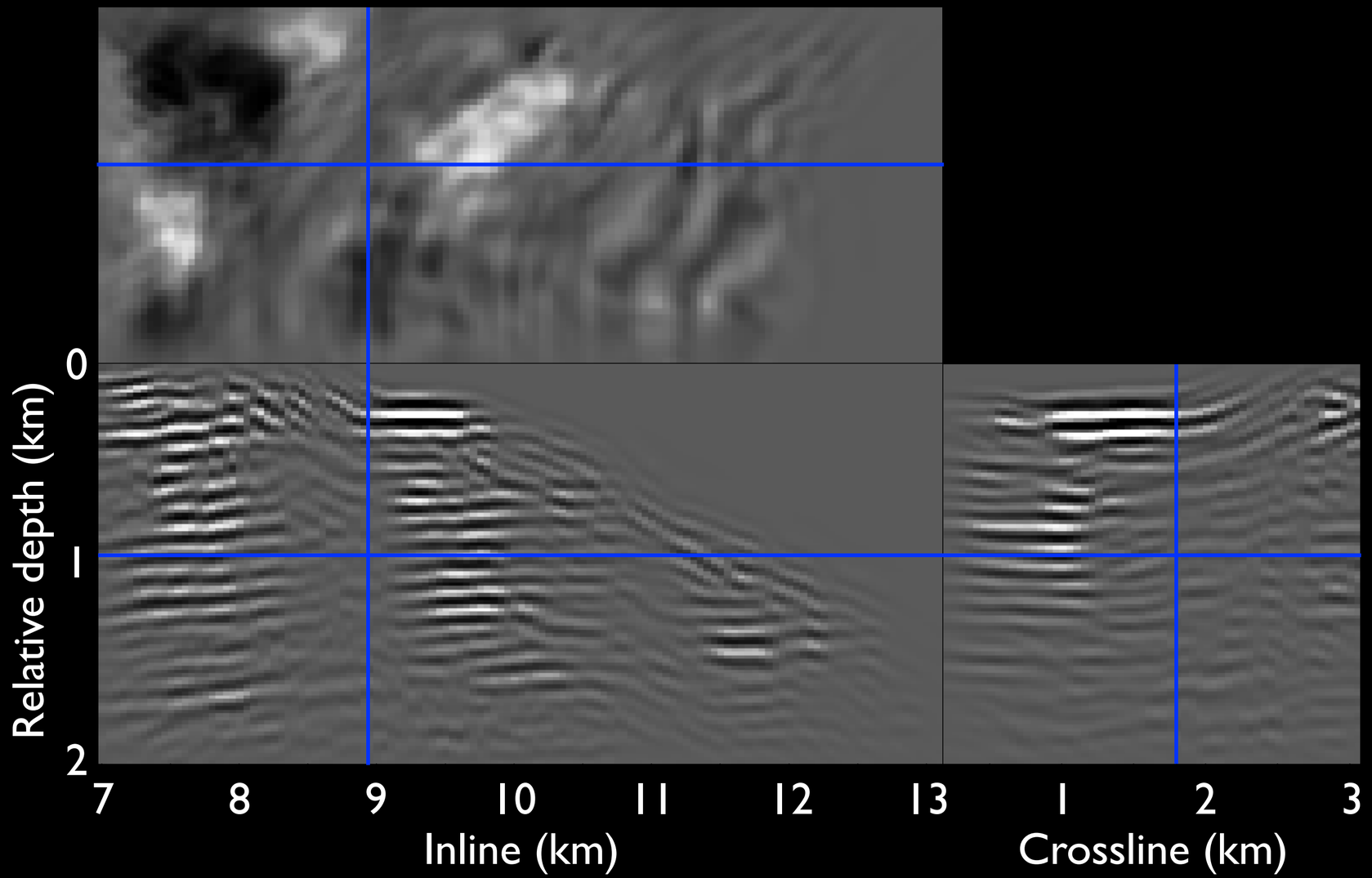
Migration


0
1
2
Relative depth (km)
7
8
9
10
11
12
13
1
2
3
Inline (km)
Crossline (km)

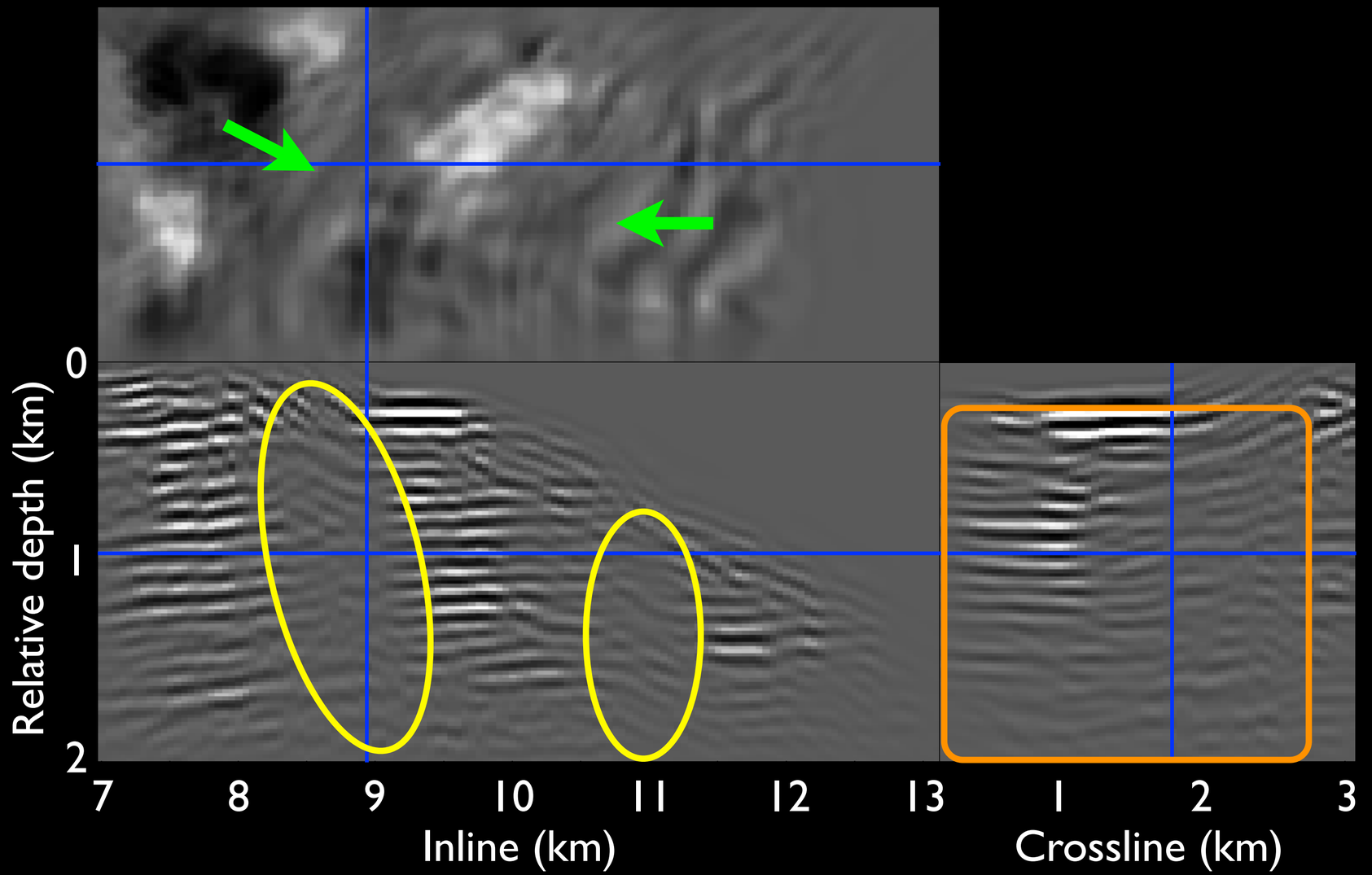
Least-squares migration (linearized inversion)



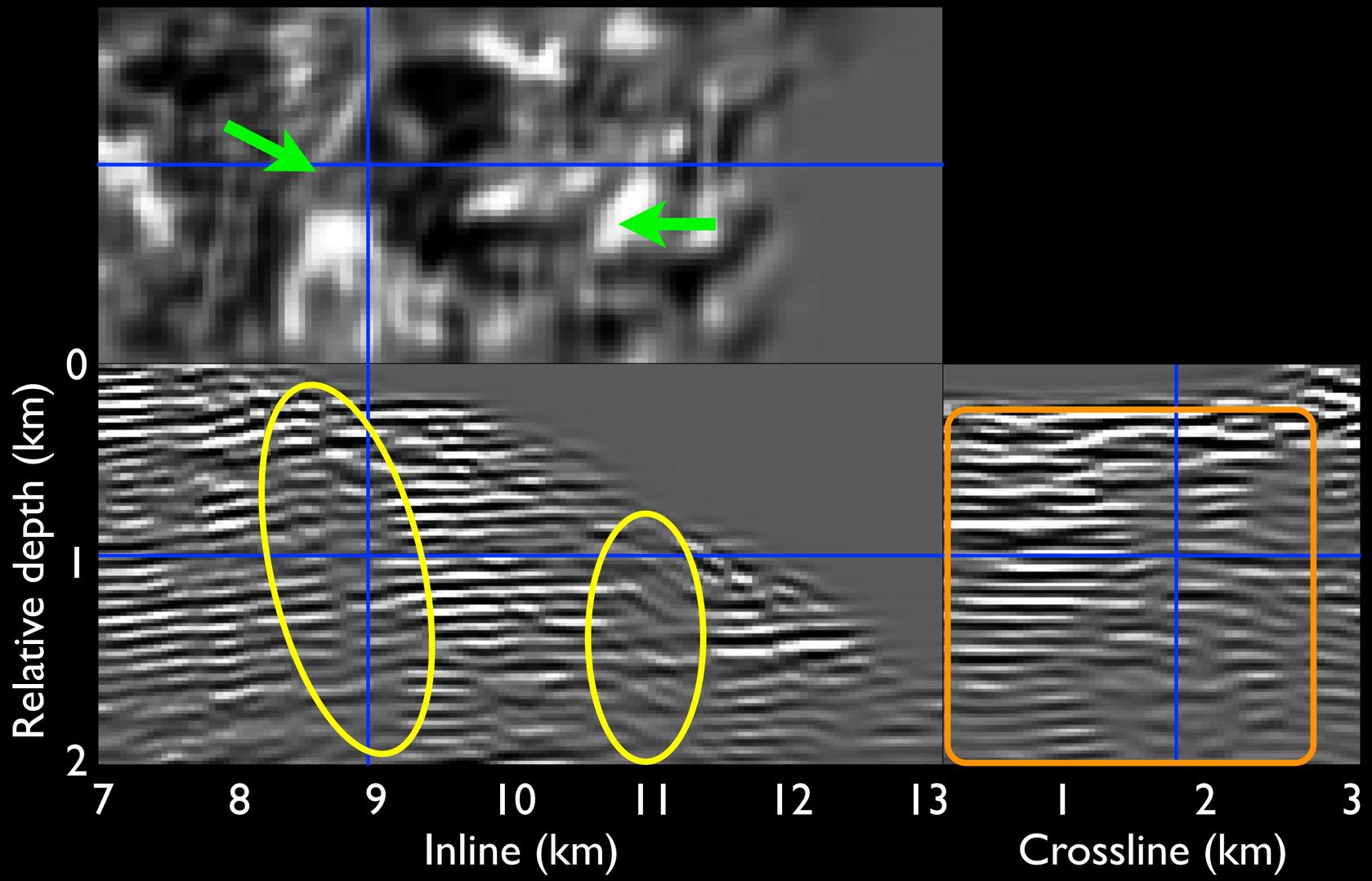
Migration


0
1
2
Relative depth (km)
7
8
9
10
11
12
13
Inline (km)
1
2
3
Crossline (km)

Migration


0
1
2
Relative depth (km)
7
8
9
10
11
12
13
Inline (km)
1
2
3
Crossline (km)

Least-squares migration (linearized inversion)

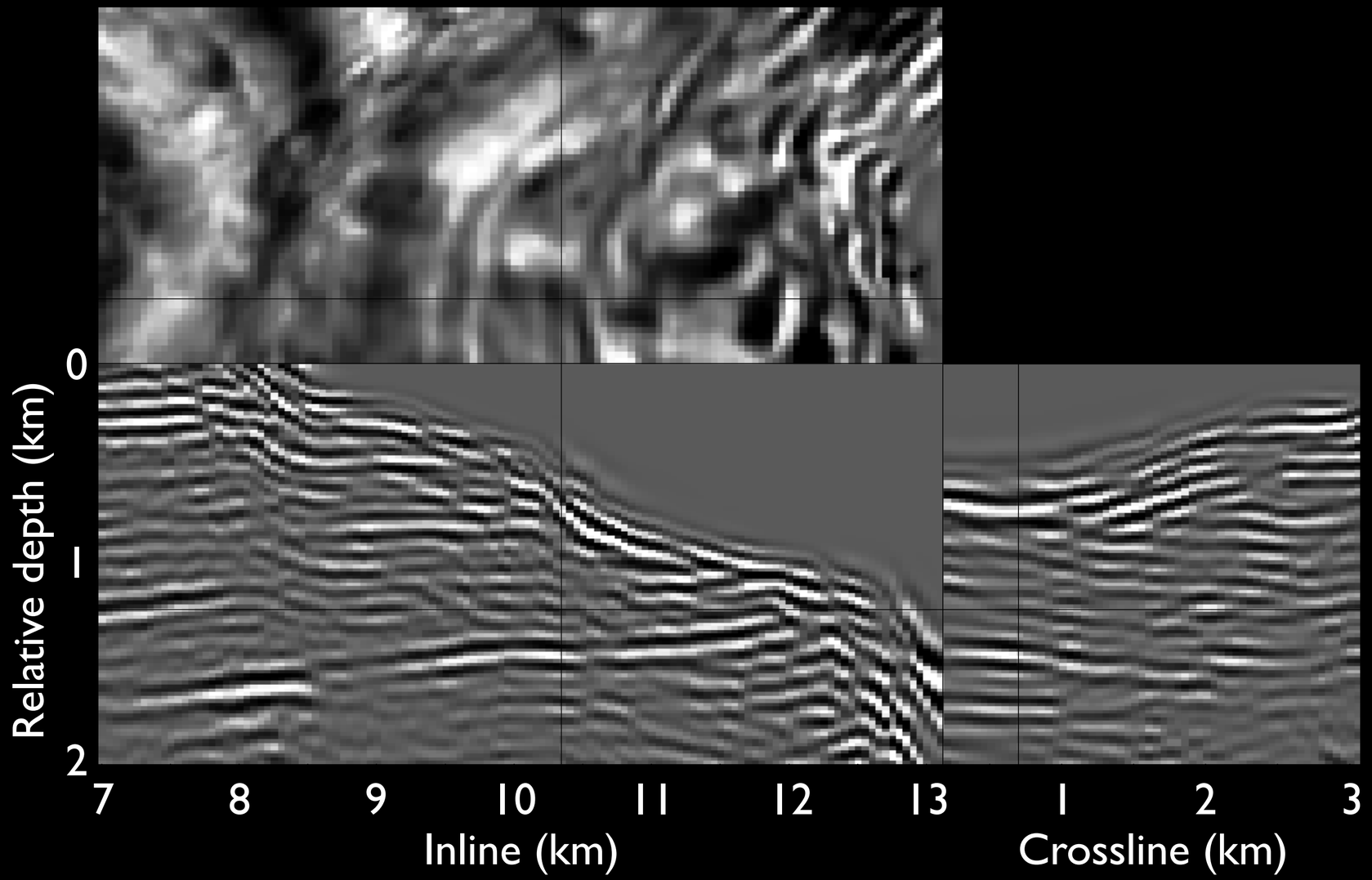


A final comparison

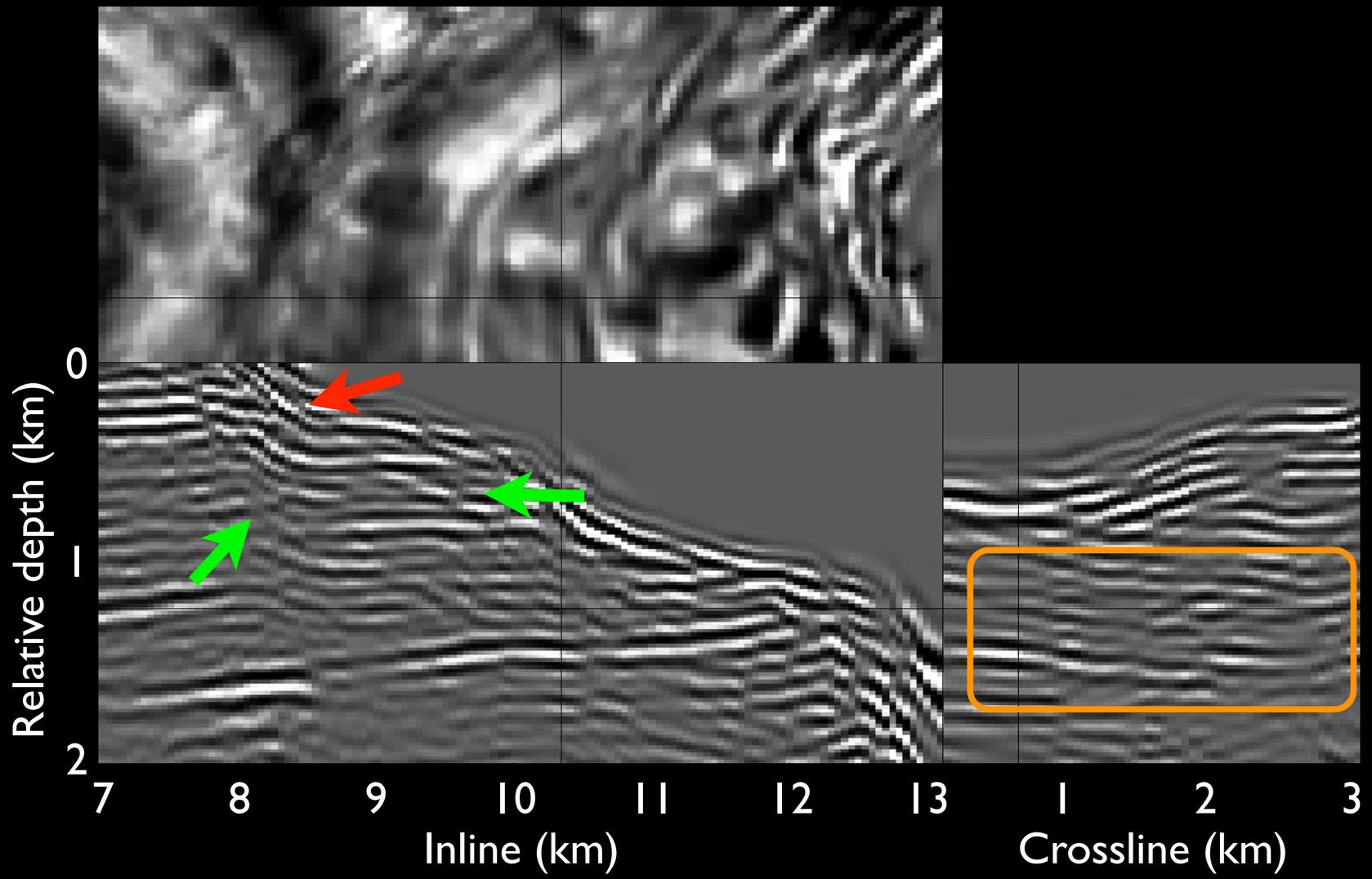
- **Migration image with AGC**

- **Least-squares migration (linearized inversion)**

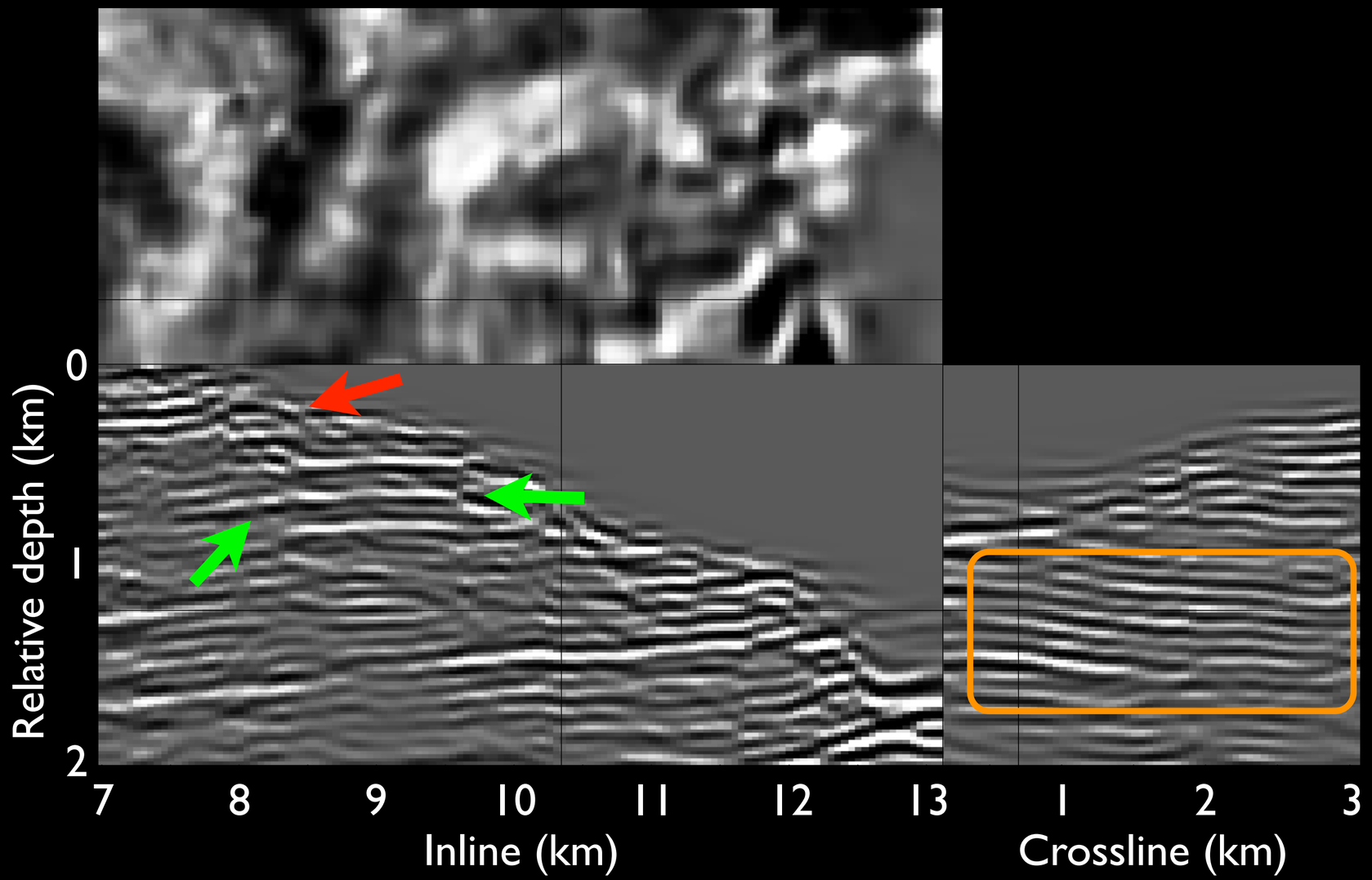
Migration with AGC



Migration with AGC



Least-squares migration (linearized inversion)



Conclusions

- **Migration produces subsalt images with shadow zones**
- **Linearized inversion fills in shadow zones and sharpens the image (higher resolution)**
- **Target-oriented Hessian can be very efficiently computed using phase encoding**

Conclusions

- Migration produces subsalt images with shadow zones
- Linearized inversion fills in shadow zones and sharpens the image (higher resolution)
- Target-oriented Hessian can be very efficiently computed using phase encoding

Conclusions

- Migration produces subsalt images with shadow zones
- Linearized inversion fills in shadow zones and sharpens the image (higher resolution)
- Target-oriented Hessian can be very efficiently computed using phase encoding

Acknowledgements

- SMAART JV for the Sigsbee2A synthetic data set
- BP and ExxonMobil for the GOM field data set
- Stanford Center for Computational Earth and Environmental Science (CEES) for providing computing resources

Thanks for your attention

Exact Hessian vs. approx. Hessian

The exact Hessian:

$$H(\mathbf{x},\mathbf{y}) = \sum_{\omega}\sum_{\mathbf{x}_s} G(\mathbf{x},\mathbf{x}_s,\omega)G'(\mathbf{y},\mathbf{x}_s,\omega)$$

$$\times \sum_{\mathbf{x}_r} w(\mathbf{x}_r,\mathbf{x}_s)G(\mathbf{x},\mathbf{x}_r,\omega)G'(\mathbf{y},\mathbf{x}_r,\omega)$$

$$a_1 b_1 + \cdots + a_n b_n$$

The approx. Hessian:

$$\widetilde{H}(\mathbf{x},\mathbf{y}) = \sum_{\omega}\sum_{\mathbf{x}_s} G(\mathbf{x},\mathbf{x}_s,\omega)G'(\mathbf{y},\mathbf{x}_s,\omega)$$

$$\times R(\mathbf{x},\omega;\mathbf{x}_s)R'(\mathbf{y},\omega;\mathbf{x}_s)$$

$$(a_1 + \cdots + a_n)(b_1 + \cdots + b_n) = a_1 b_1 + \cdots + a_n b_n + \sum_{i\neq j} a_i b_j$$

Whoops, cross-talk...

$$\widetilde{H}(\mathbf{x},\mathbf{y}) = H(\mathbf{x},\mathbf{y}) + C(\mathbf{x},\mathbf{y})$$

True Hessian Crosstalk

$$(a_1 + \cdots + a_n)(b_1 + \cdots + b_n) = a_1 b_1 + \cdots + a_n b_n + \sum_{i \neq j} a_i b_j$$

Whoops, cross-talk...

$$\widetilde{H}(\mathbf{x},\mathbf{y}) = H(\mathbf{x},\mathbf{y}) + C(\mathbf{x},\mathbf{y})$$

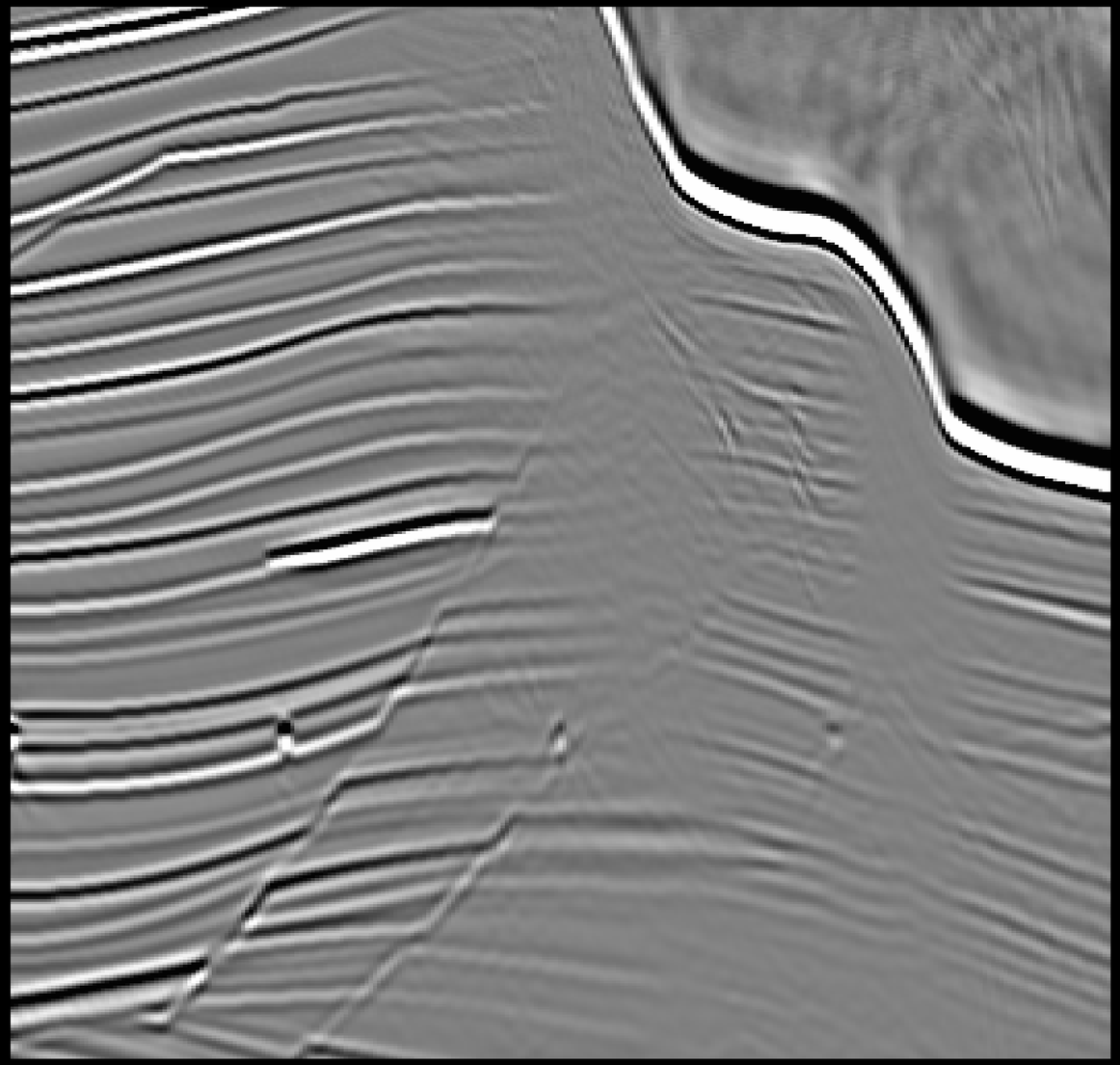
True Hessian Crosstalk

$$(a_1 + \cdots + a_n)(b_1 + \cdots + b_n) = a_1 b_1 + \cdots + a_n b_n + \sum_{i \neq j} a_i b_j$$

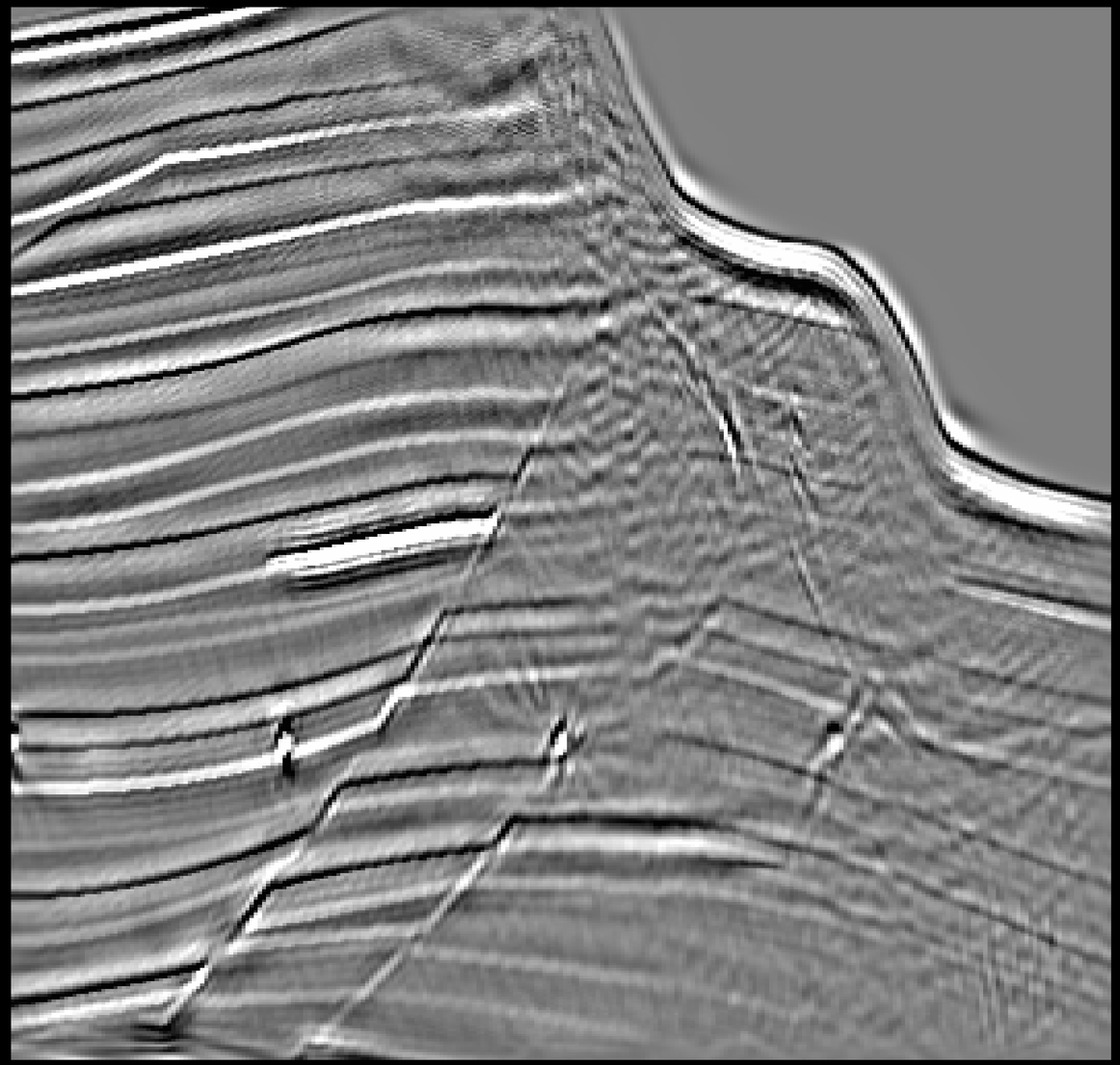
We can use phase encoding to attenuate the cross-talk !

Damped inversion

Migration

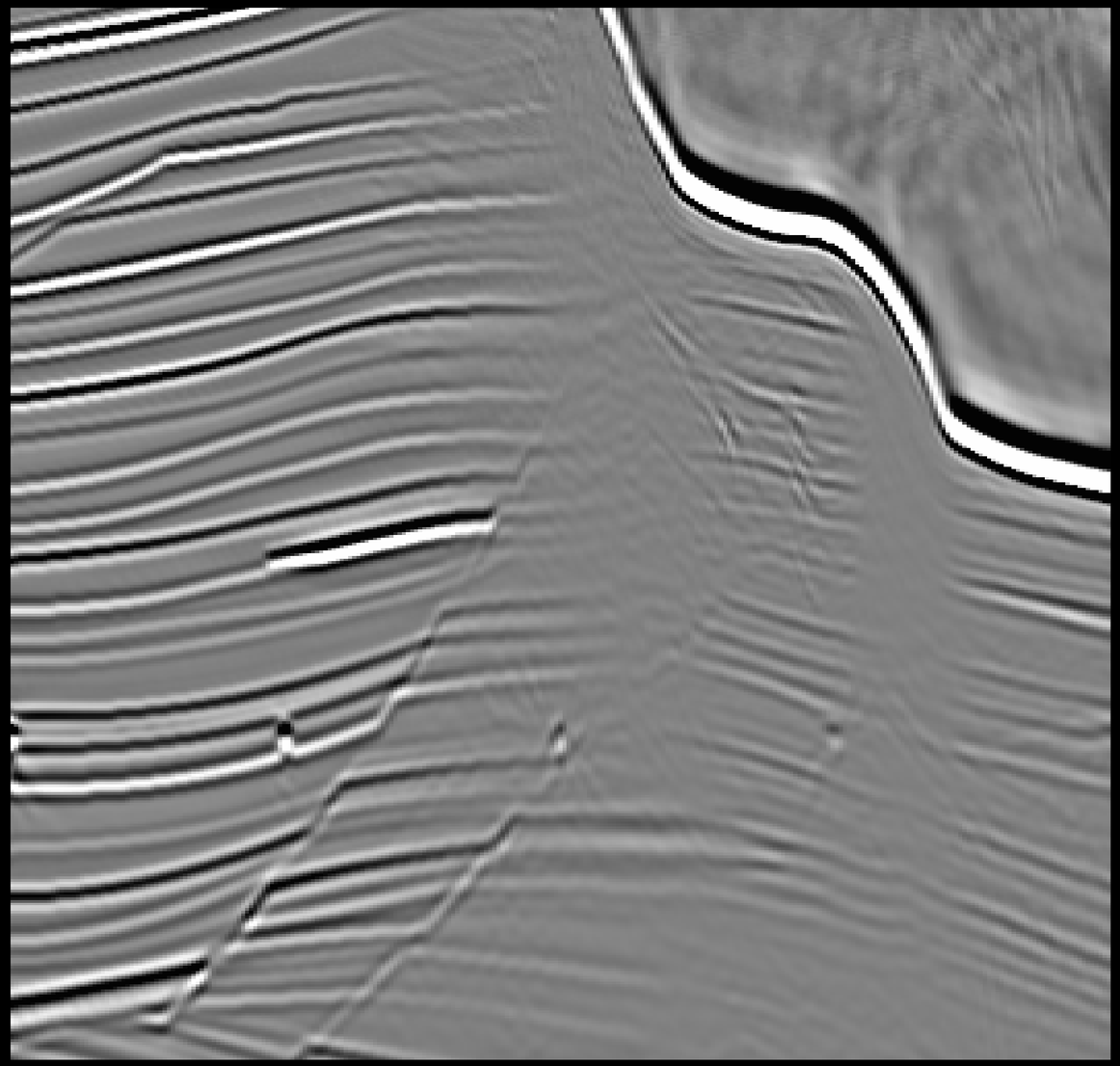


Inversion

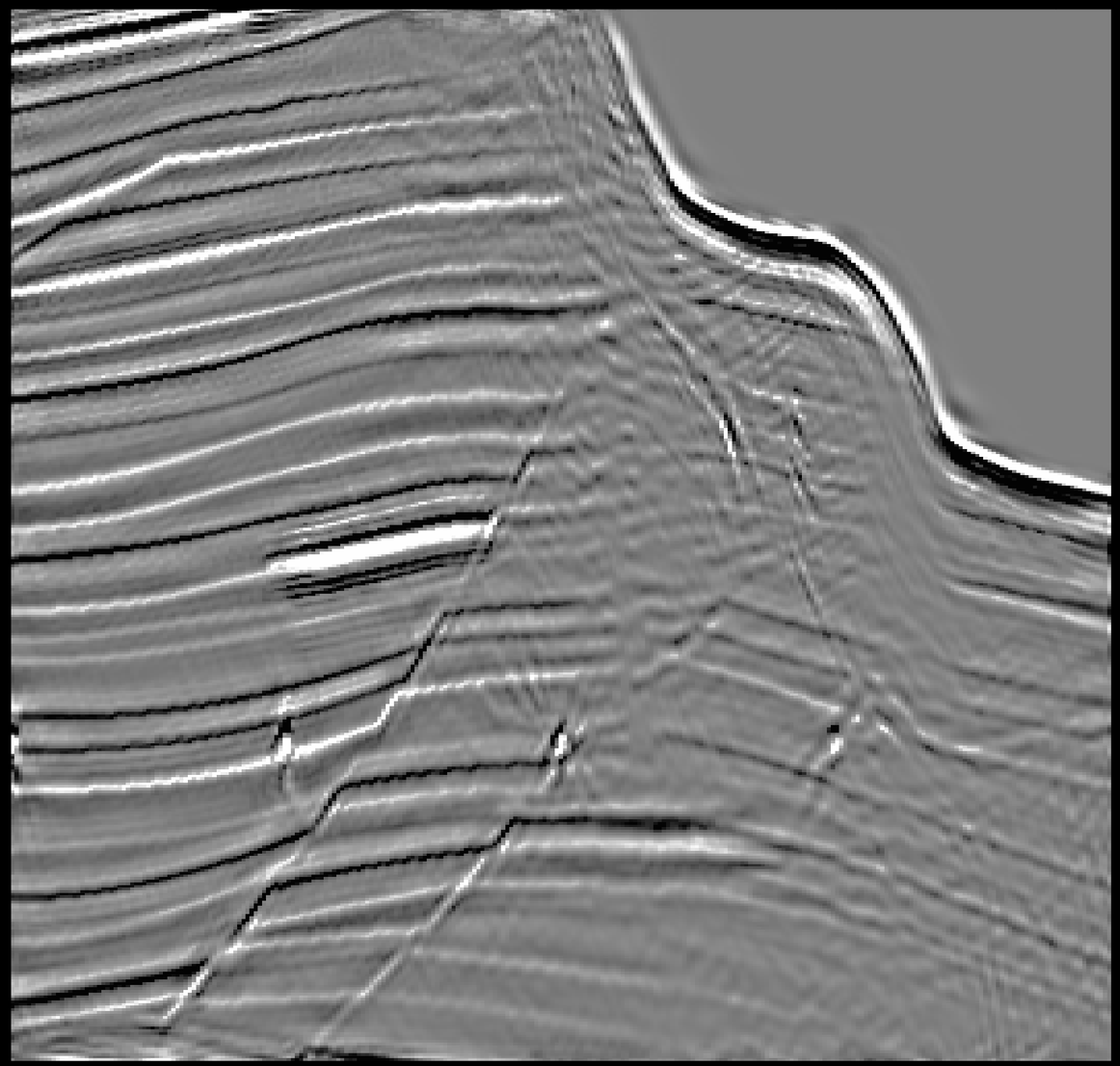


Sparsity promoted inversion

Migration

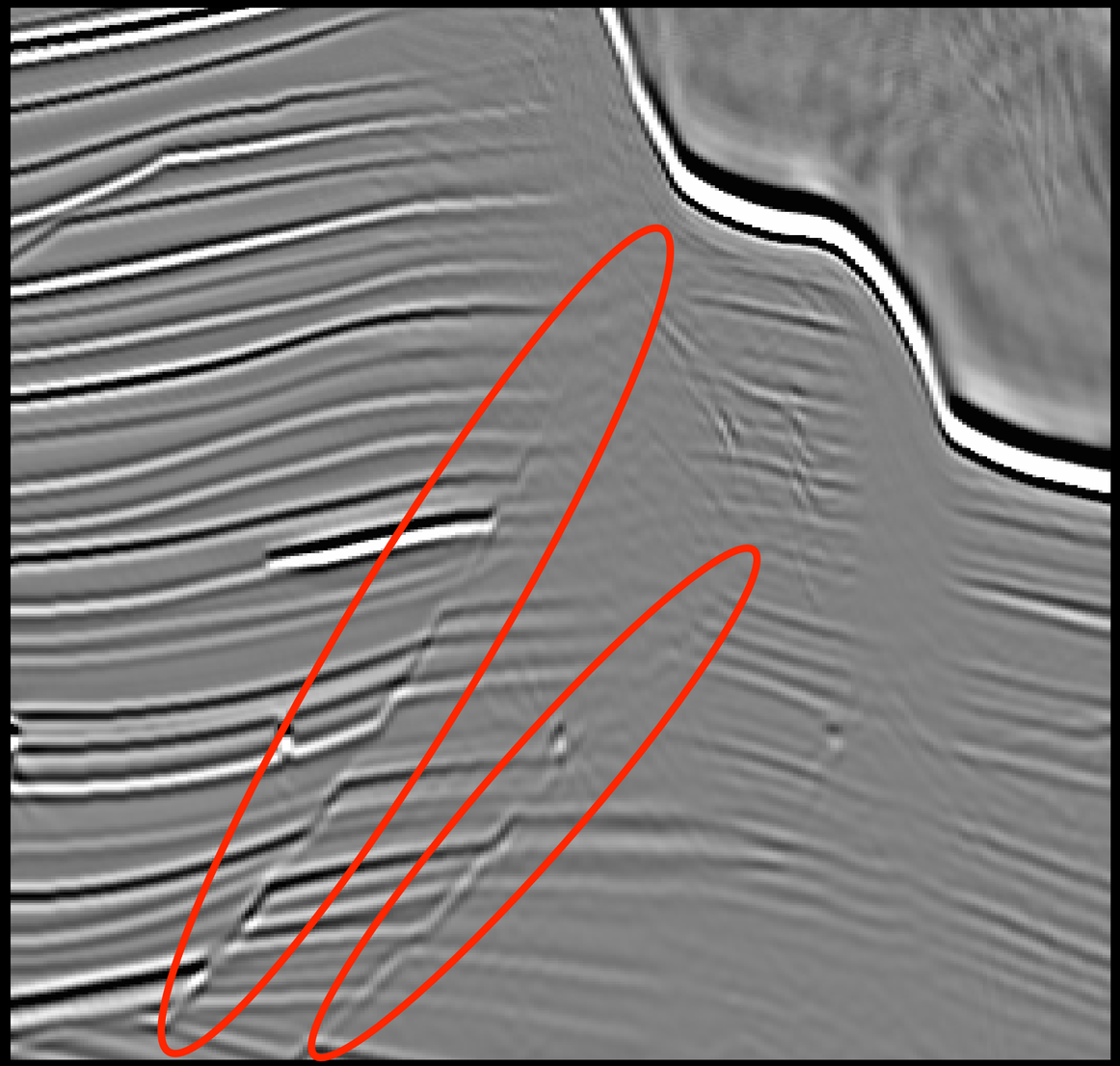


Inversion

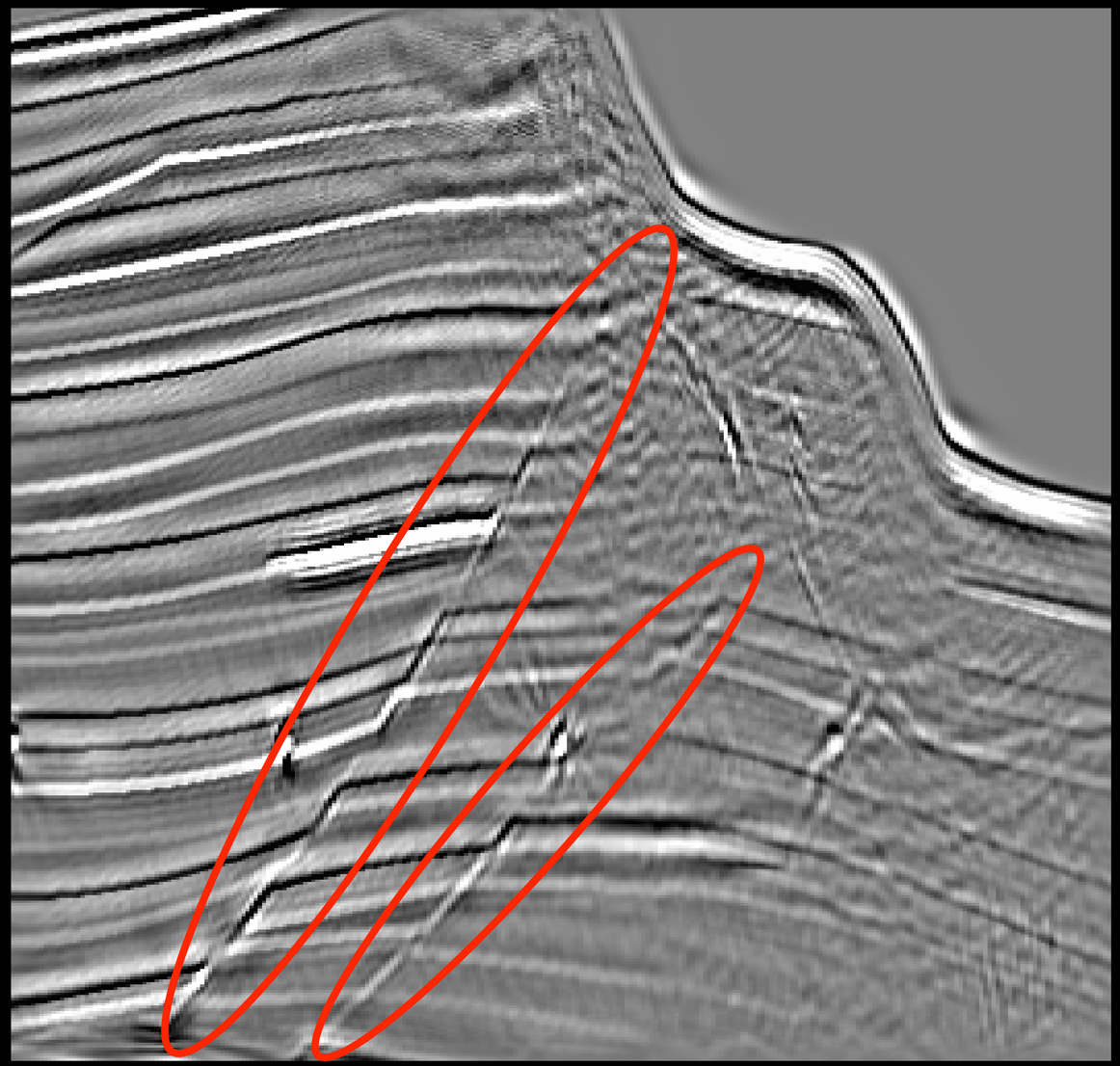


Linearized inversion fills shadow zones

Migration

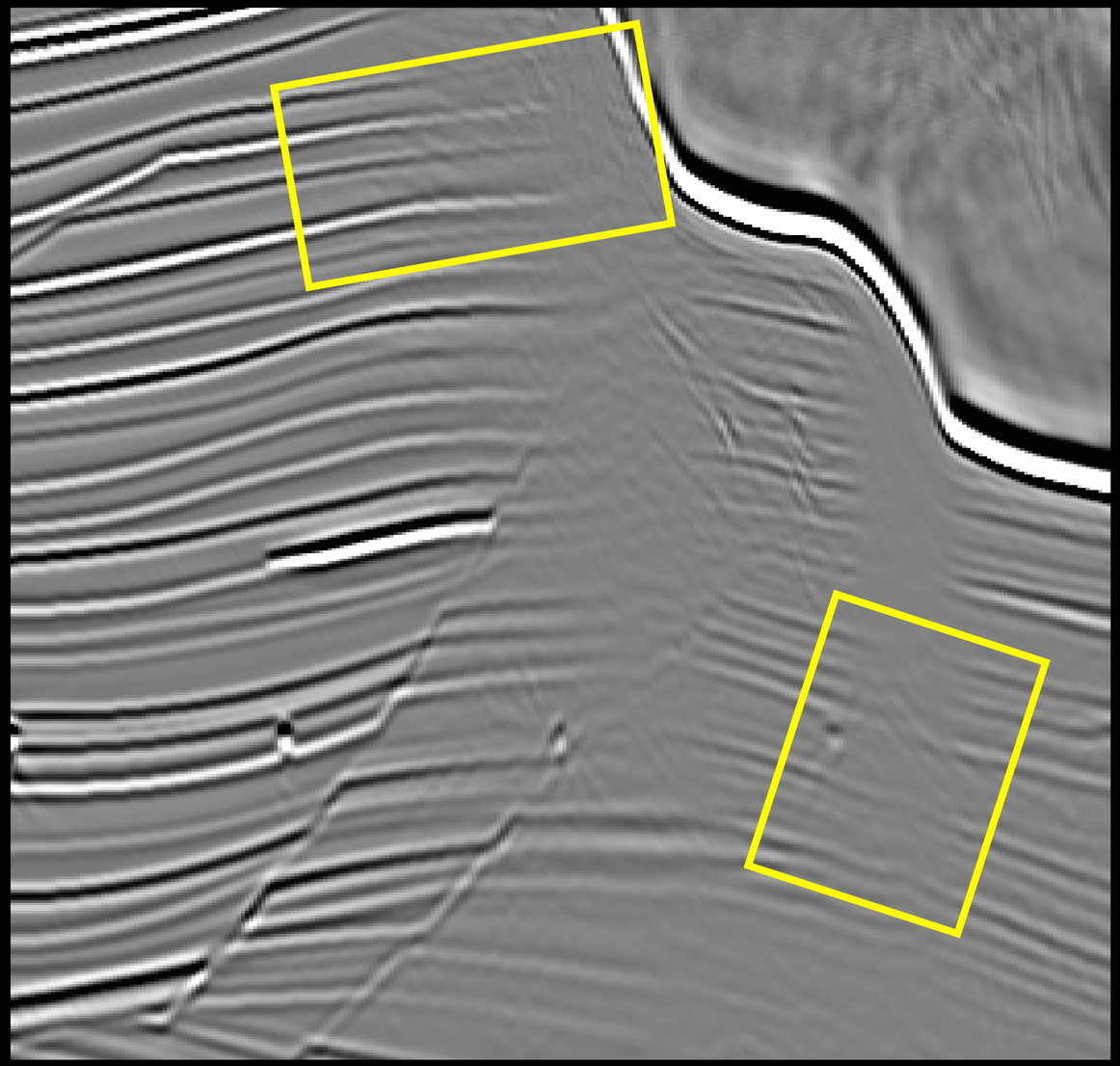


Inversion

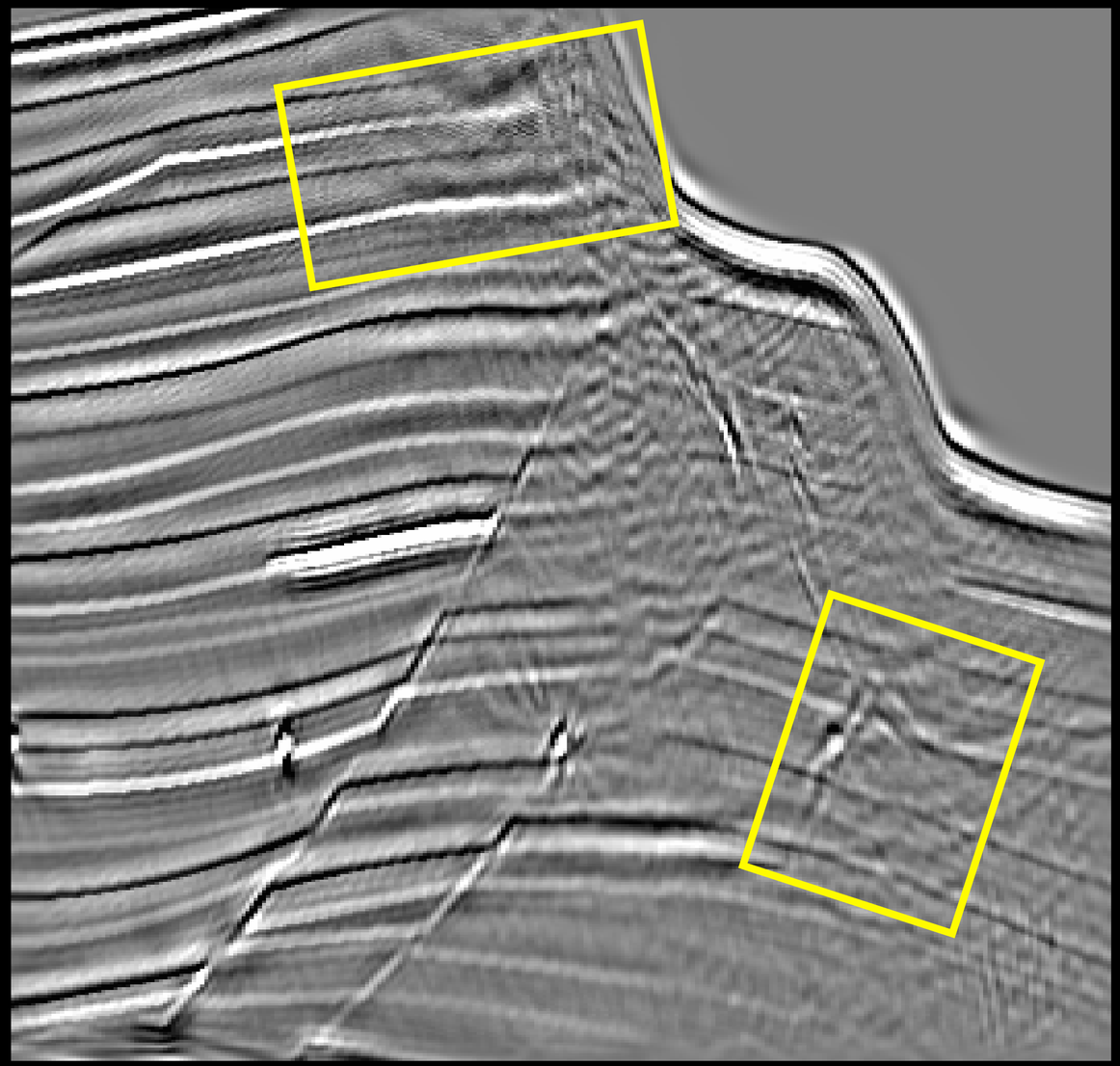


Linearized inversion fills shadow zones

Migration

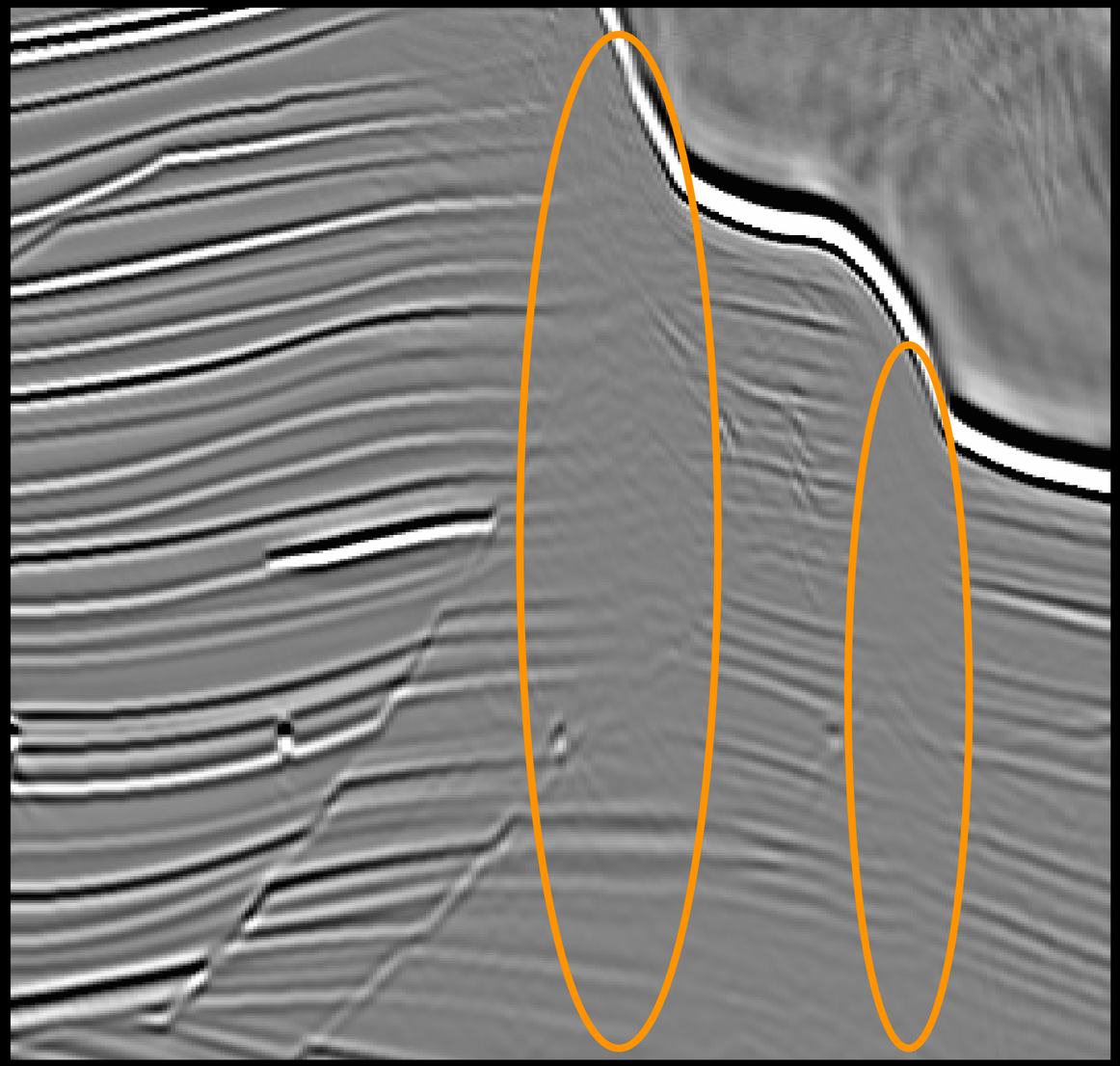


Inversion

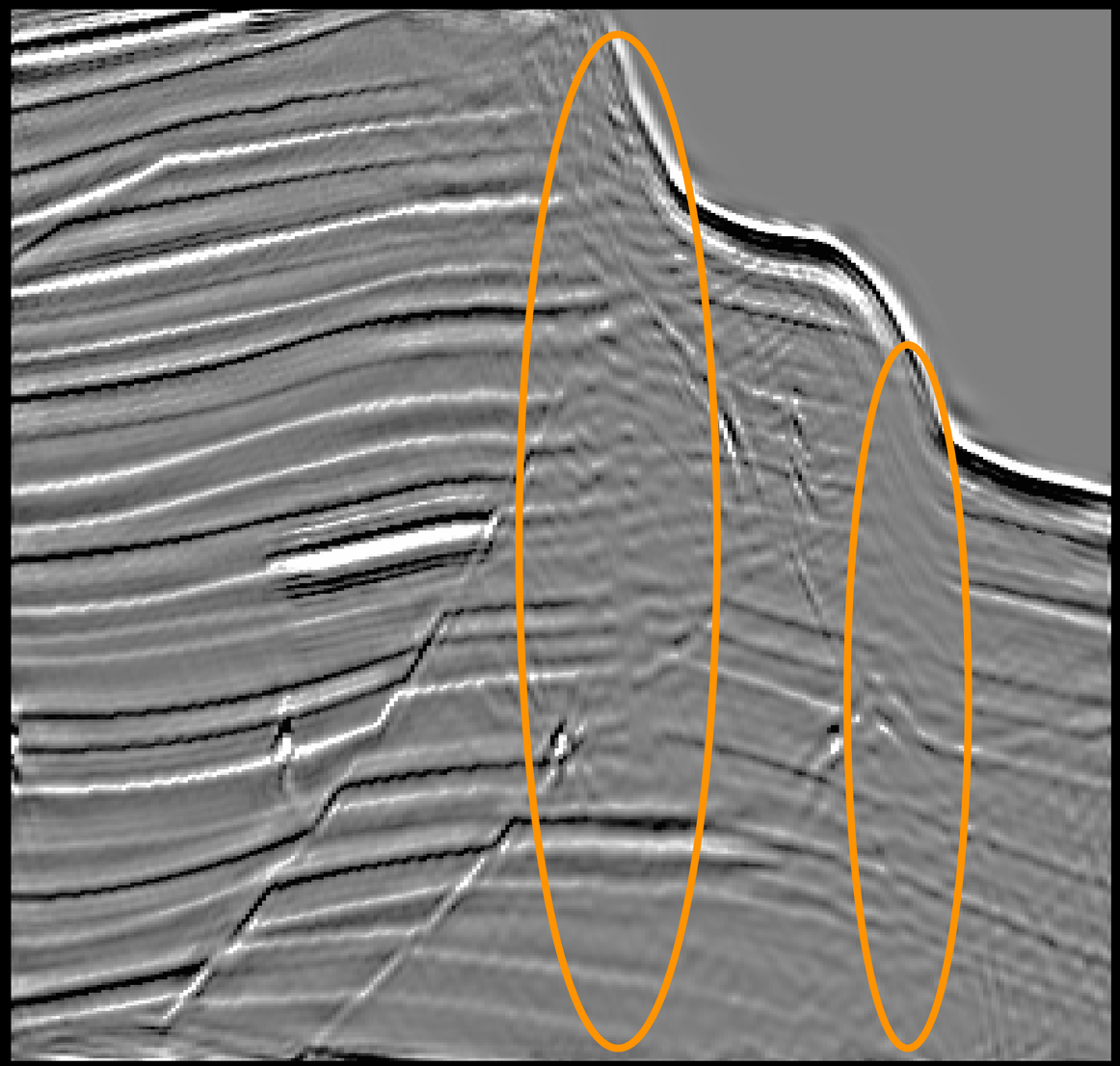


Least-squares migration fills shadow zones

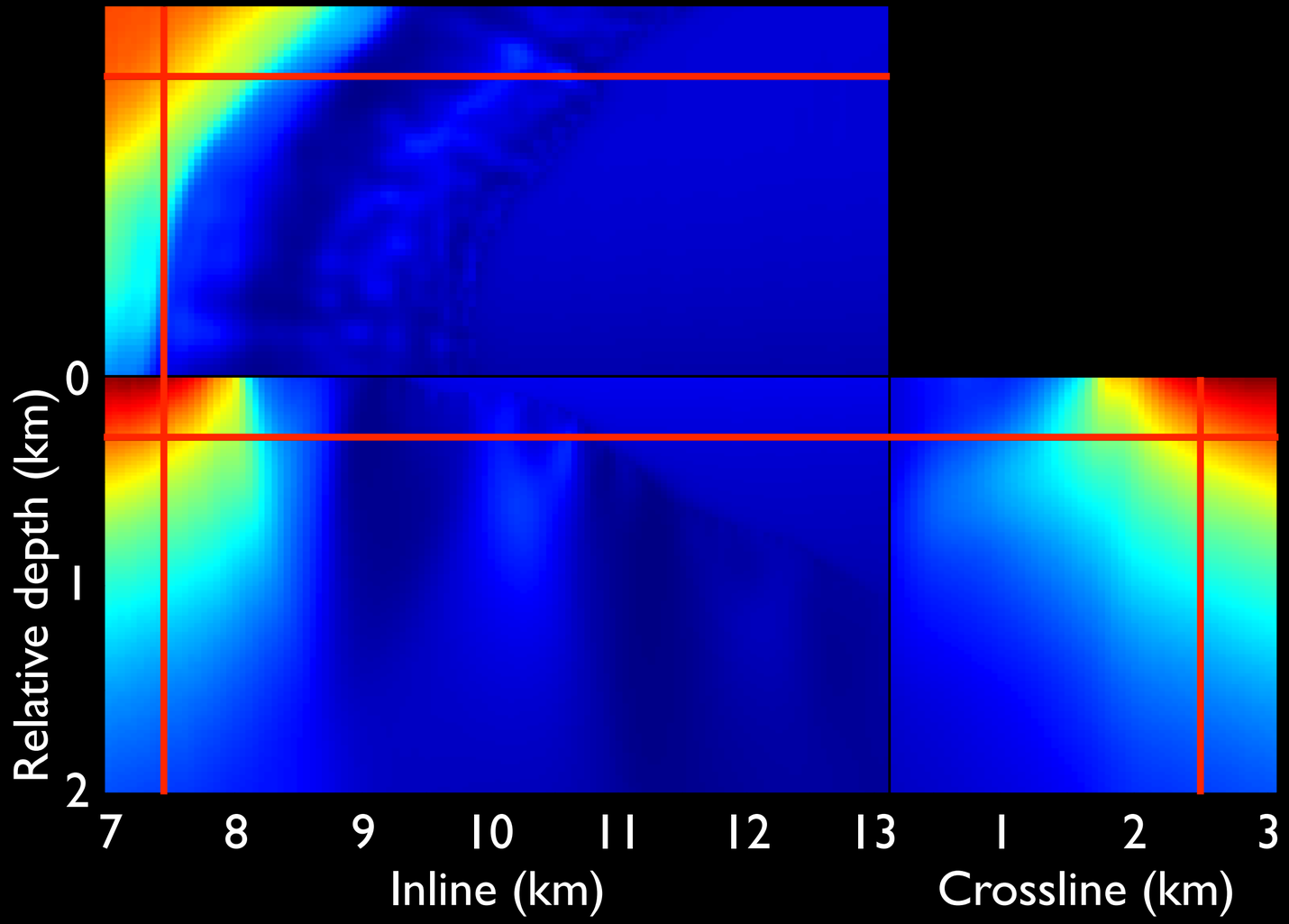
Migration



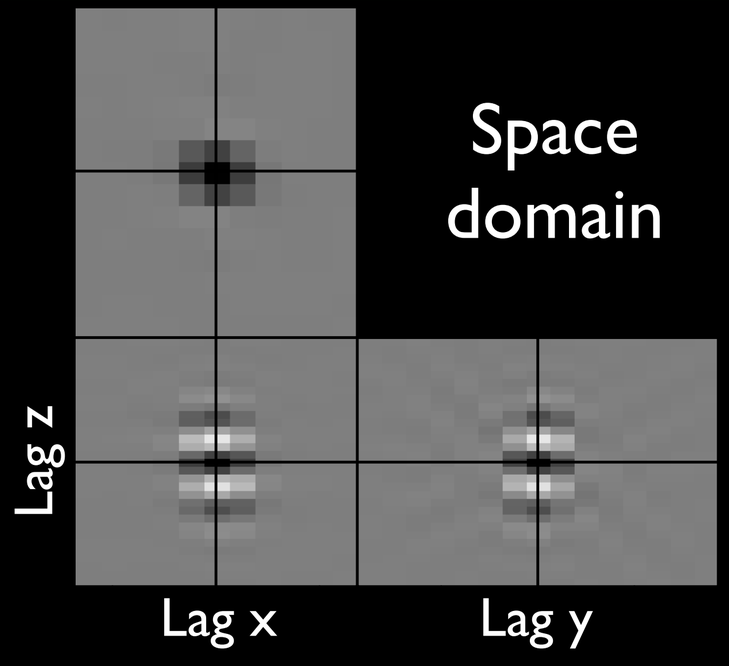
Least-squares migration

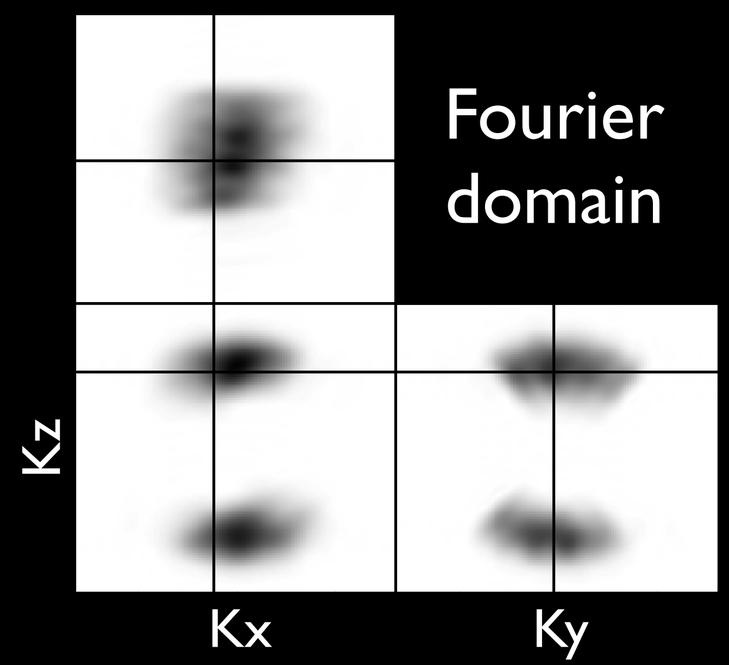


The 3-D Hessian matrix

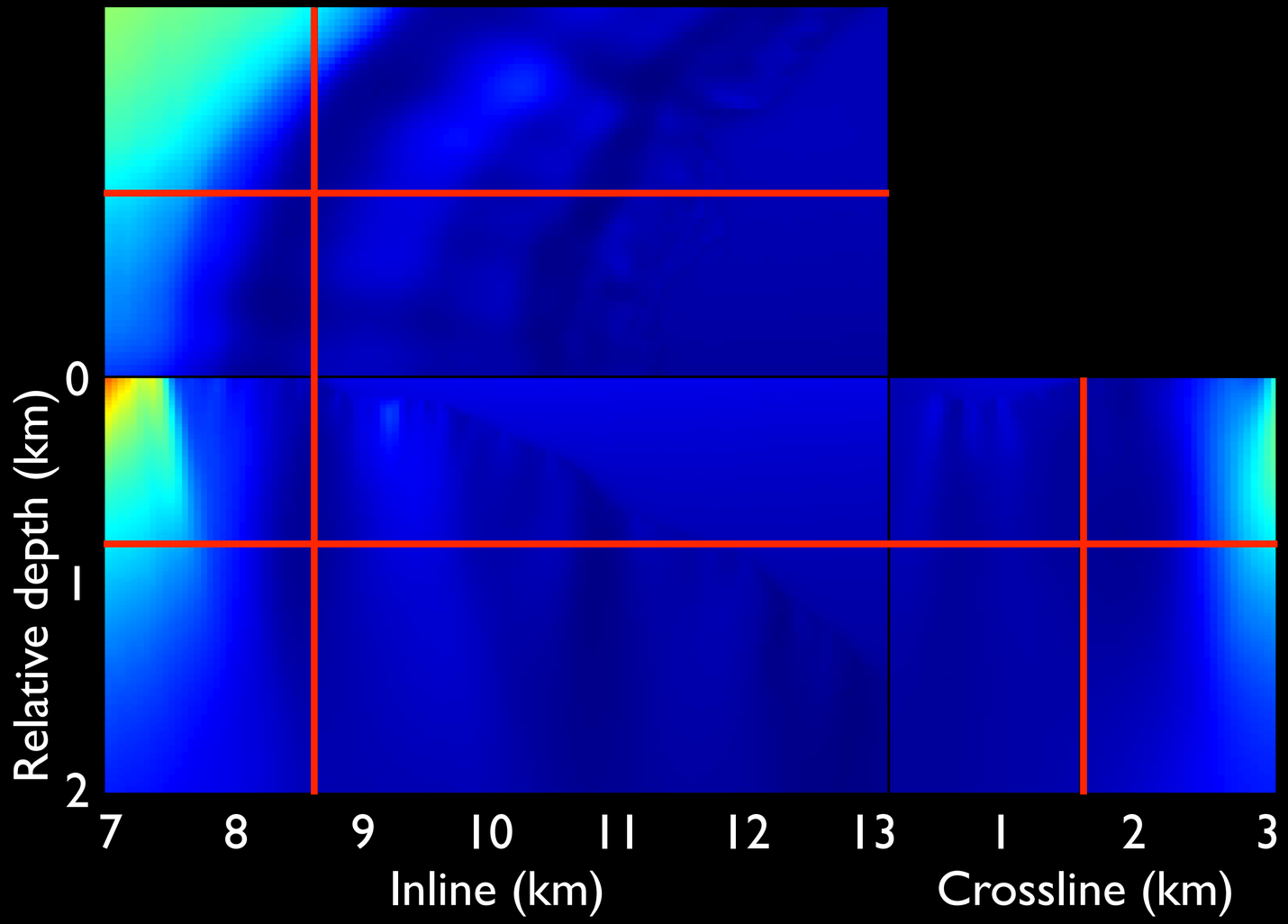


Hessian diagonals

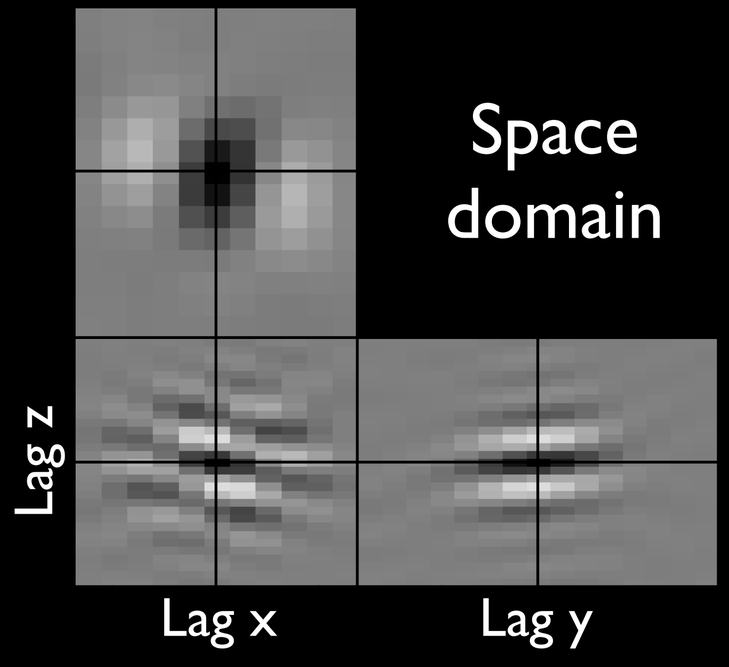


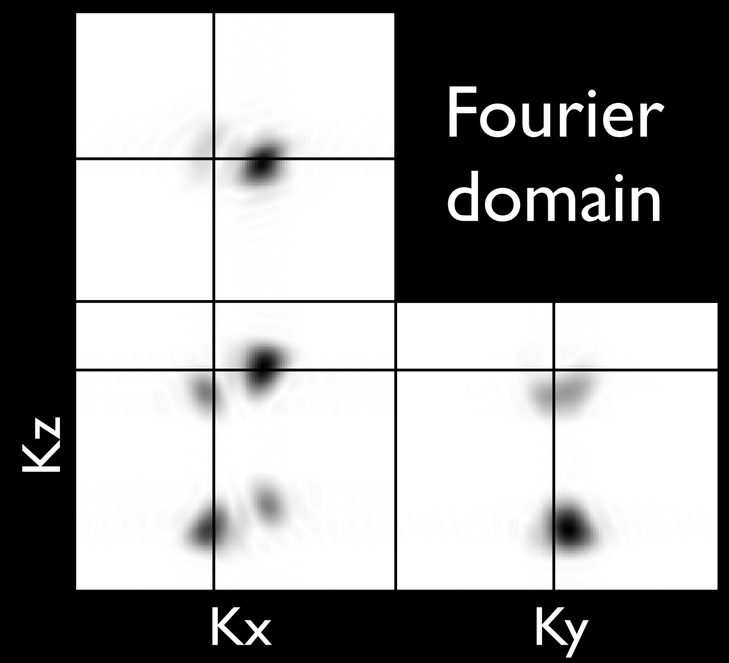


The 3-D Hessian matrix

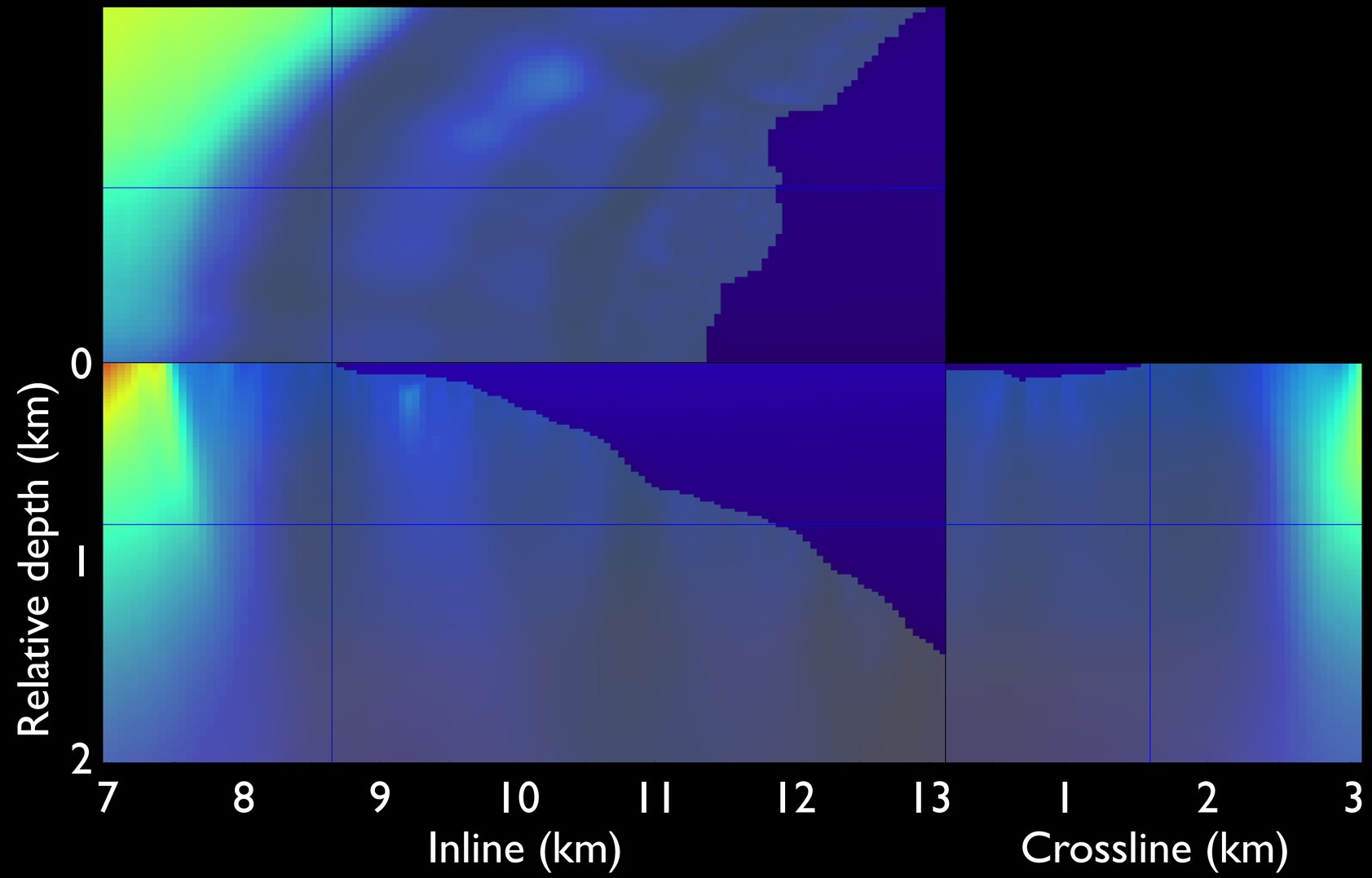


Hessian diagonals

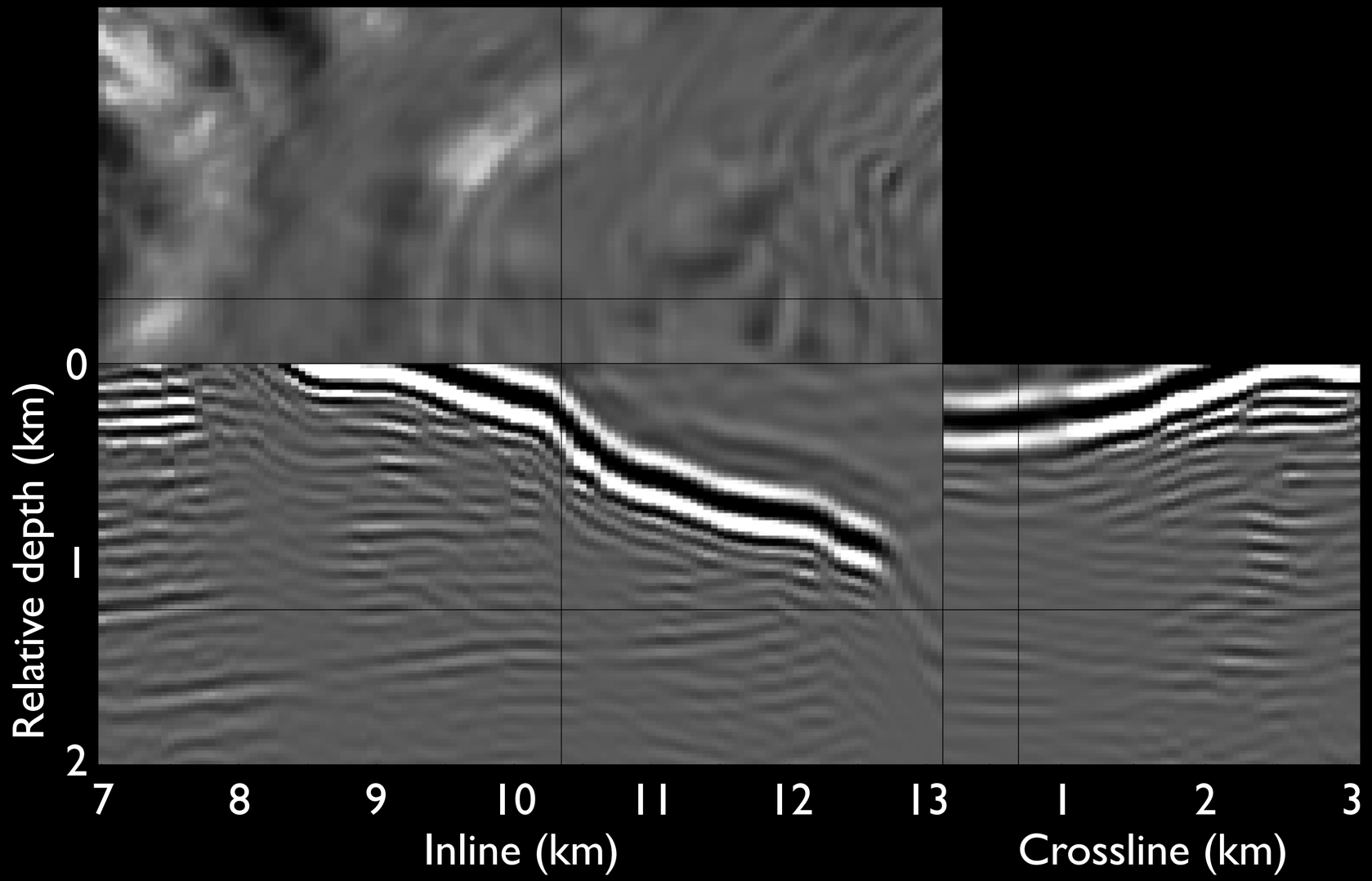




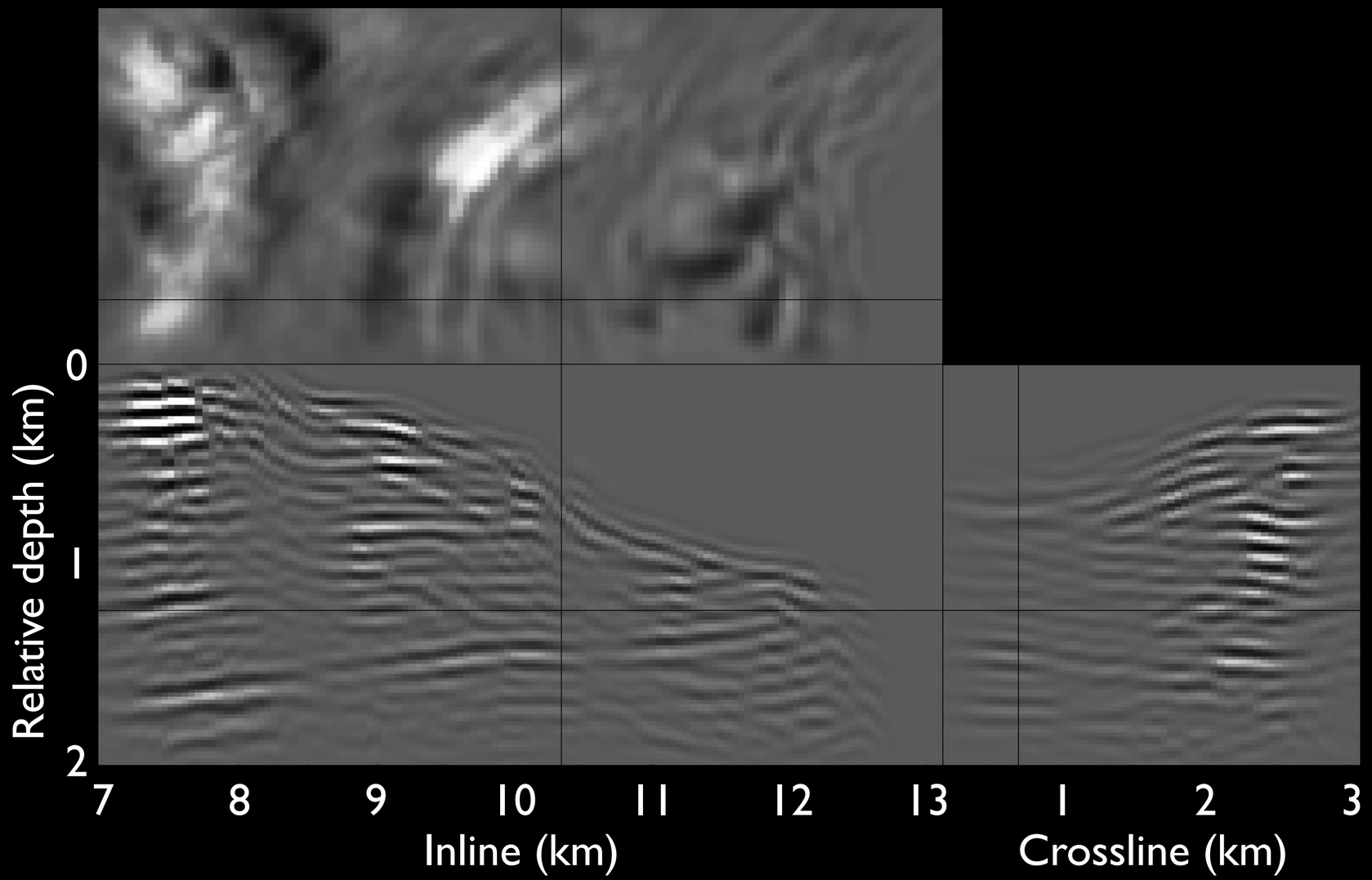
The diagonal of the 3-D Hessian



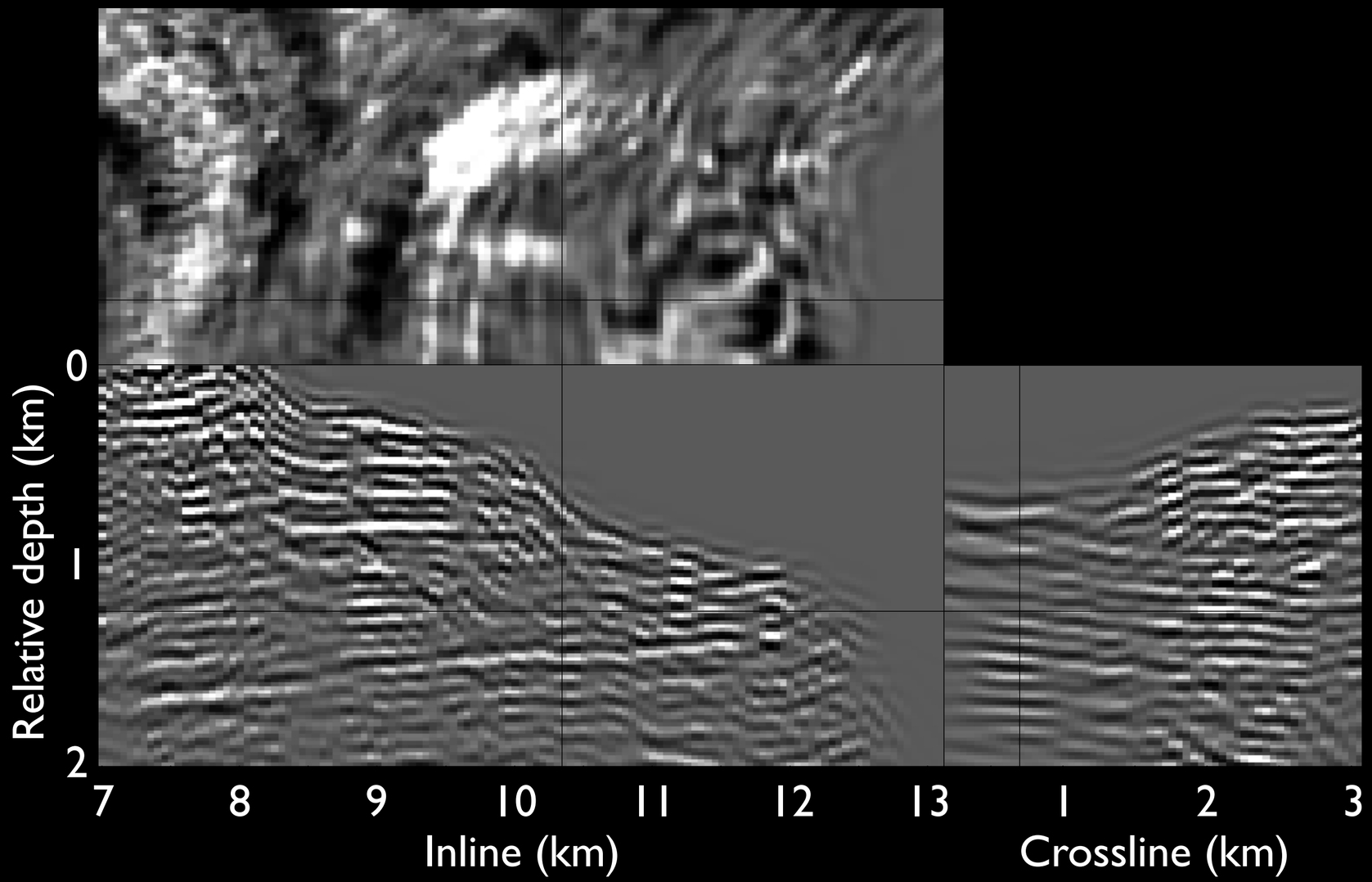
Migration


0
1
2
Relative depth (km)
7
8
9
10
11
12
13
Inline (km)
1
2
3
Crossline (km)

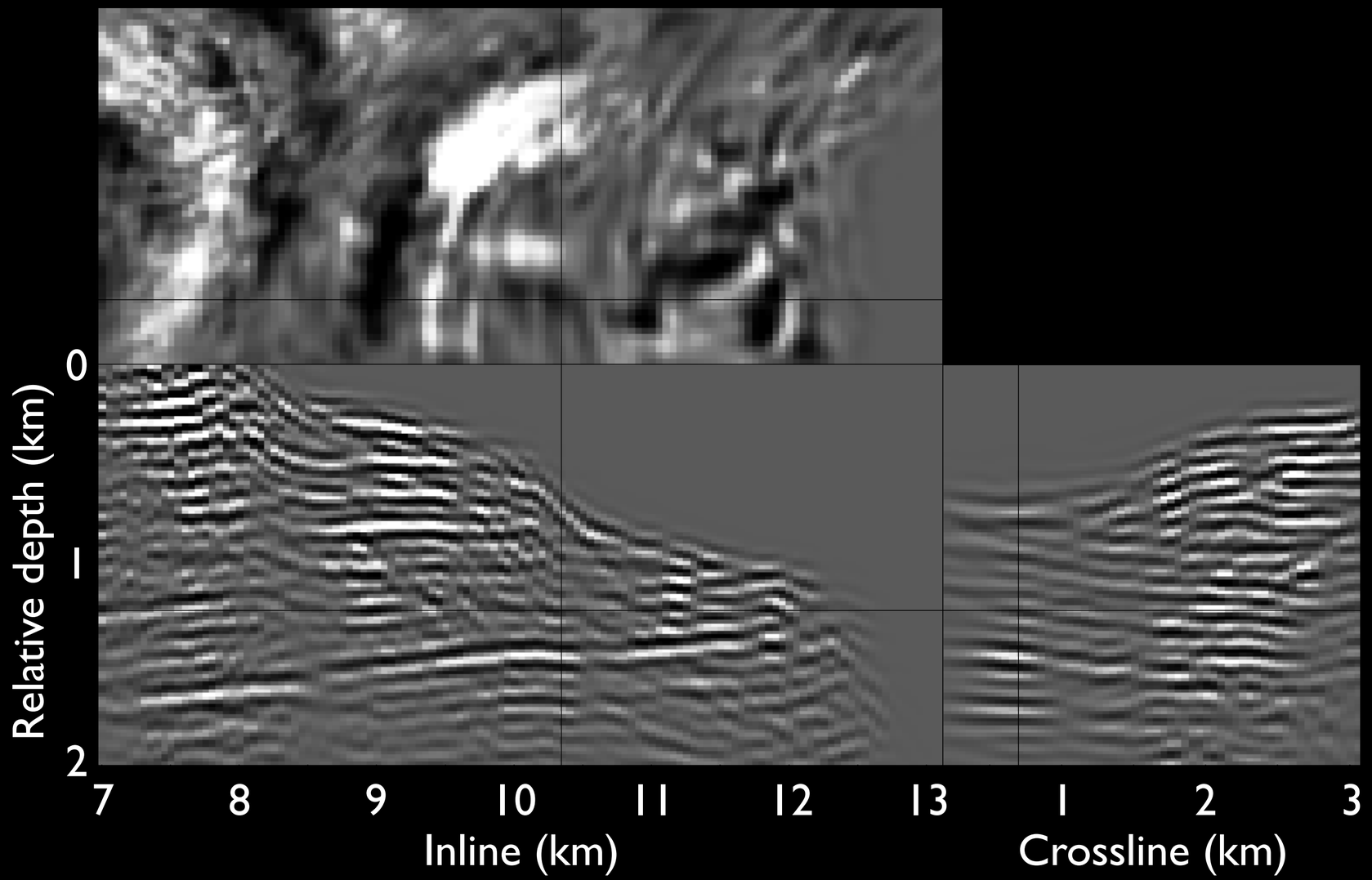
Migration


0
1
2
Relative depth (km)
7
8
9
10
11
12
13
Inline (km)
1
2
3
Crossline (km)

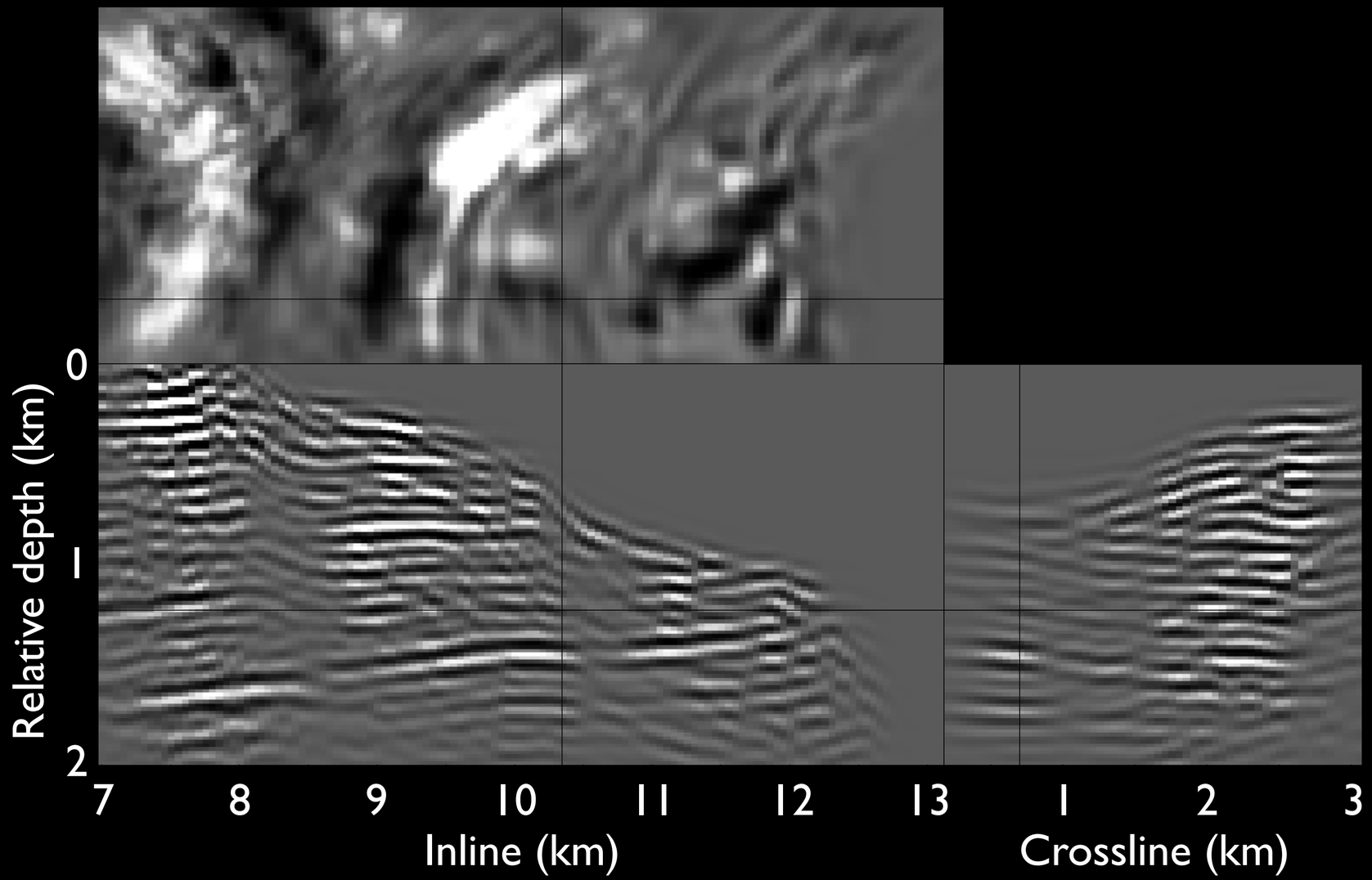
Inversion without regularization



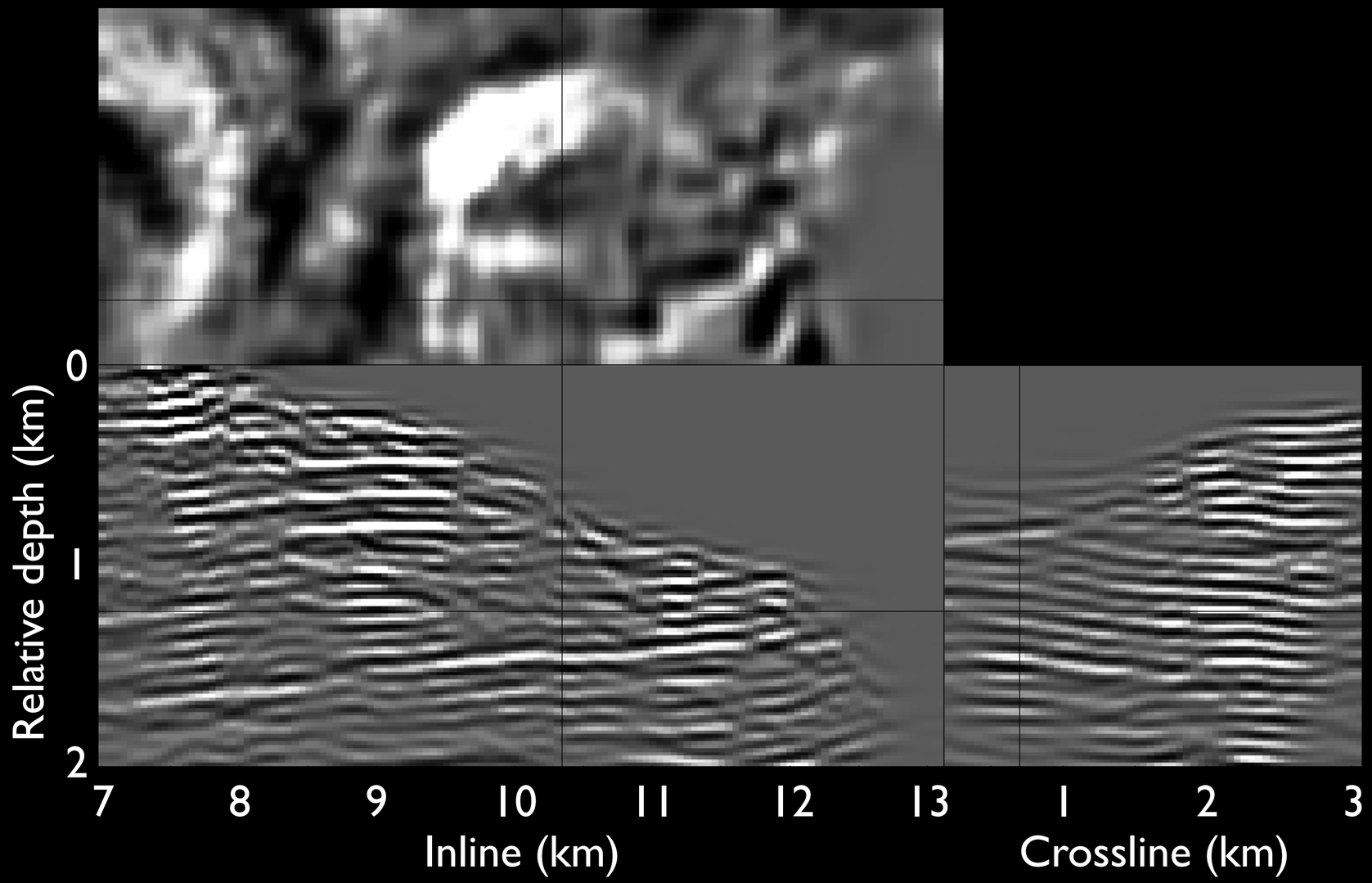
Inversion with damping ($\varepsilon = 0.02$)



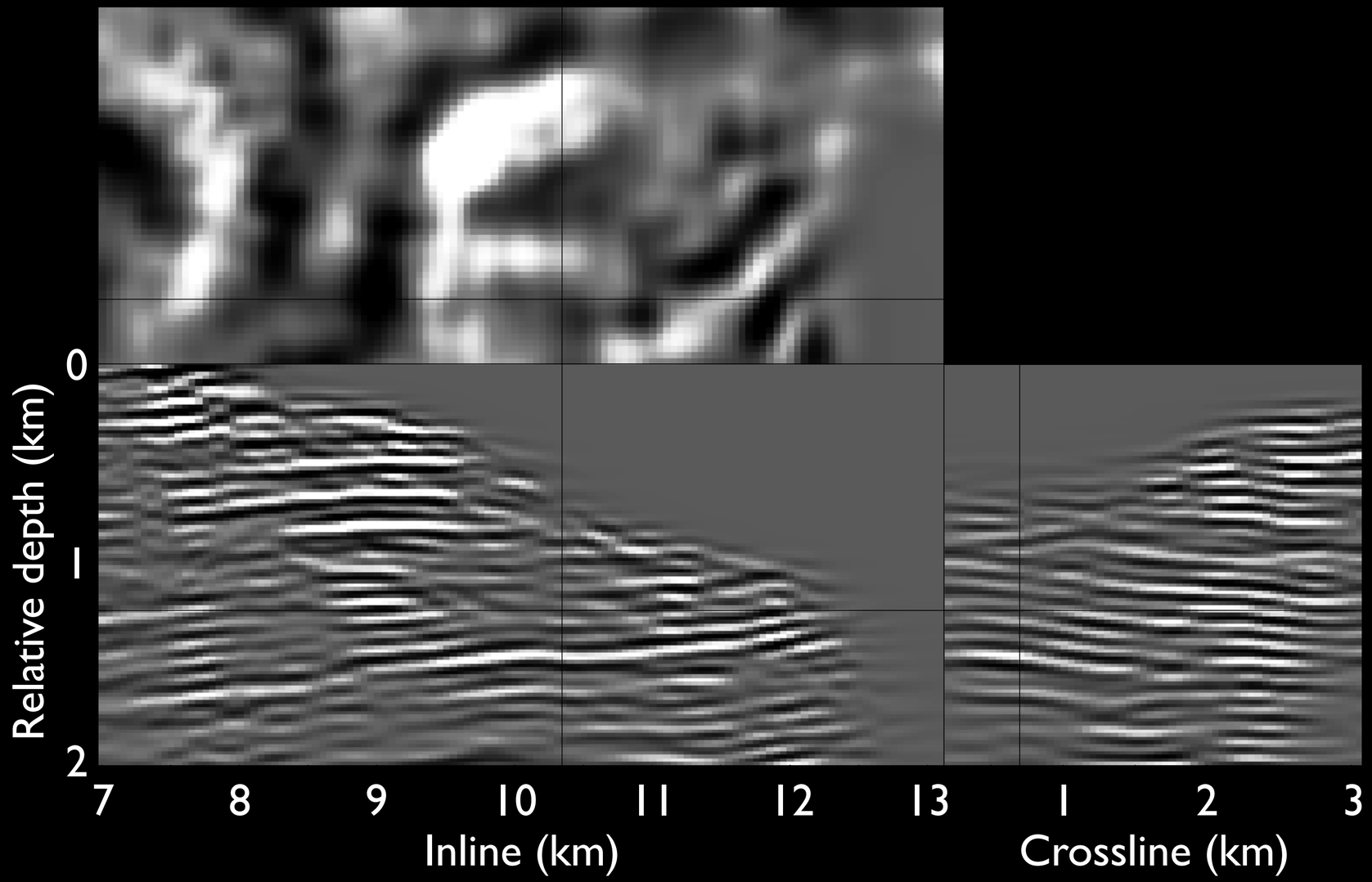
Inversion with damping ($\varepsilon = 0.05$)



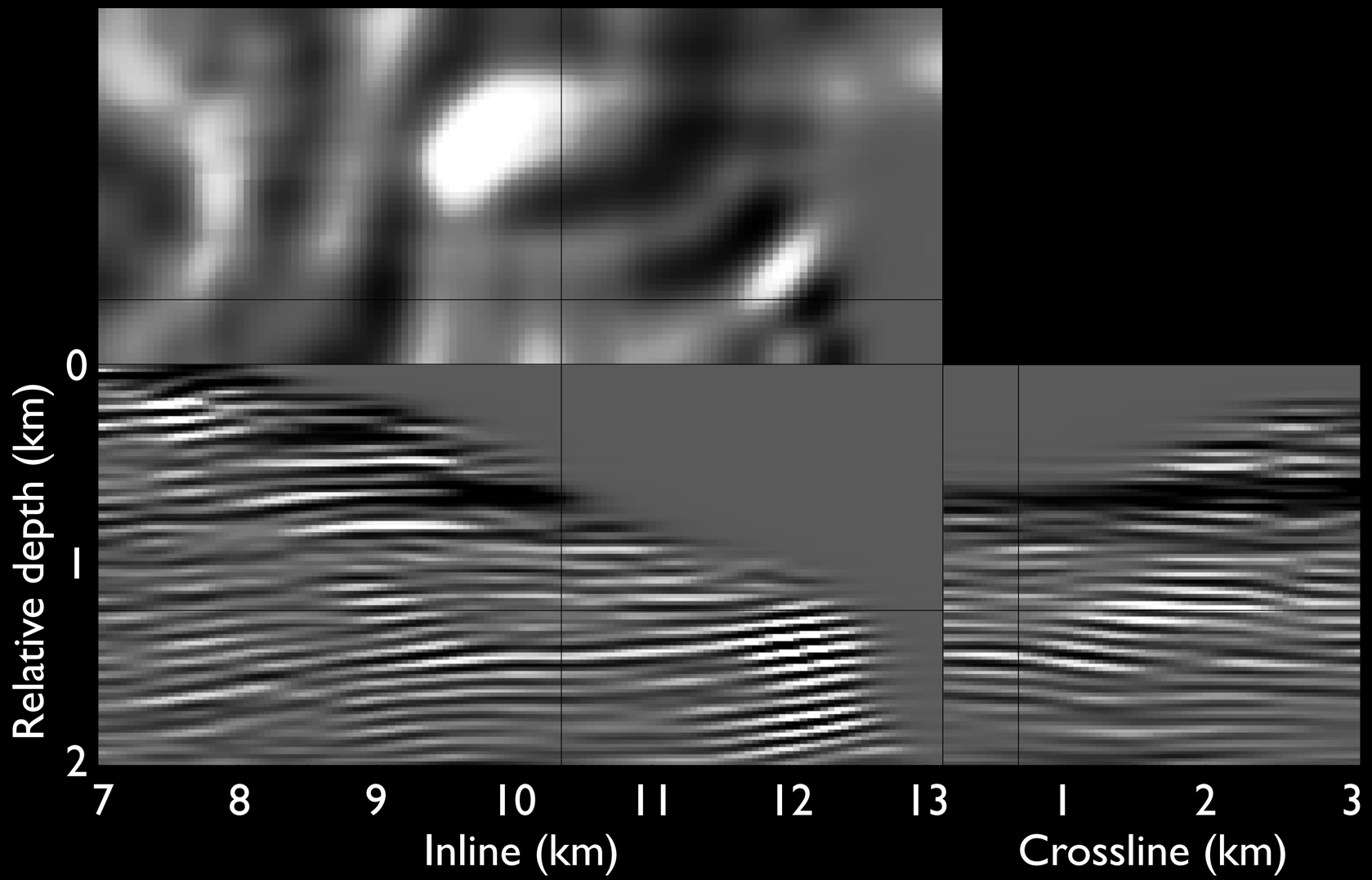
Inversion preconditioned with weak dip filtering



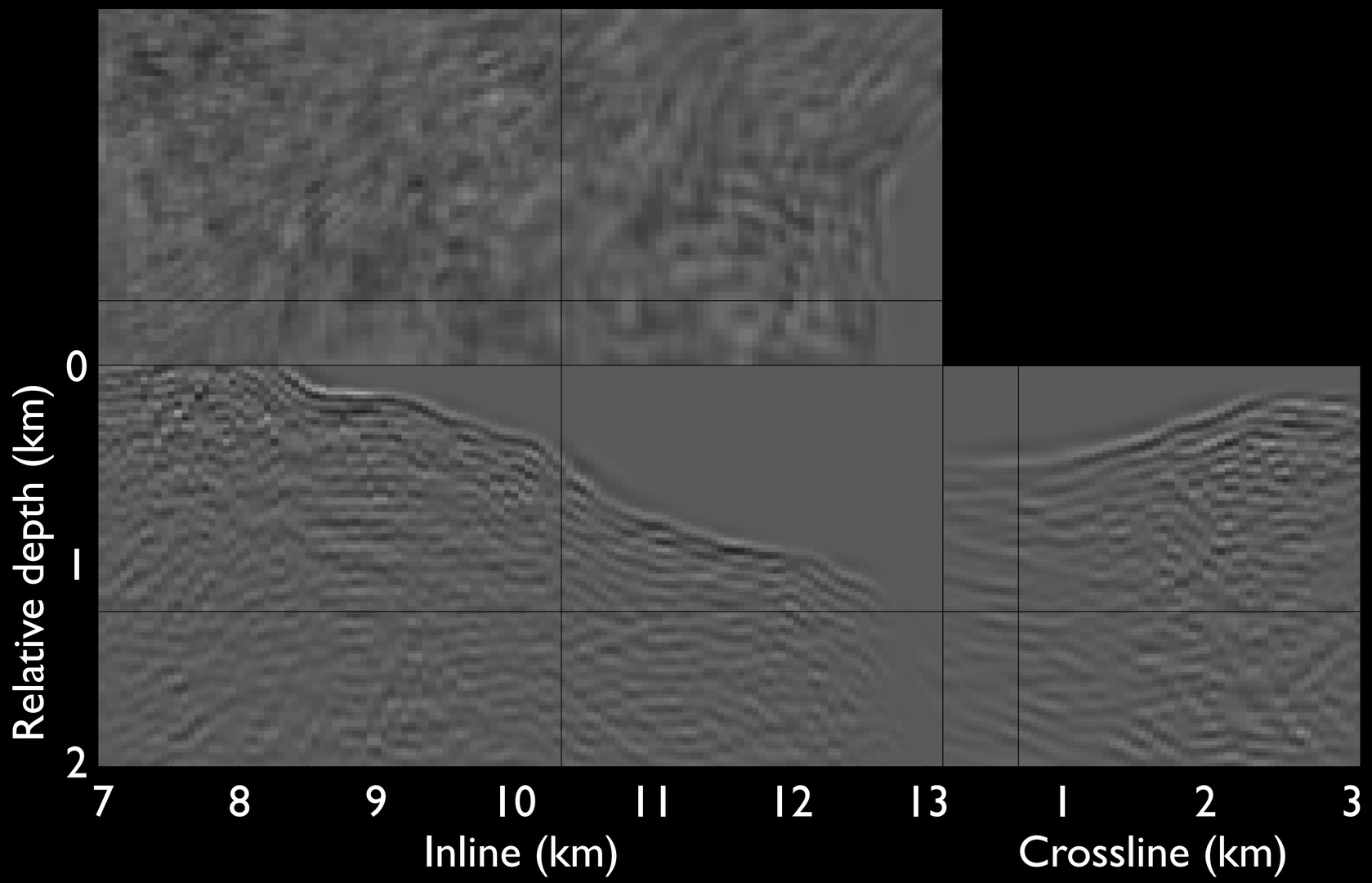
Inversion preconditioned with moderate dip filtering



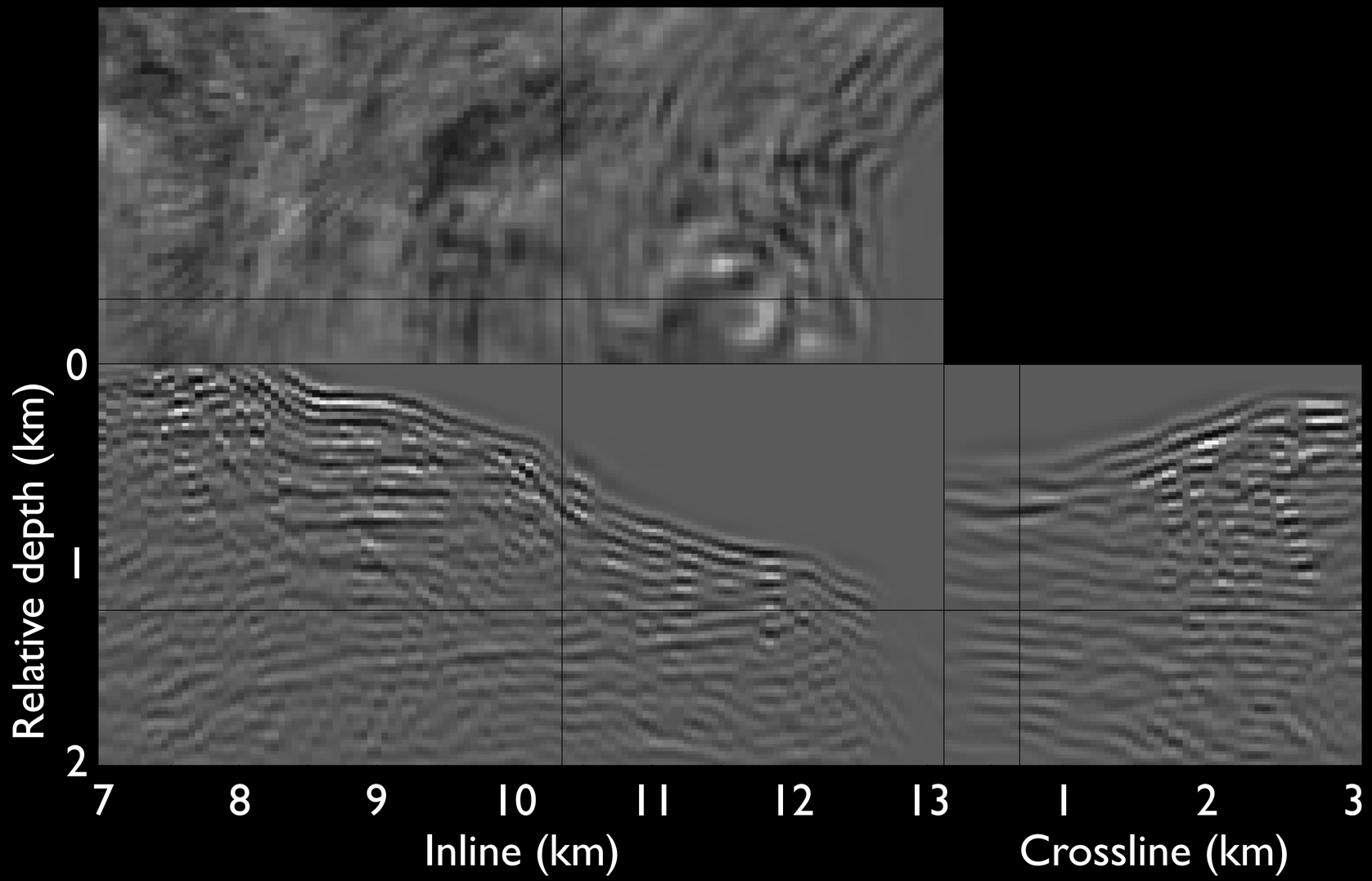
Inversion preconditioned with strong dip filtering



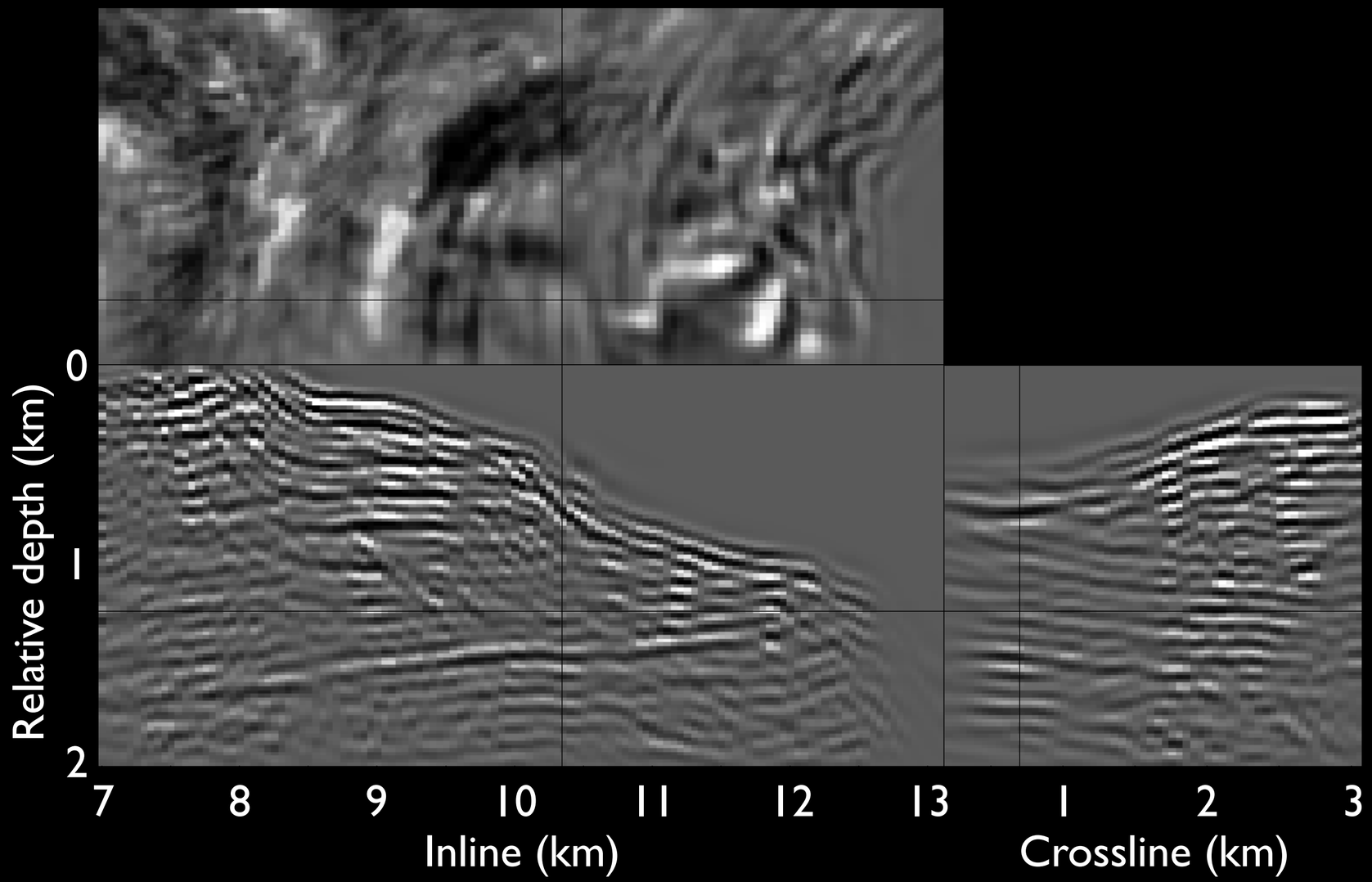
Inversion without regularization



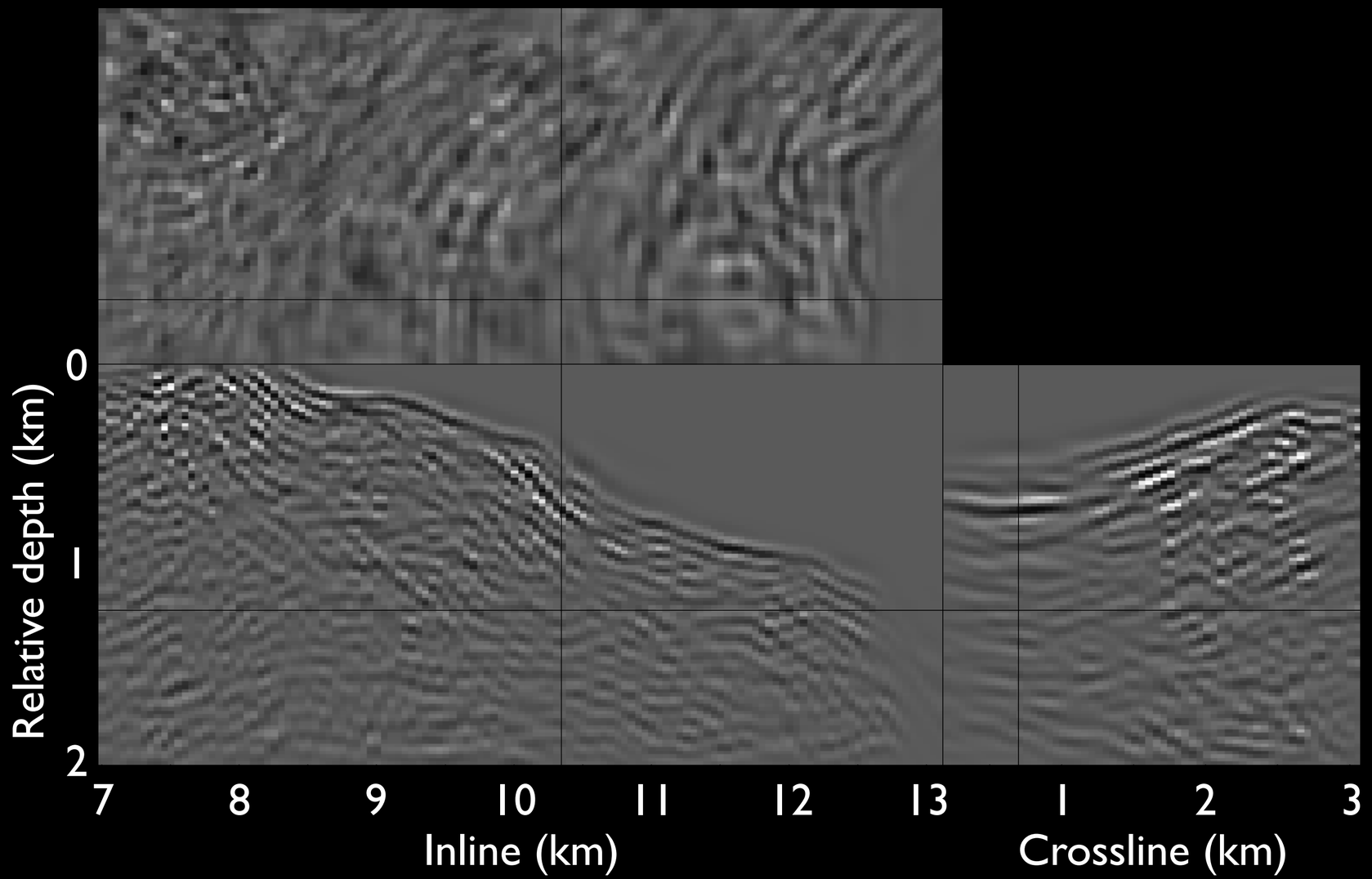
Inversion with damping ($\varepsilon = 0.02$)



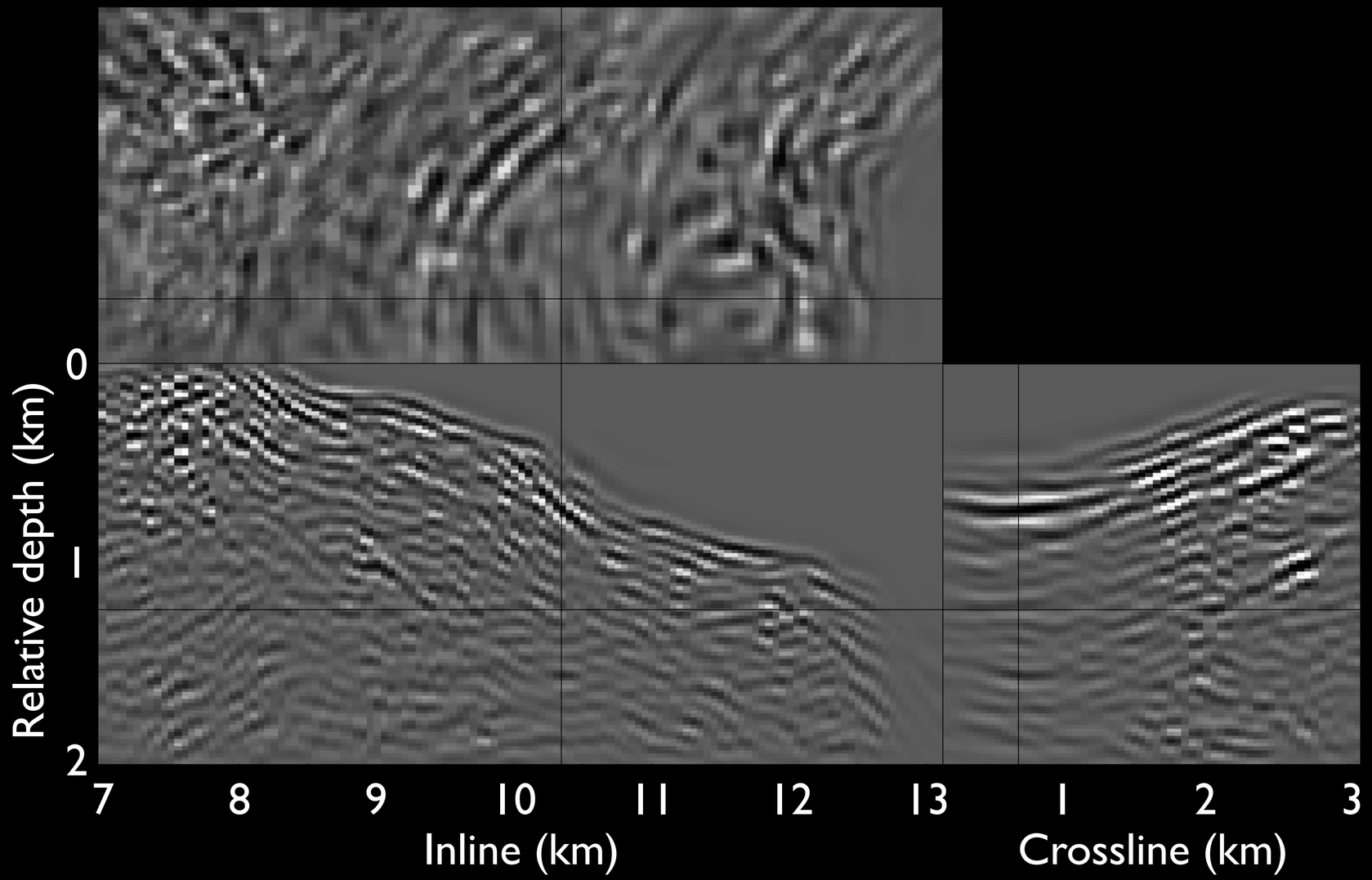
Inversion with damping ($\varepsilon = 0.05$)



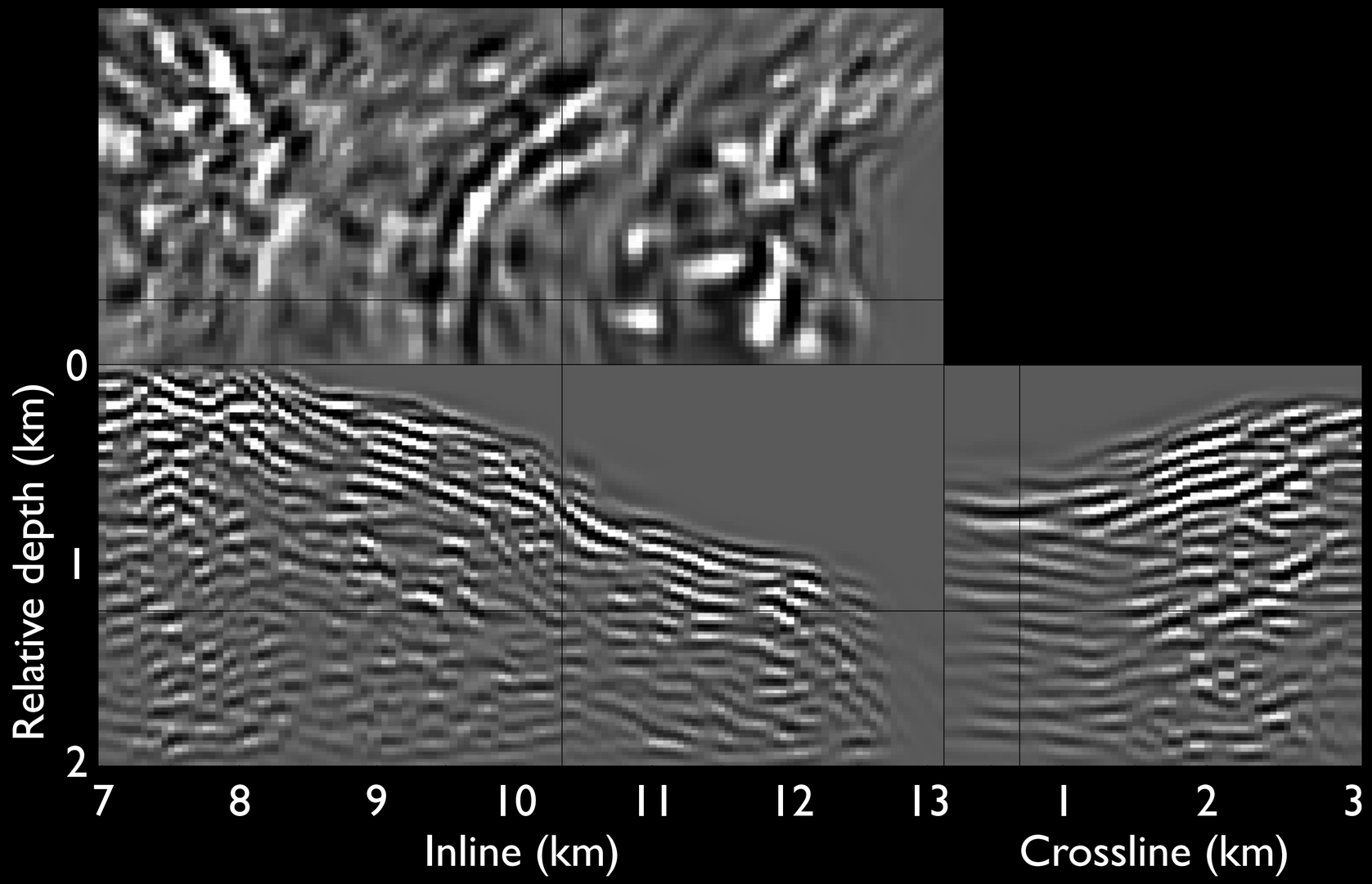
Inversion preconditioned with weak dip filtering



Inversion preconditioned with moderate dip filtering

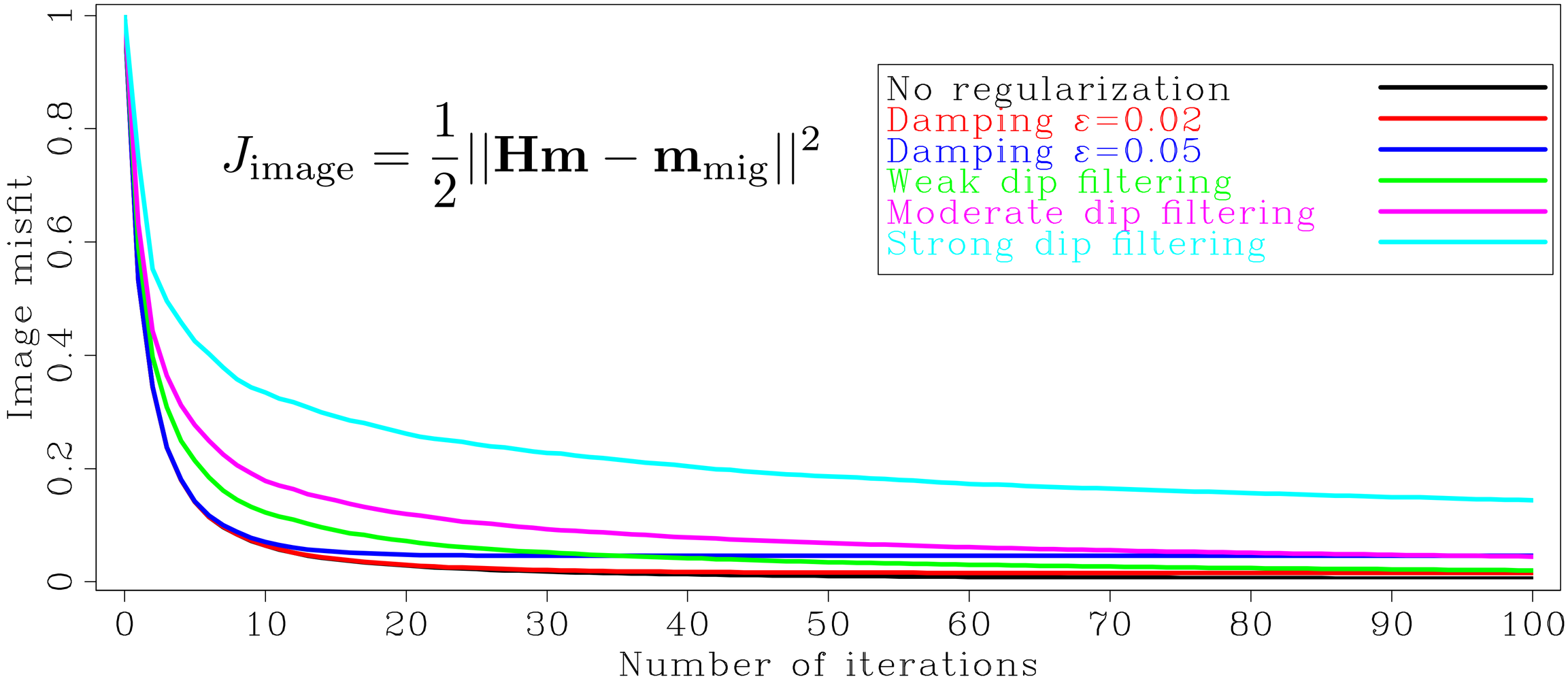


Inversion preconditioned with strong dip filtering

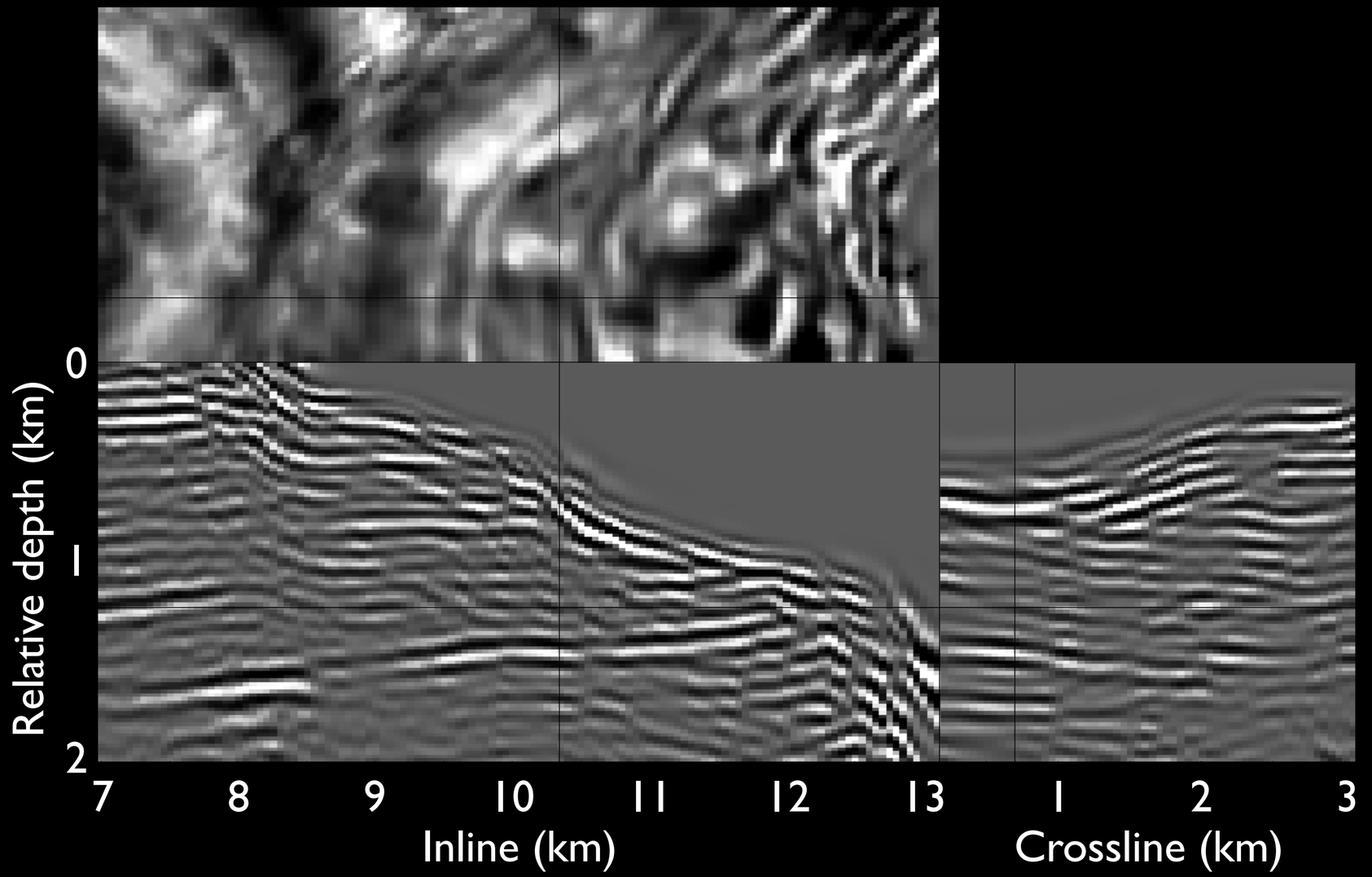


Convergence of image misfit

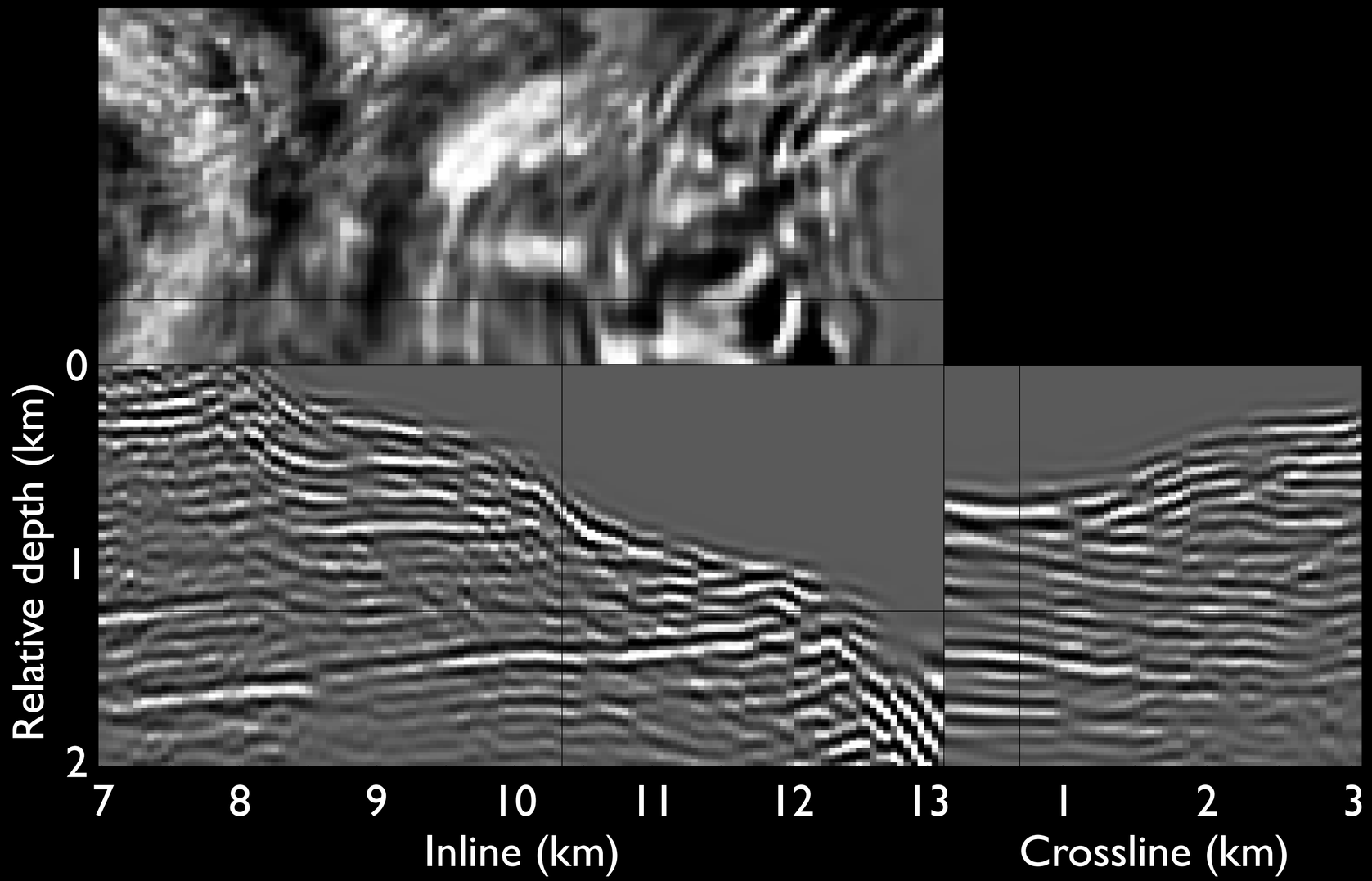
$$J_{\text{image}} = \frac{1}{2}||\mathbf{Hm} - \mathbf{m}_{\text{mig}}||^2$$



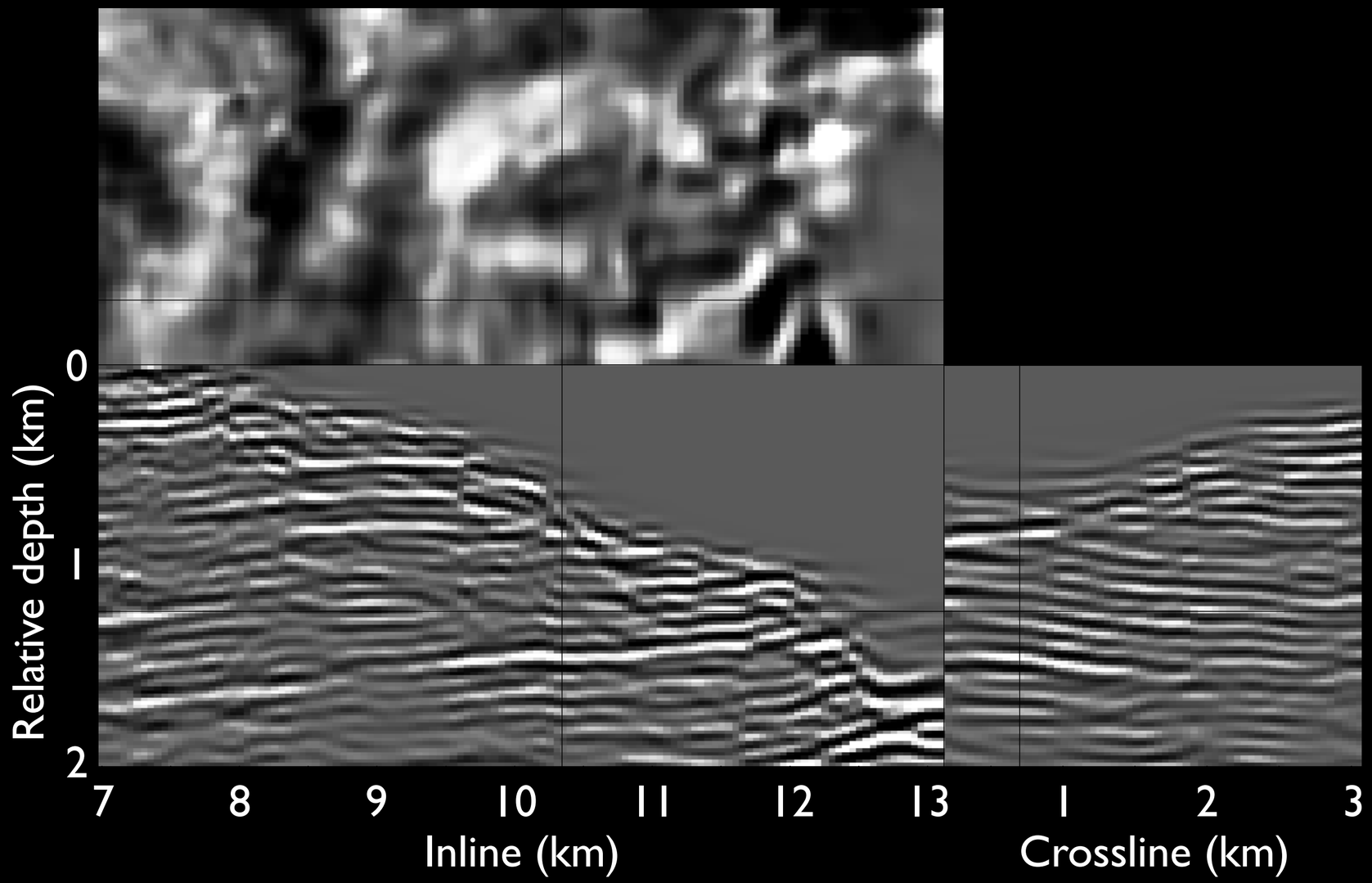
Migration with AGC



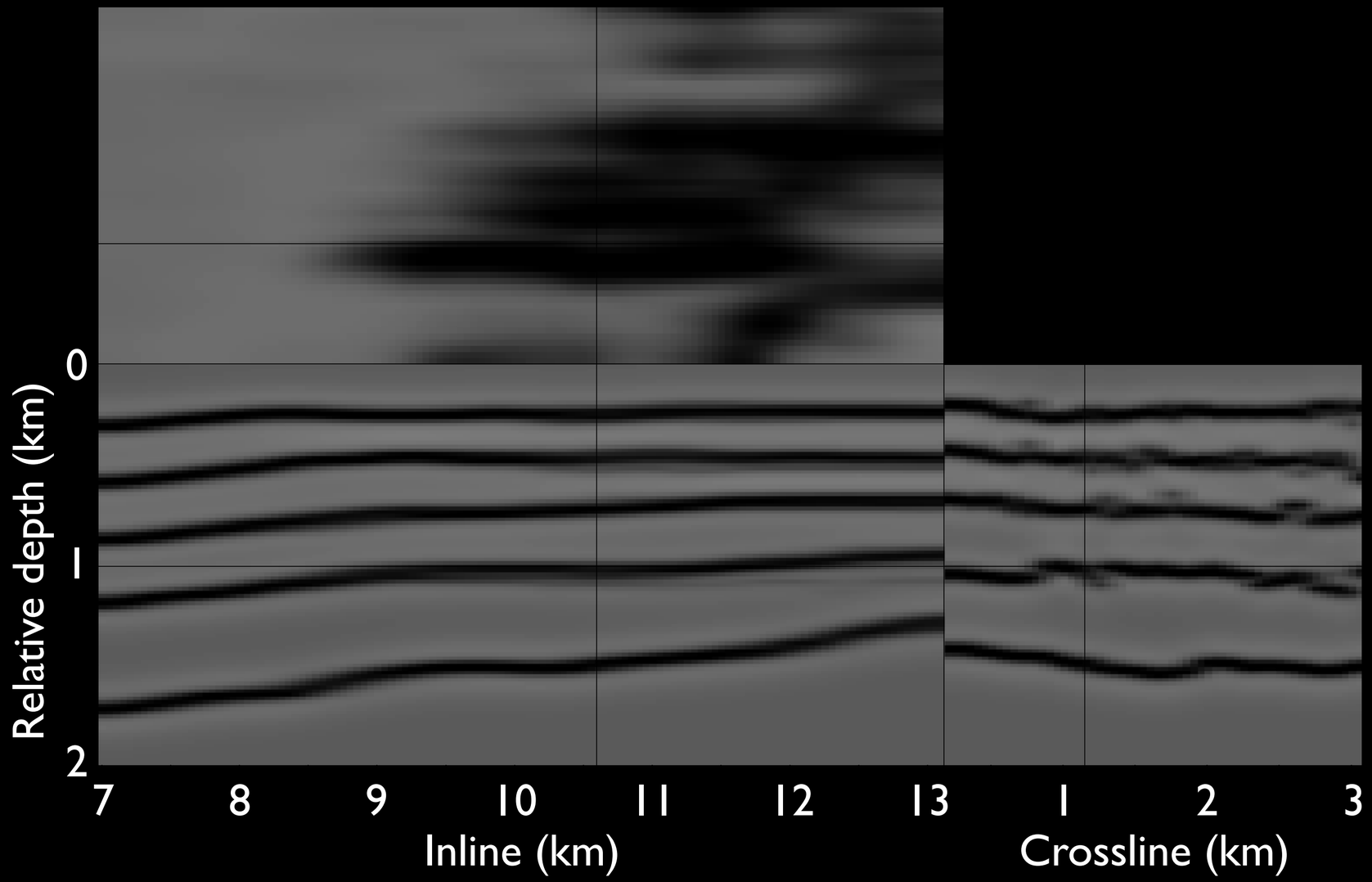
Damped inversion with AGC ($\varepsilon = 0.02$)



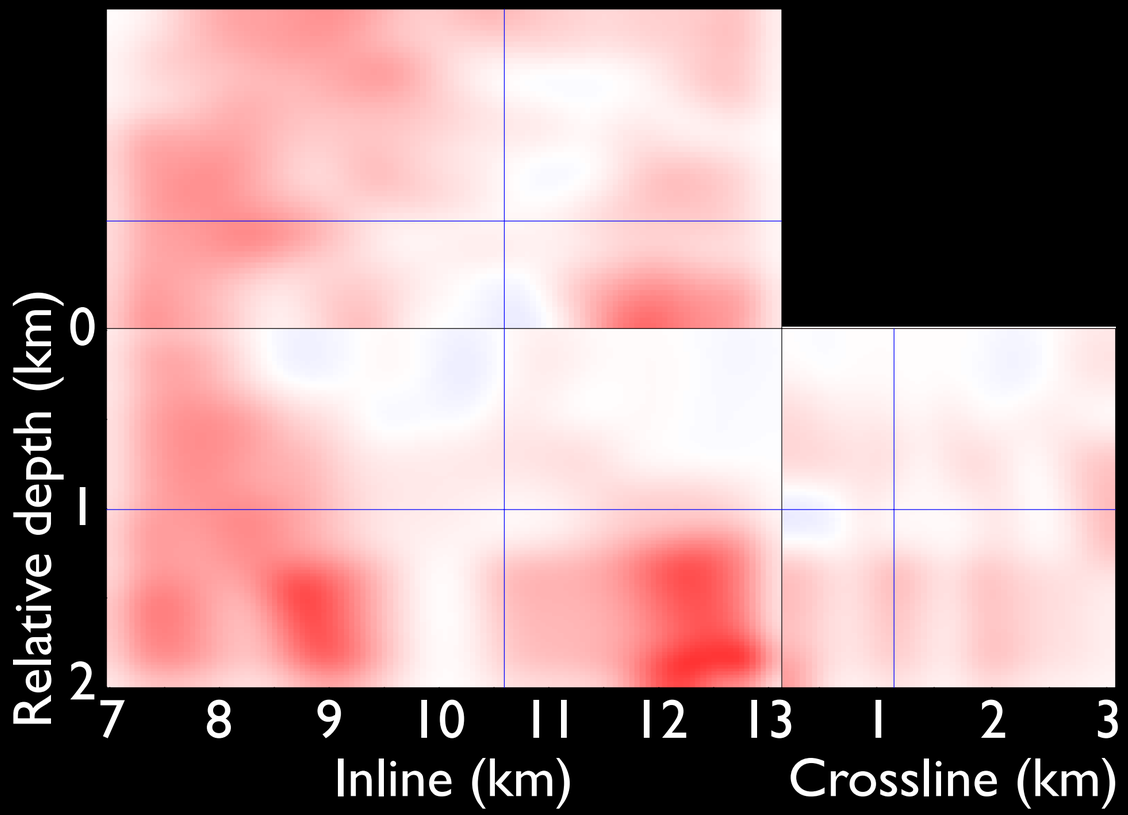
Dip-constrained inversion with AGC (weak smoothing)

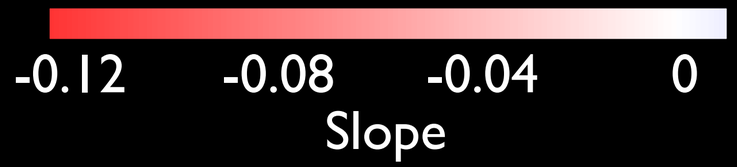


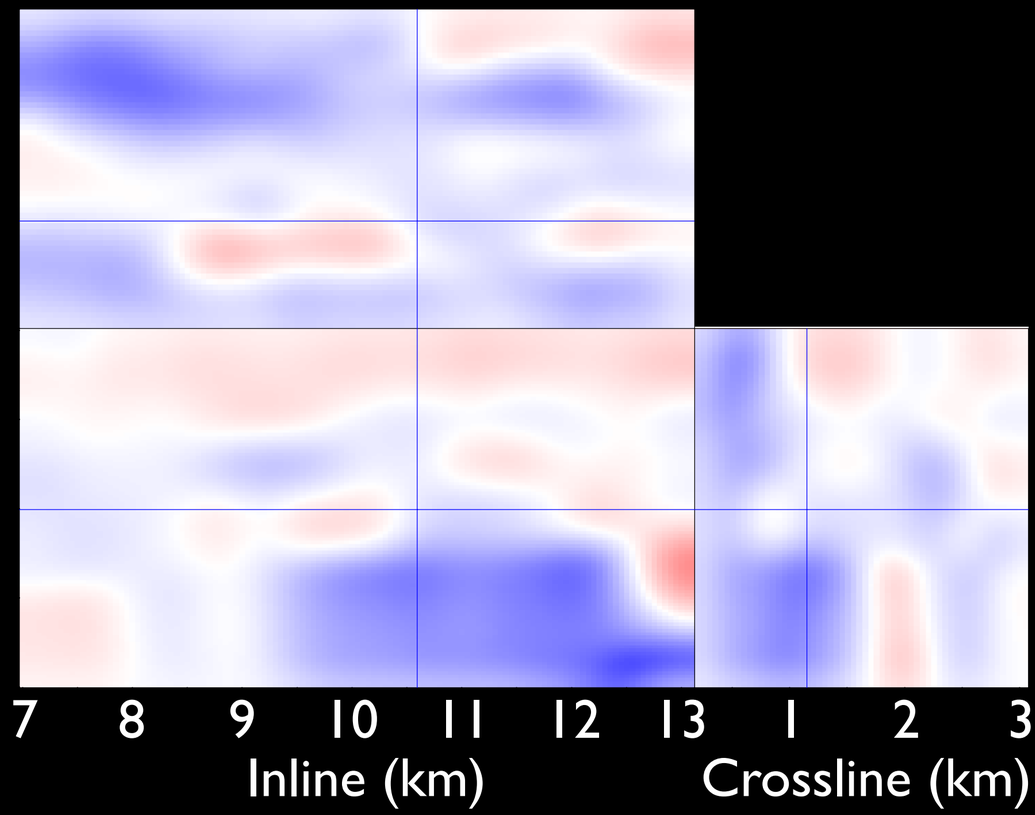
Interpreted horizons

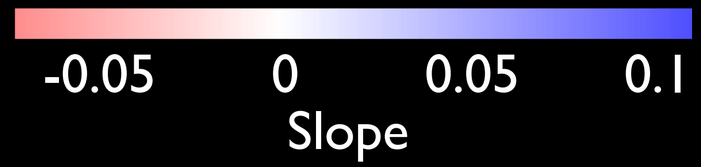


Dip fields estimated from interpreted horizons

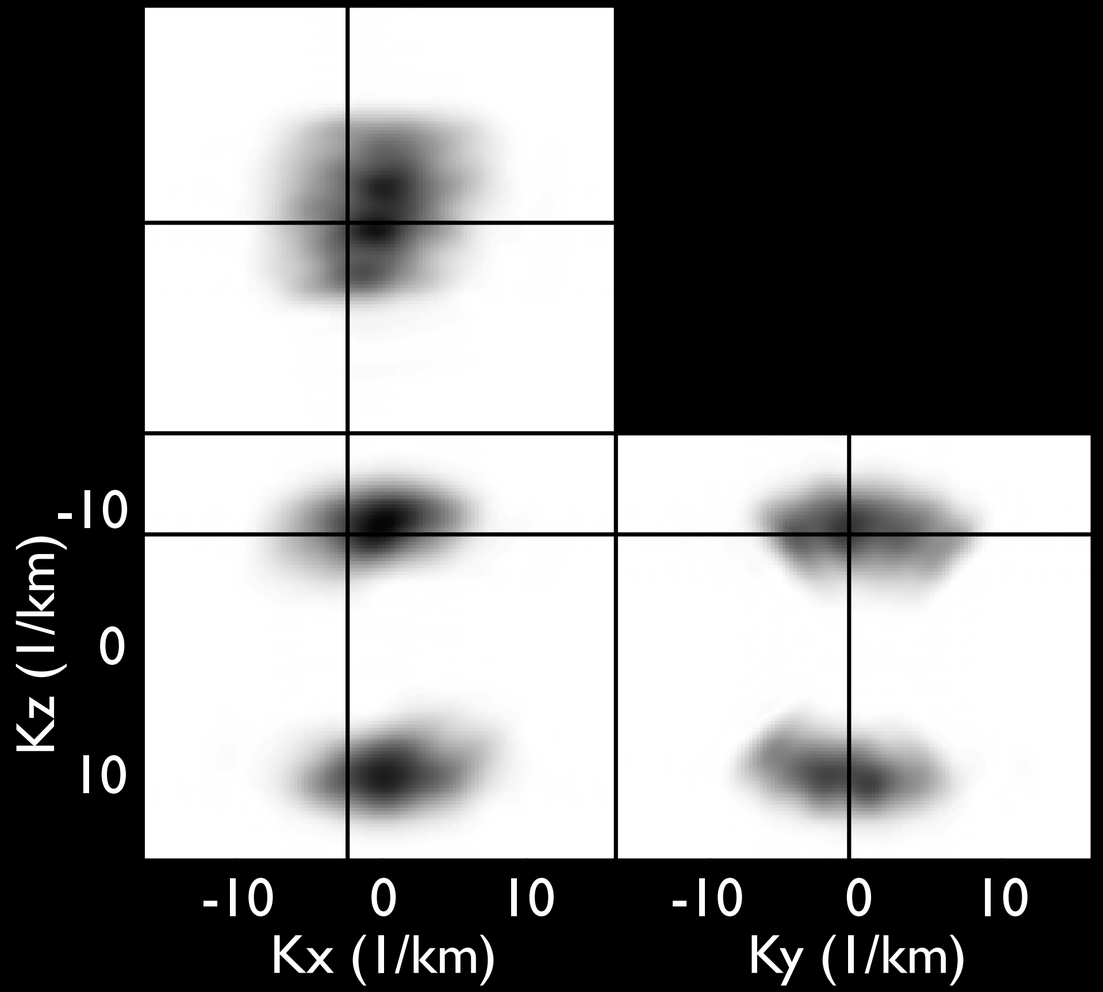


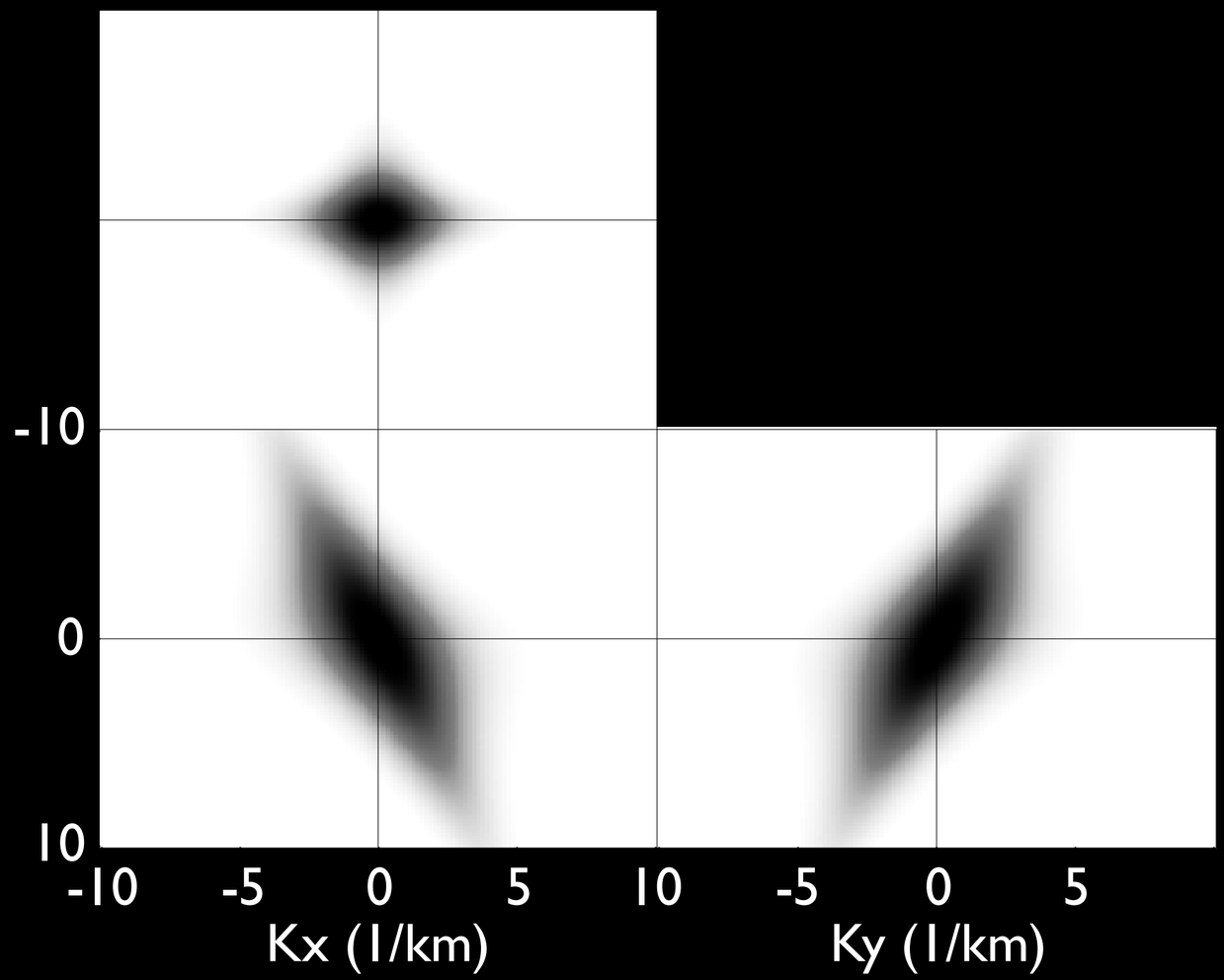




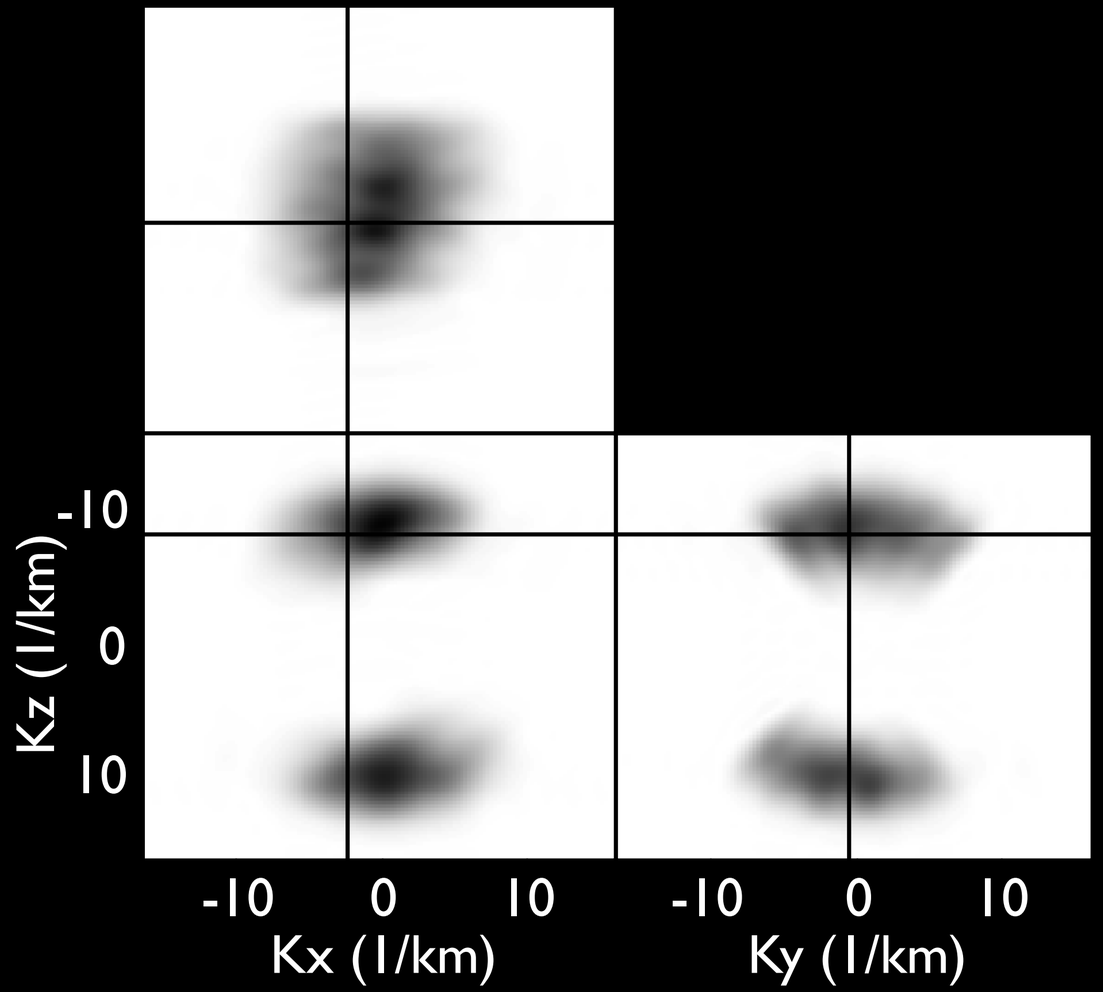


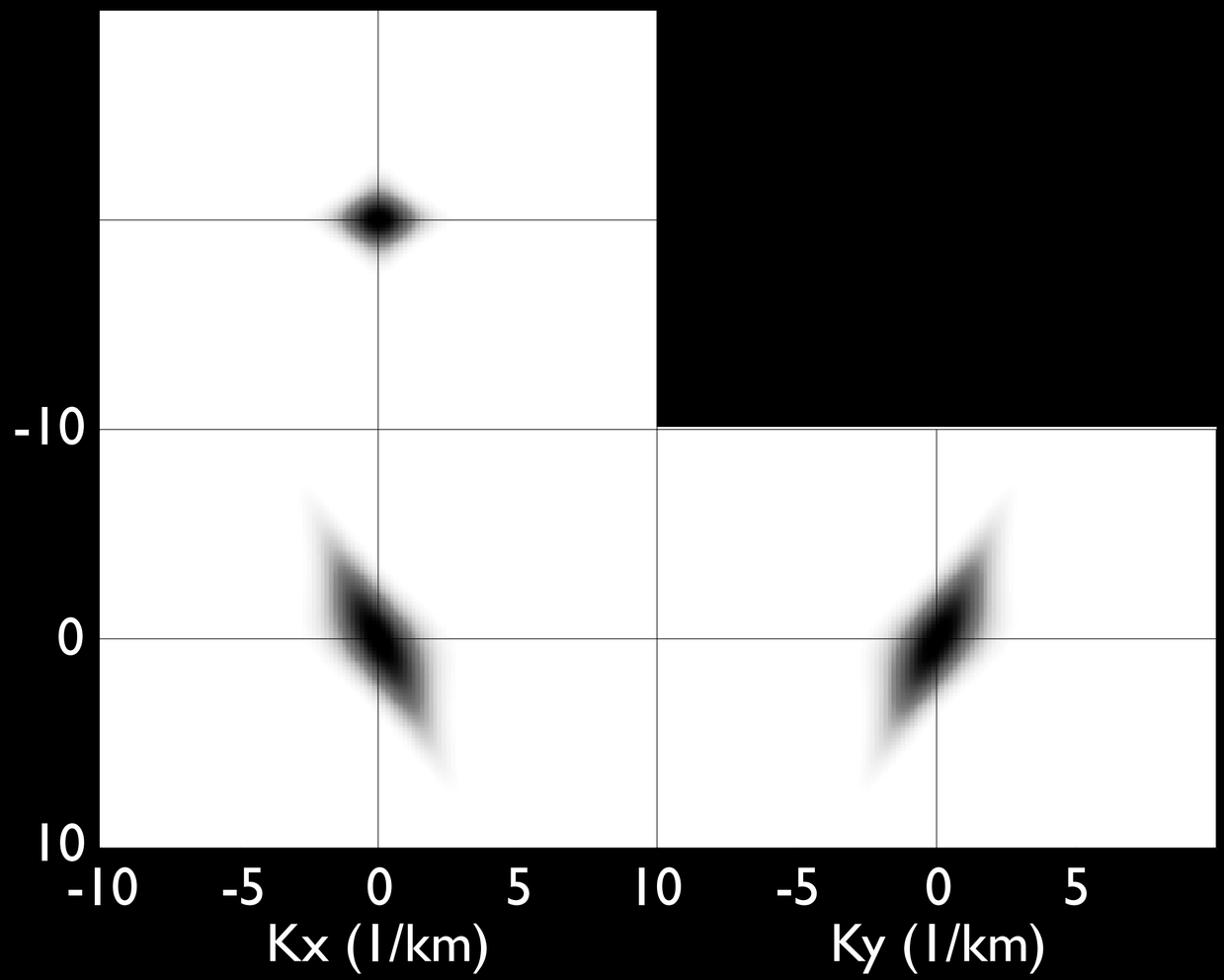
Dip filter spectrum (weaking smoothing)



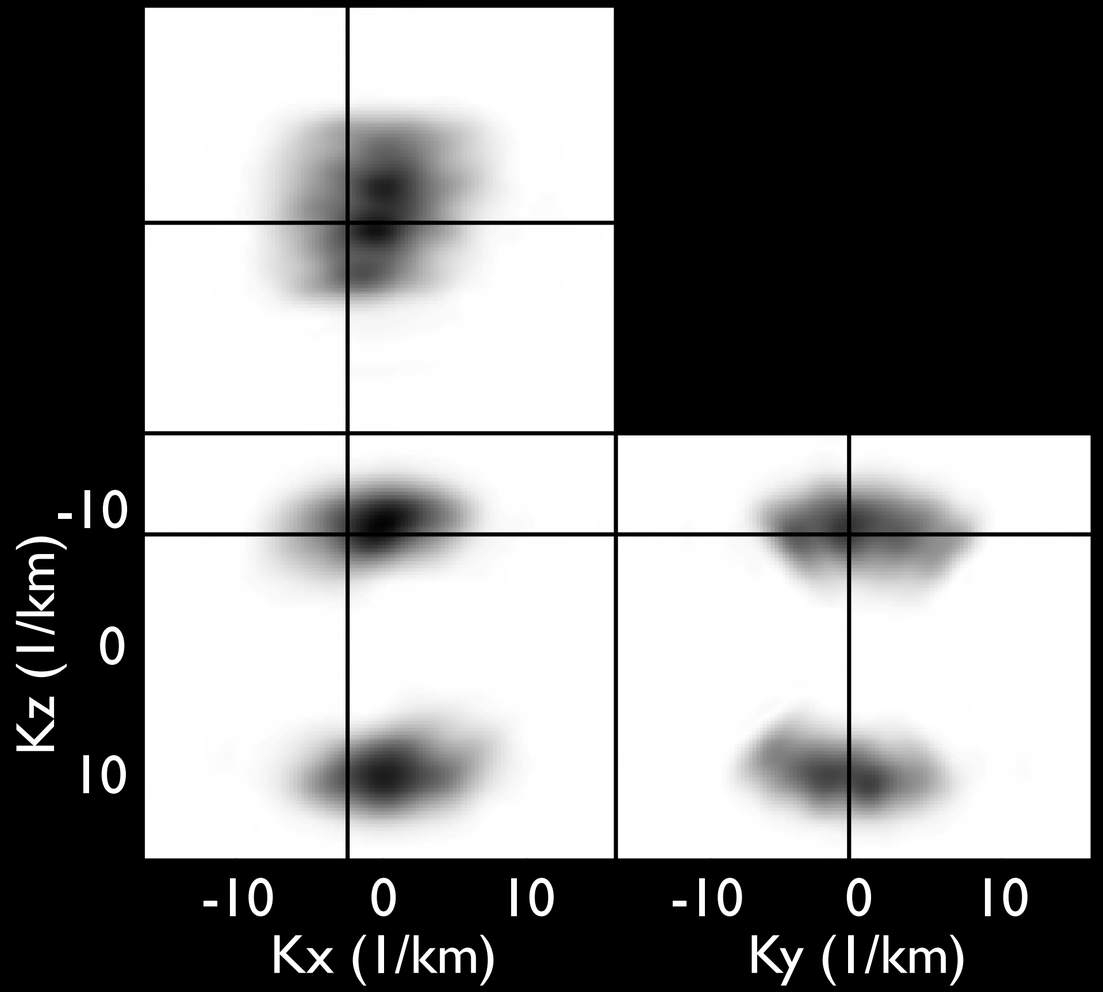


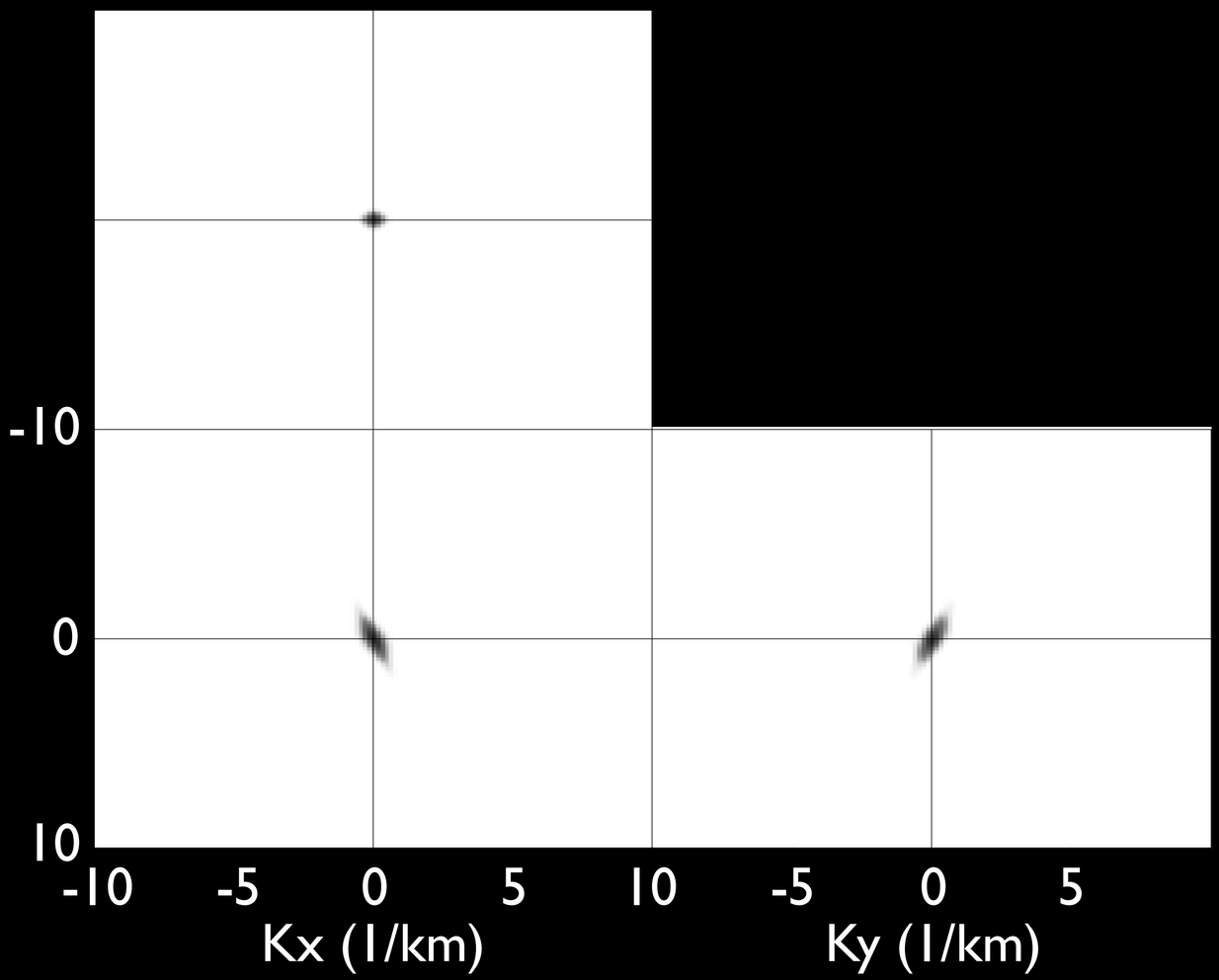
Dip filter spectrum (moderate smoothing)





Dip filter spectrum (strong smoothing)





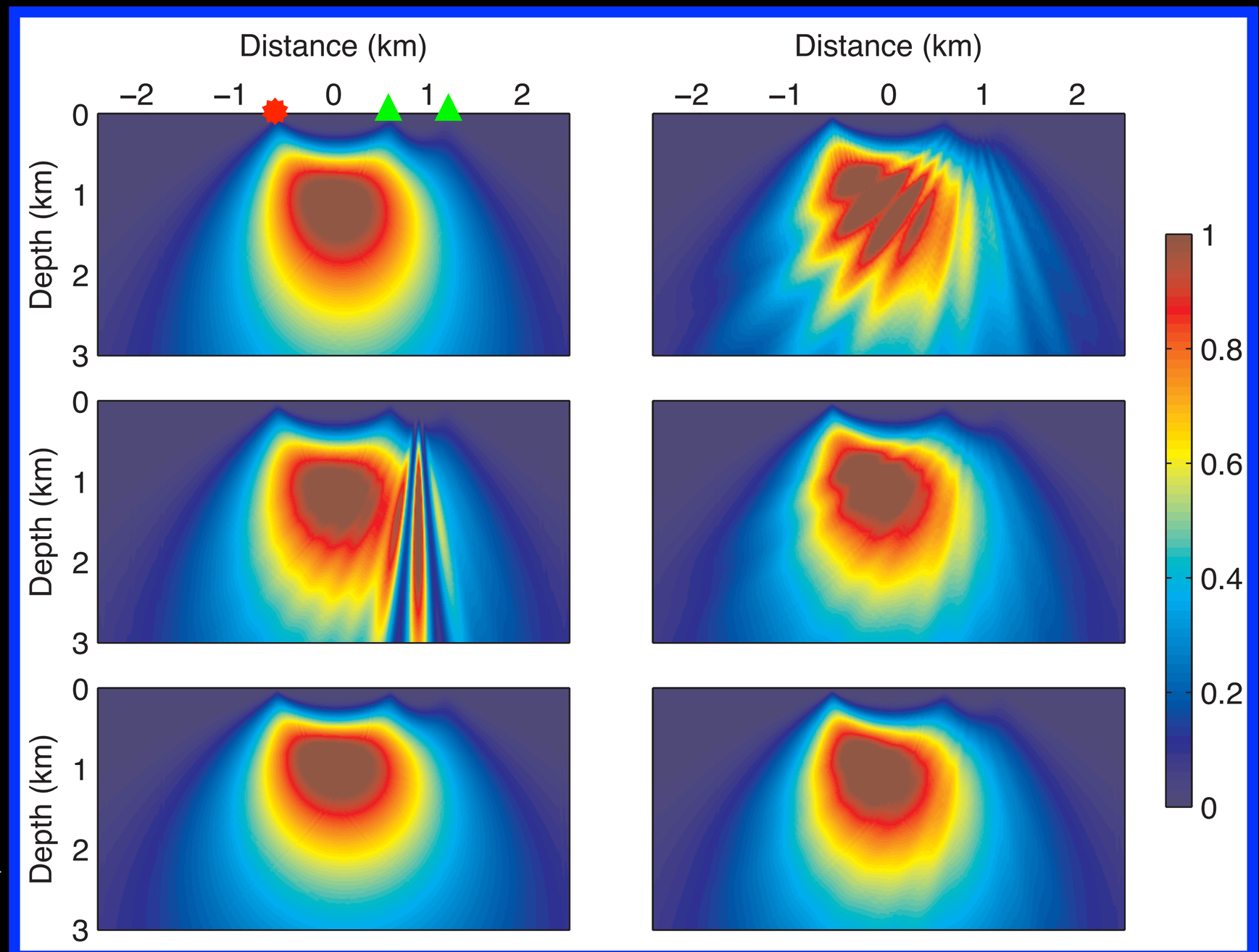
Hessian main diagonal

Exact Hessian

Hessian with crosstalk

Plane-wave encoding

$N_{\text{plane}} = 31$



Random encoding

$N_{\text{realize}} = 1$

$N_{\text{realize}} = 5$

$N_{\text{realize}} = 20$

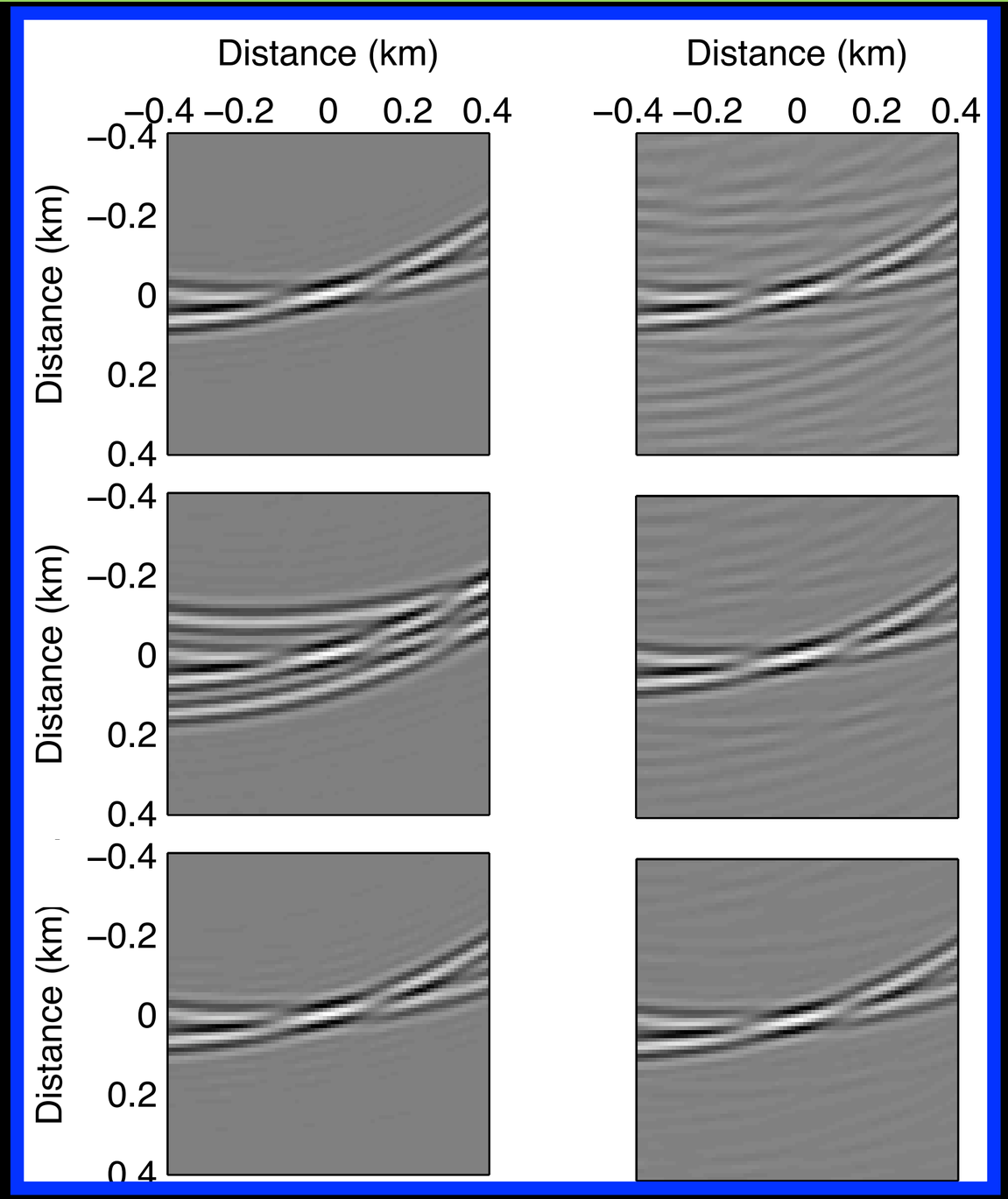
Hessian off-diagonals

Exact Hessian

Hessian with crosstalk

Plane-wave encoding

$N_{\text{plane}} = 31$



Random encoding

$N_{\text{realize}} = 1$

$N_{\text{realize}} = 5$

$N_{\text{realize}} = 20$

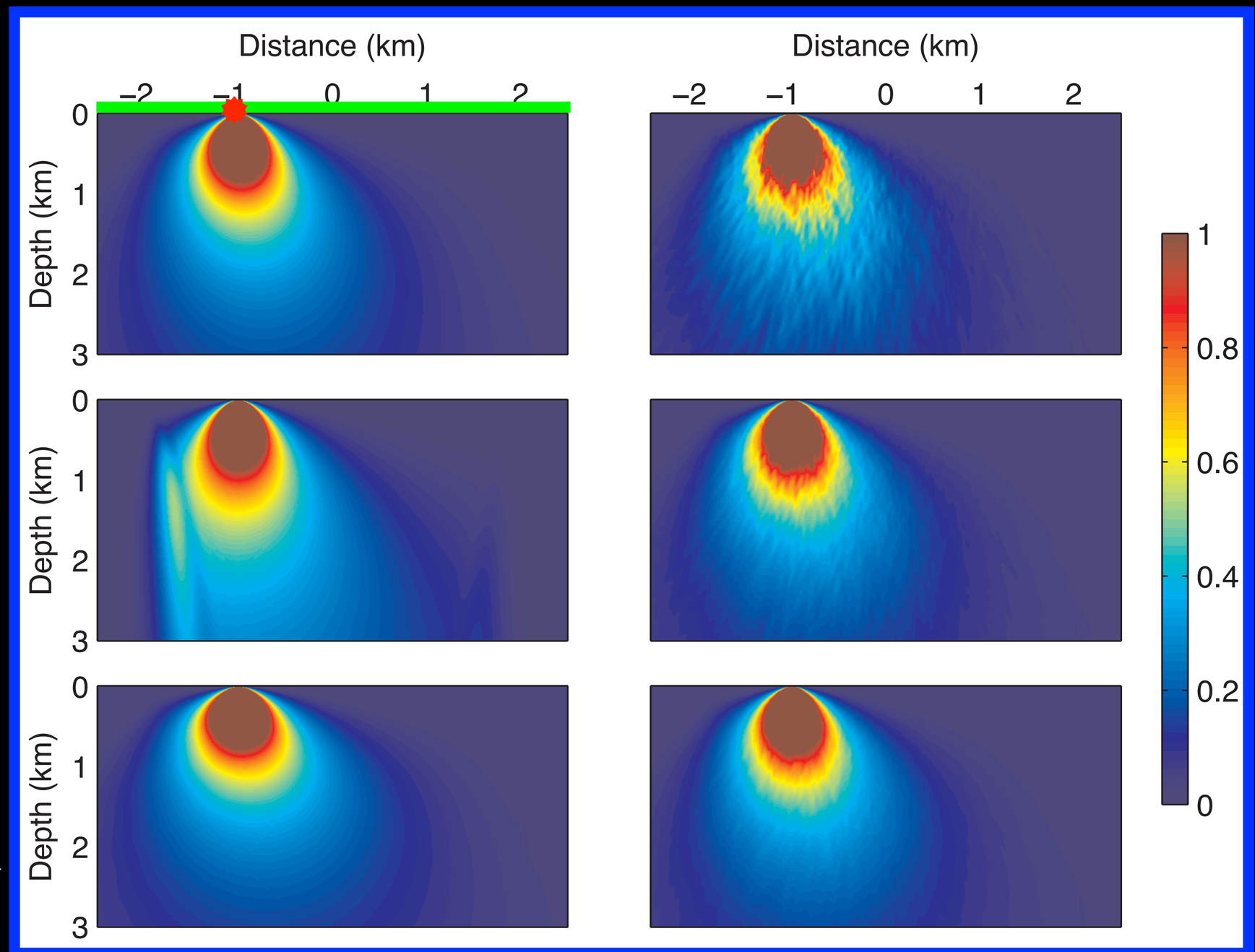
Hessian main diagonal

Exact Hessian

Hessian with crosstalk

Plane-wave encoding

$N_{\text{plane}} = 31$



Random encoding

$N_{\text{realize}} = 1$

$N_{\text{realize}} = 5$

$N_{\text{realize}} = 20$

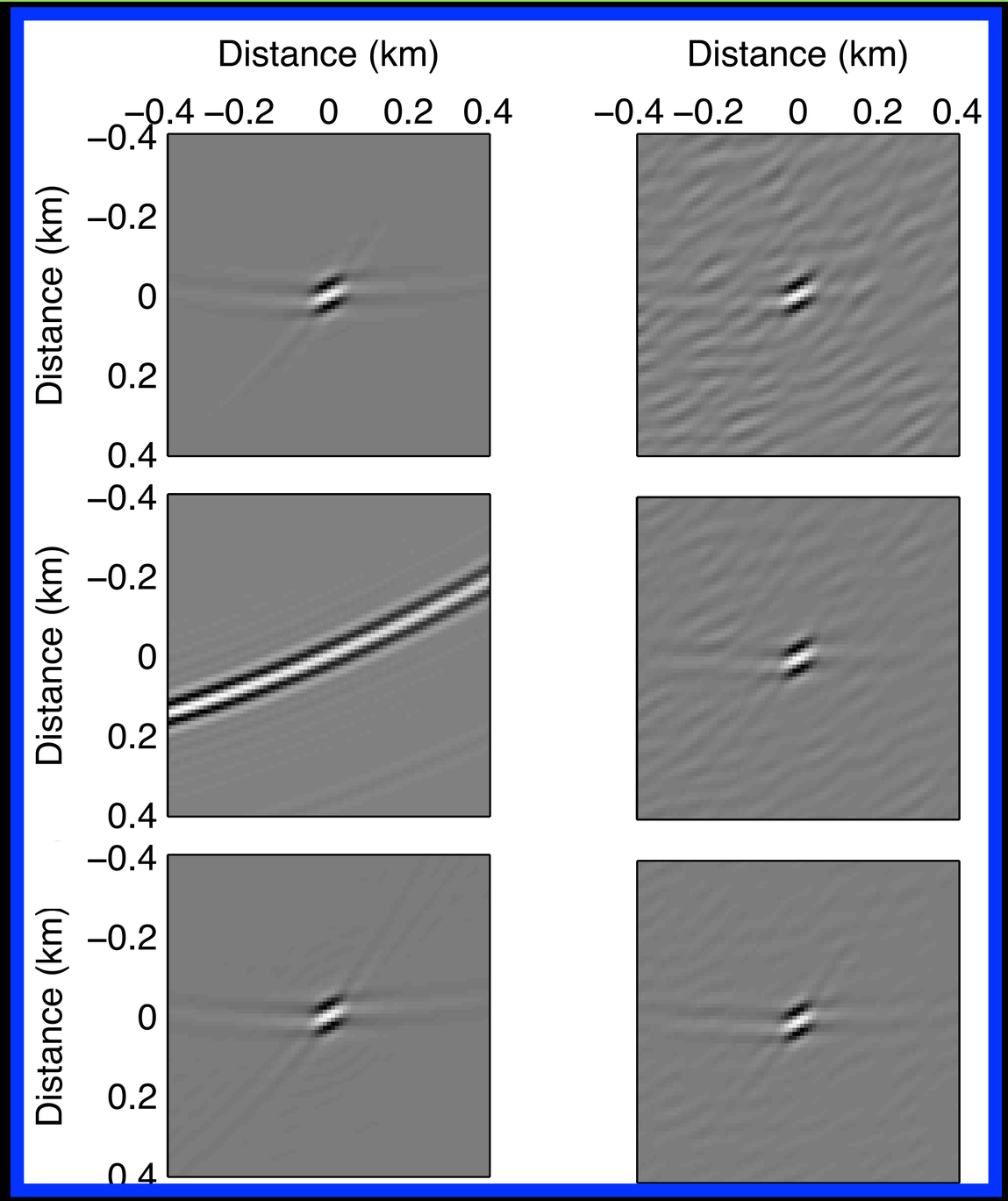
Hessian off-diagonals

Exact Hessian

Hessian with crosstalk

Plane-wave encoding

$N_{\text{plane}} = 31$



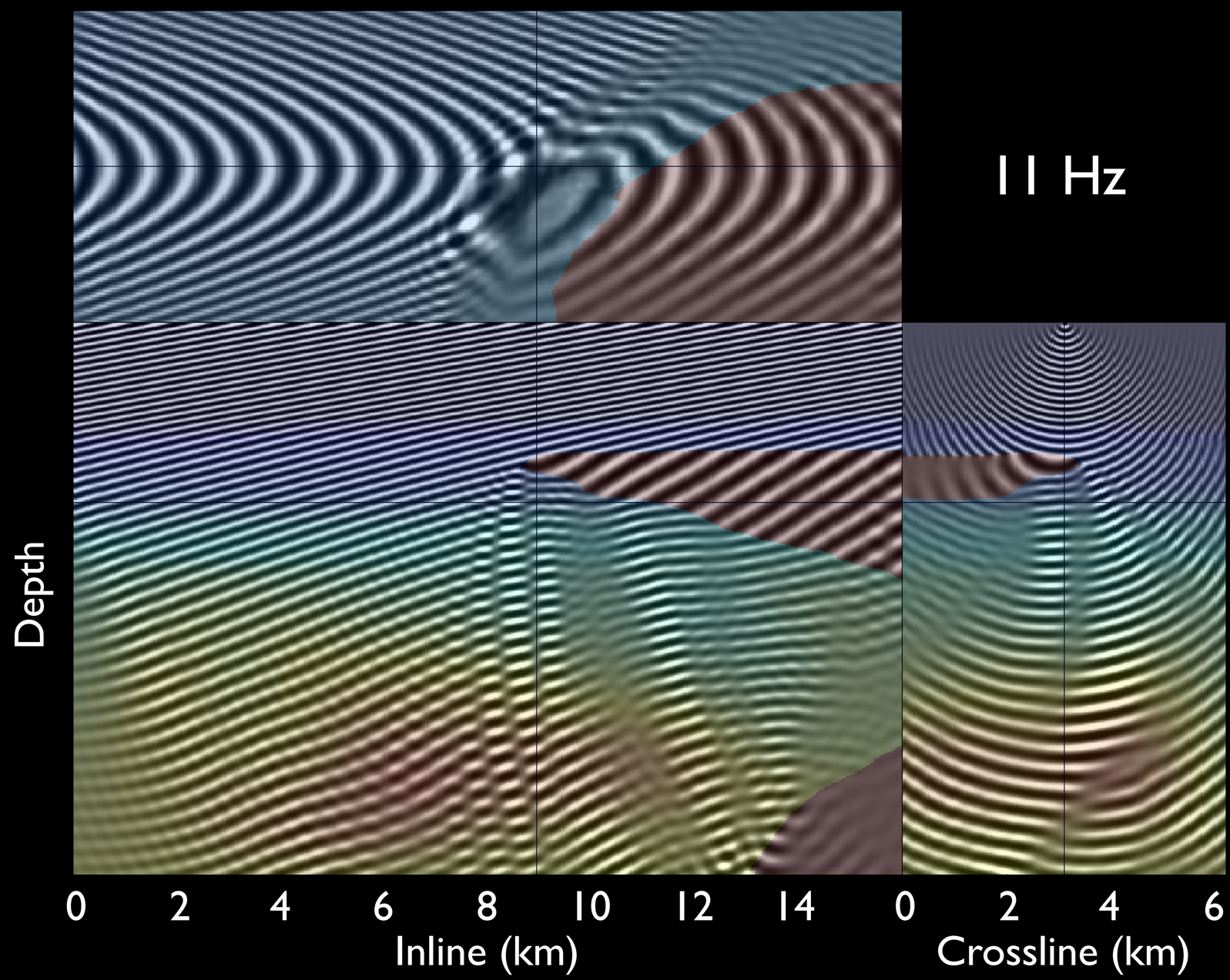
Random encoding

$N_{\text{realize}} = 1$

$N_{\text{realize}} = 5$

$N_{\text{realize}} = 20$

Monochromatic encoded source-side Green's function



Monochromatic encoded receiver-side Green's function