Estimating an image of Galilee

Jon Claerbout and Antoine Guitton

Estimating an image of Galilee

- Inverse theory says data is imperfect and should be understood to have additive noise.
- In practice I find the modeling less perfect than the data.
- The Galilee data set illustrates the usual case, that the data requires a more complicated model.

Image estimation is nontrivial inversion

- Largest easily invertible matrix has 10,000 columns.
- A small image (say 100x100) has 10,000 values.

Conclusion:

- Image estimation requires iterative methods.
- Speed of convergence is a significant issue.

These complicated issues mostly ignored here today.

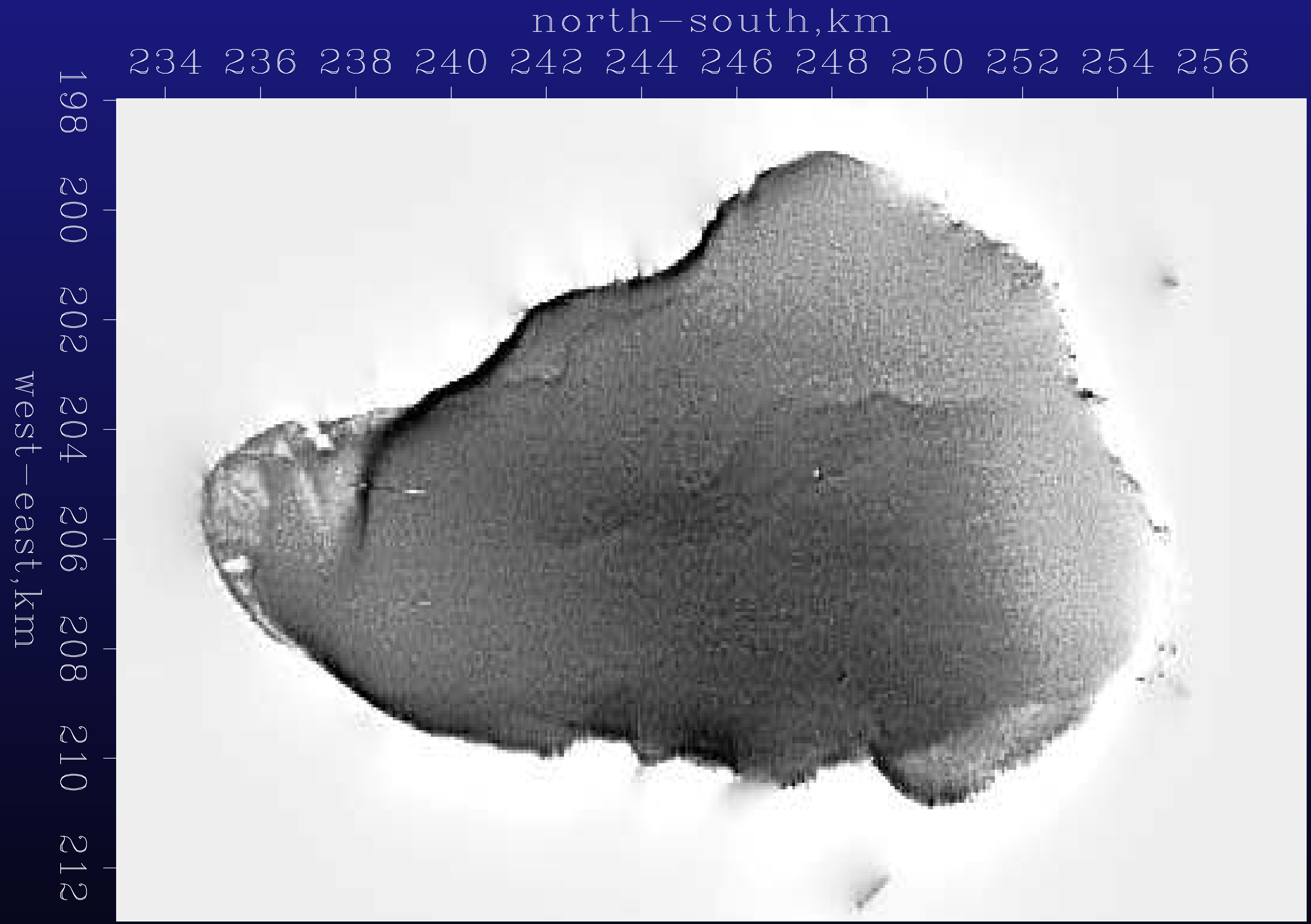
Estimating an image of Galilee

Next three slides

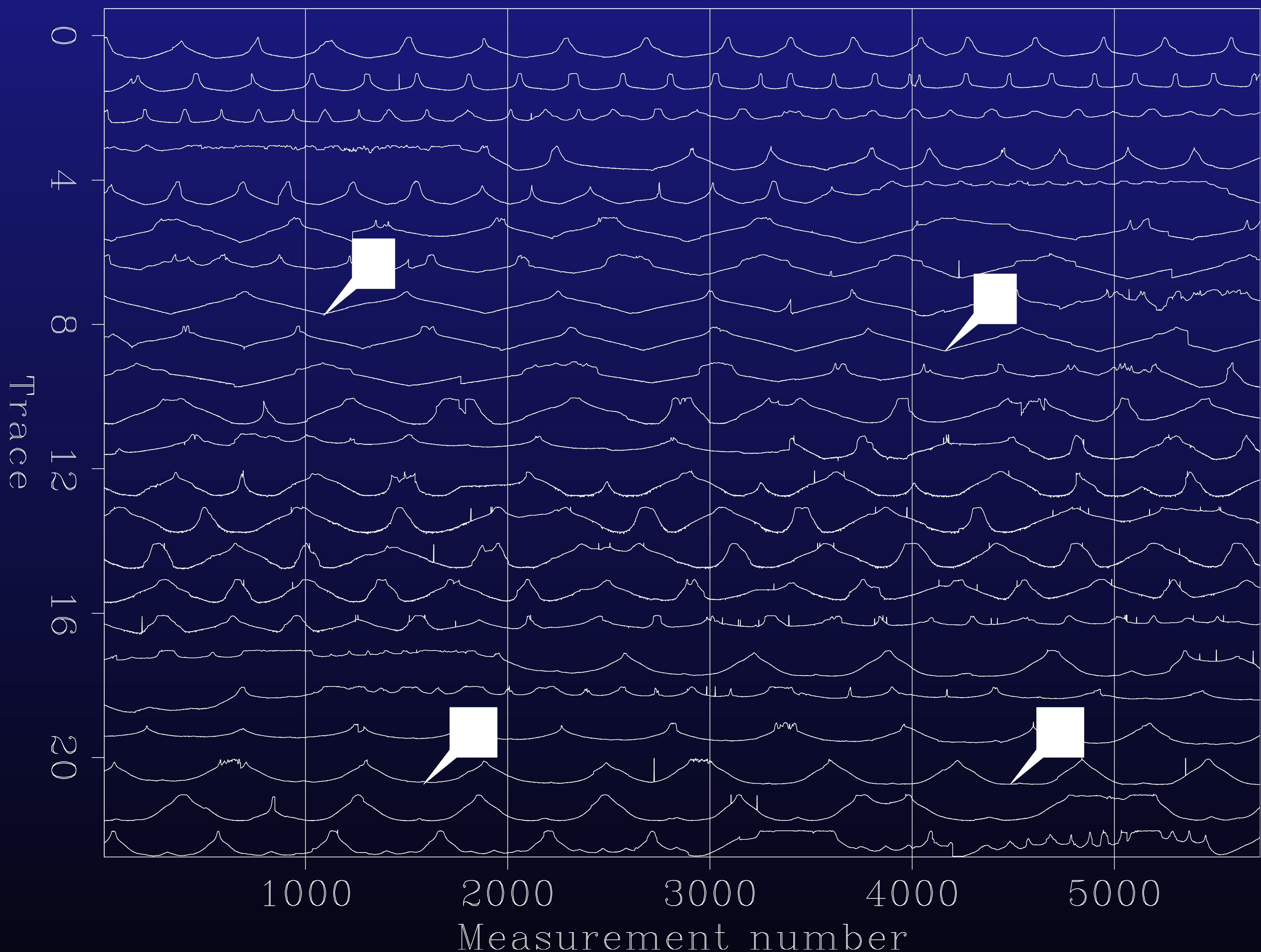
1. Model is depth h(x,y).

2. Data is (x,y,z) at 132,044 locations.

3. Operator (sparse matrix) is the transpose of binning.

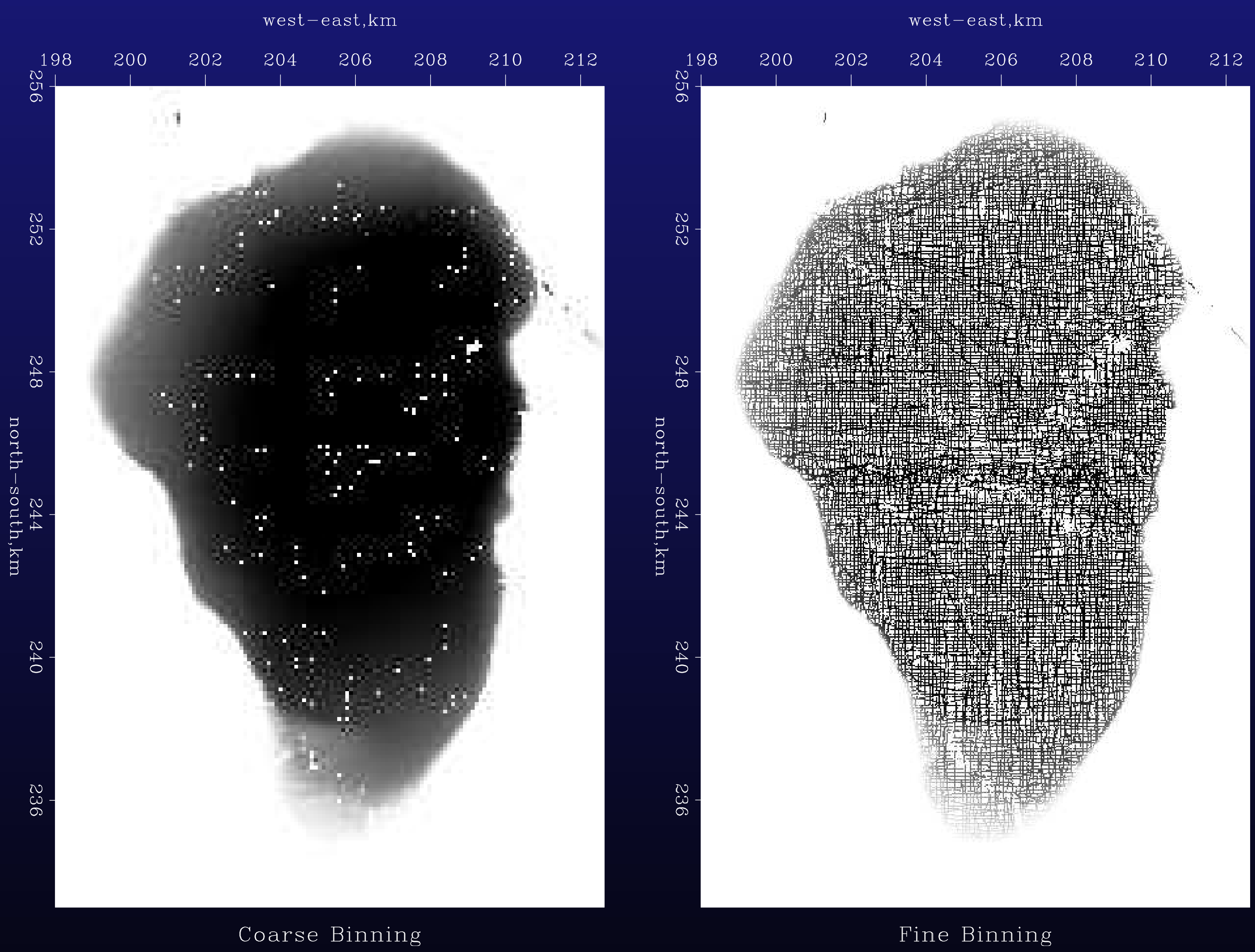


Fitting drift

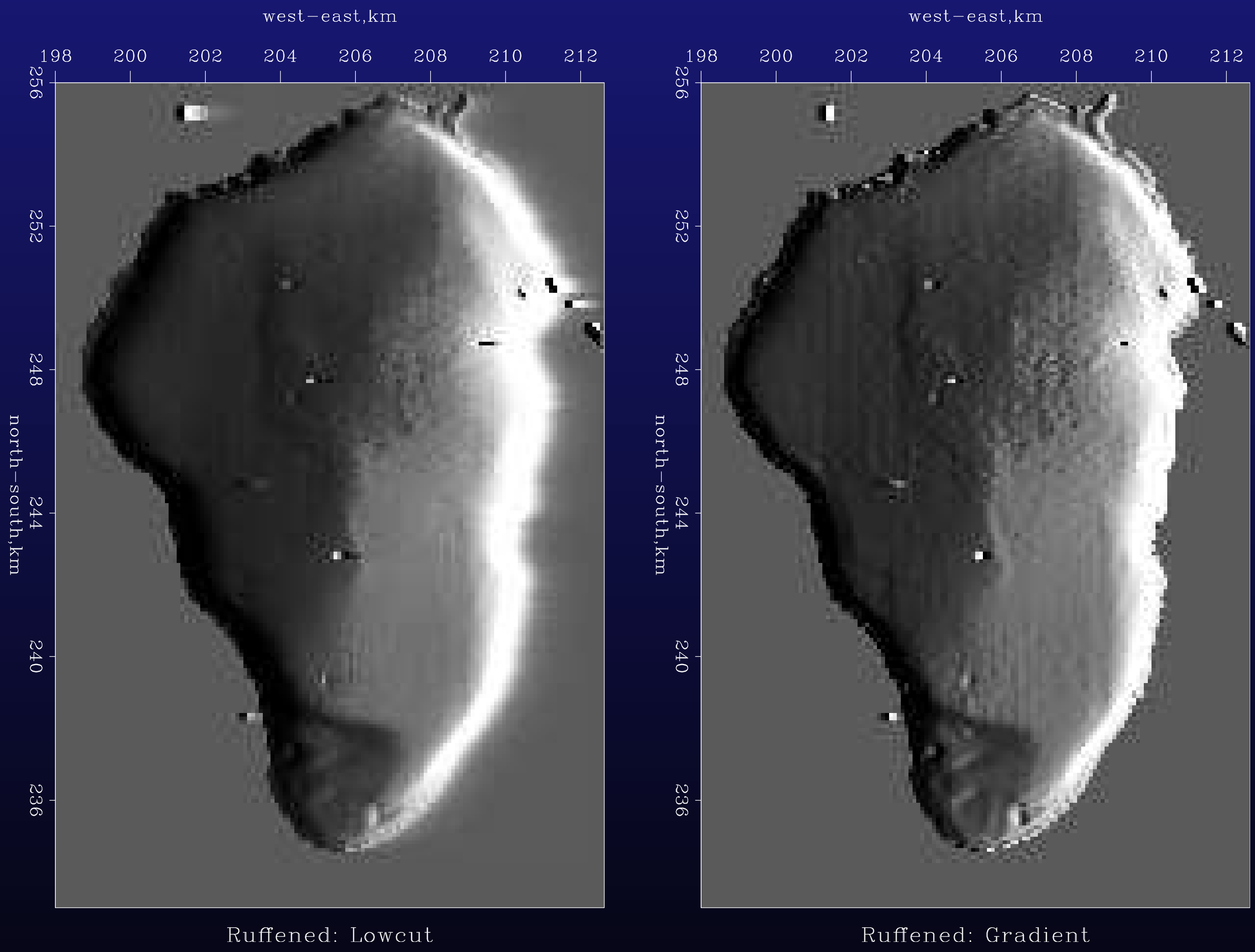

0
4
8
12
16
20
Trace
1000
2000
3000
4000
5000
Measurement number

```
module bin2 {
# Data-push binning in 2-D.
integer :: m1, m2
real    ::  o1,d1,o2,d2
real, dimension (:,:), pointer :: xy
#% _init(     m1,m2, o1,d1,o2,d2,xy)
#% _lop ( mm (m1,m2),  dd (:))
integer    i1,i2, id

do id = 1, size(dd) {

        i1 = 1.5 + (xy(id,1)-o1)/d1
        i2 = 1.5 + (xy(id,2)-o2)/d2
        if( 1<=i1 && i1<=m1 &&
            1<=i2 && i2<=m2   )

                if( transpose )
                        mm(i1,i2) = mm(i1,i2) +  dd( id)
                else
                        dd( id)   = dd( id)   +  mm(i1,i2)

        }

}
```

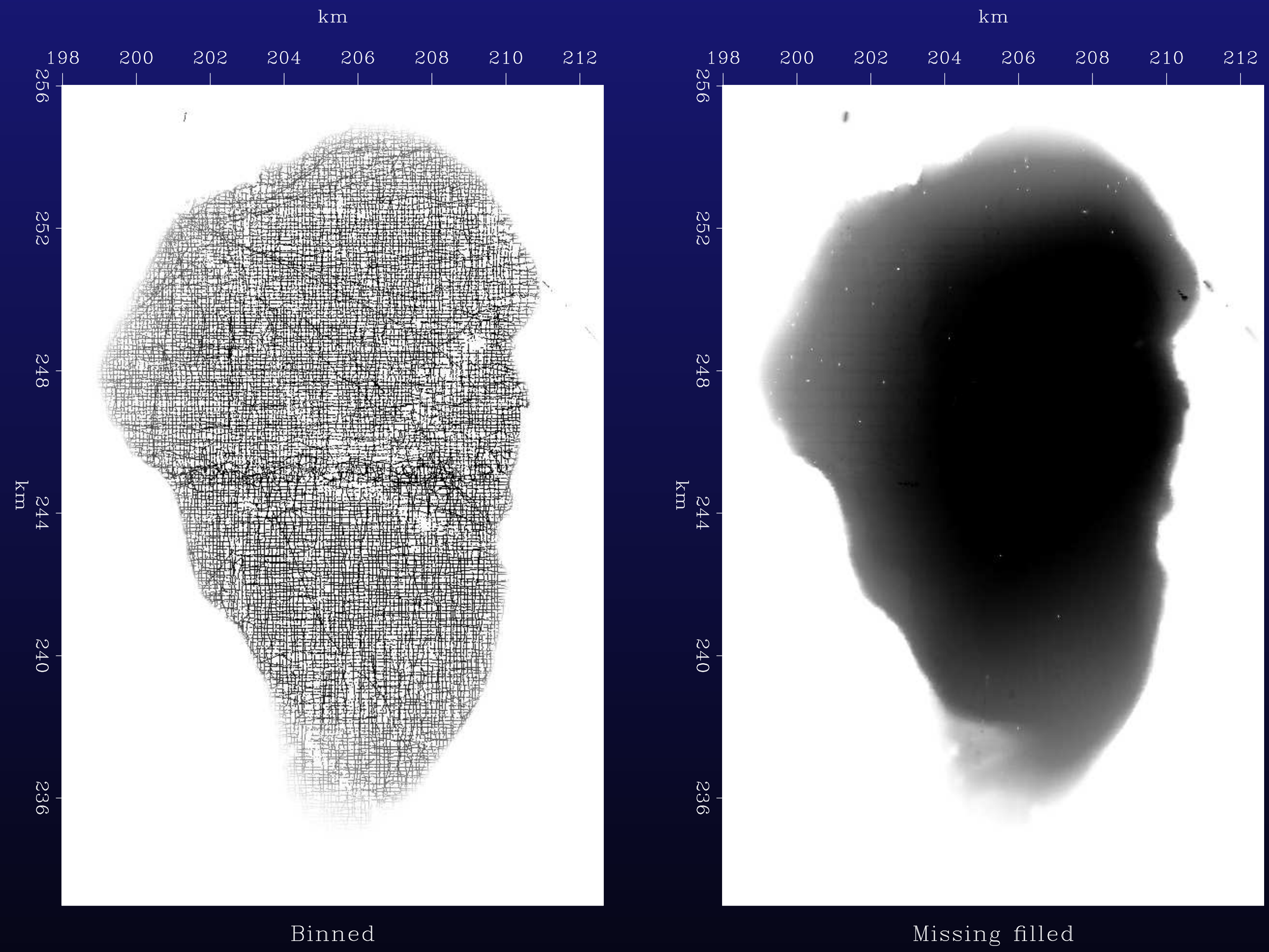
Data binned, coarse and fine



Roughen with east-west derivative



Regularize the empty bins



Depth $h(x,y)$ is a poor variable.

- It is too smooth for viewing convenience.
- Iterative convergence prefers an IID variable.

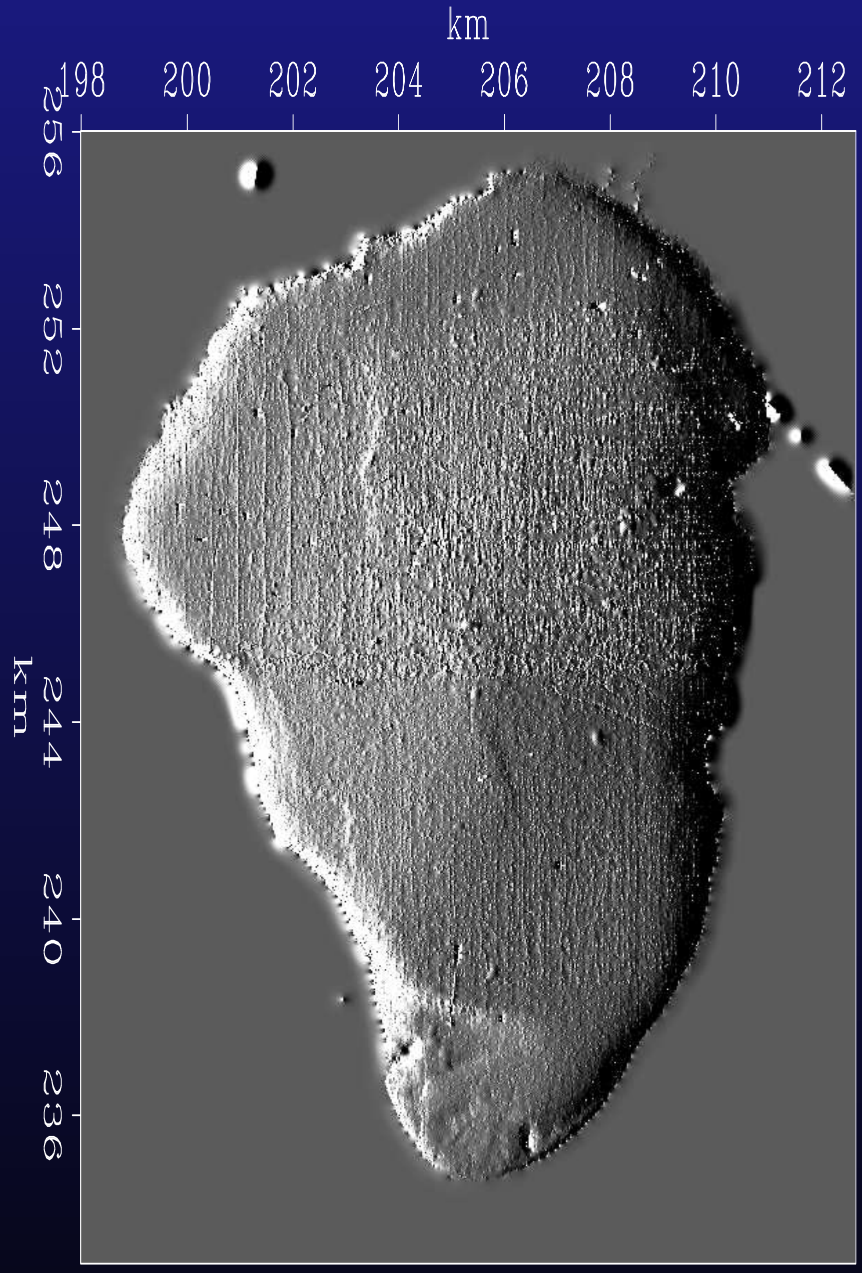
Define a "preconditioned" variable

$$p(x,y) \quad = \quad \text{roughened} \quad h(x,y)$$

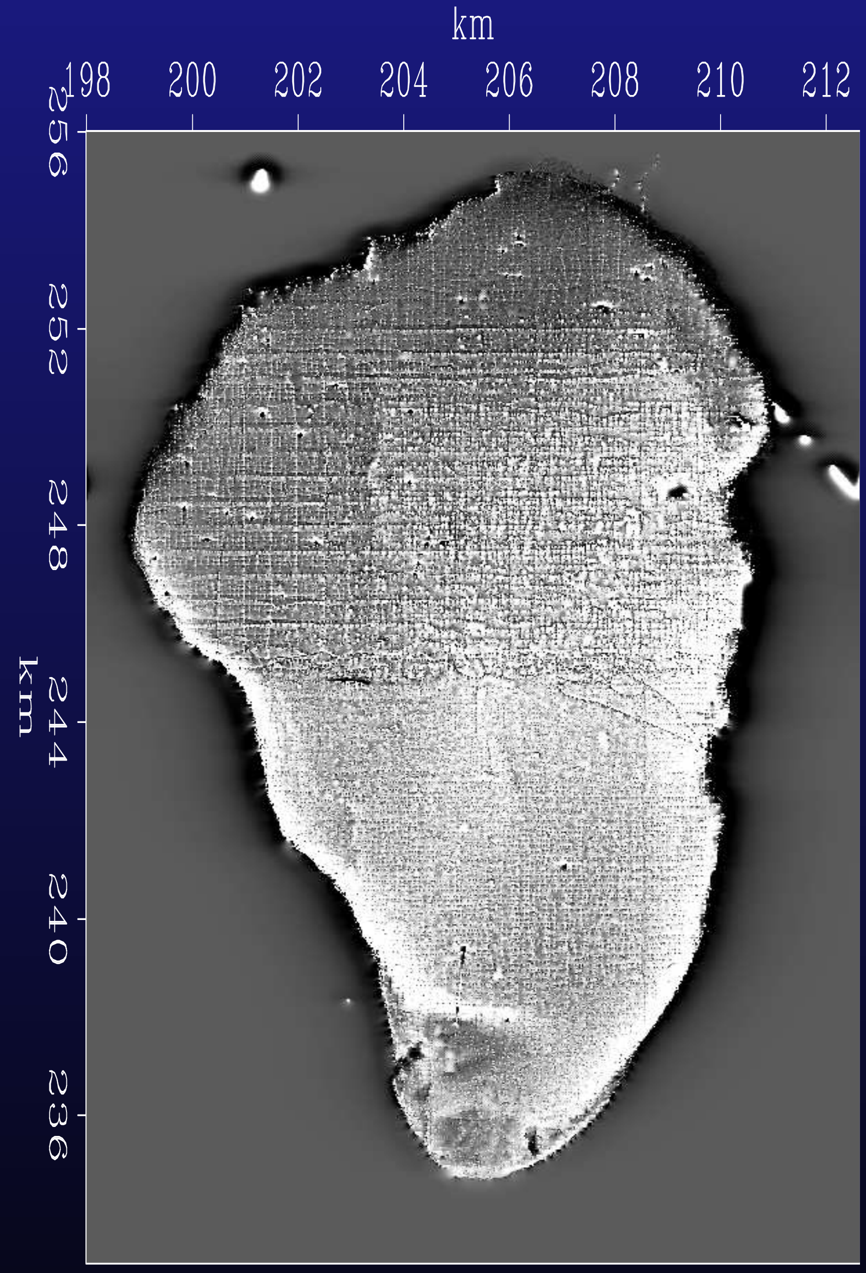
$$\text{roughening filter} = FT\sqrt{k_x^2 + k_y^2}$$

We call it a "helix derivative."

Equivalent to regularizing with the Laplacian.



Filled and d/dx



Filled and helix deriv

Model the drift along the track

The "track axis" is an integer s going with the boat.

Drift (on s axis) is output of random numbers (unknowns) into a low-pass filter.

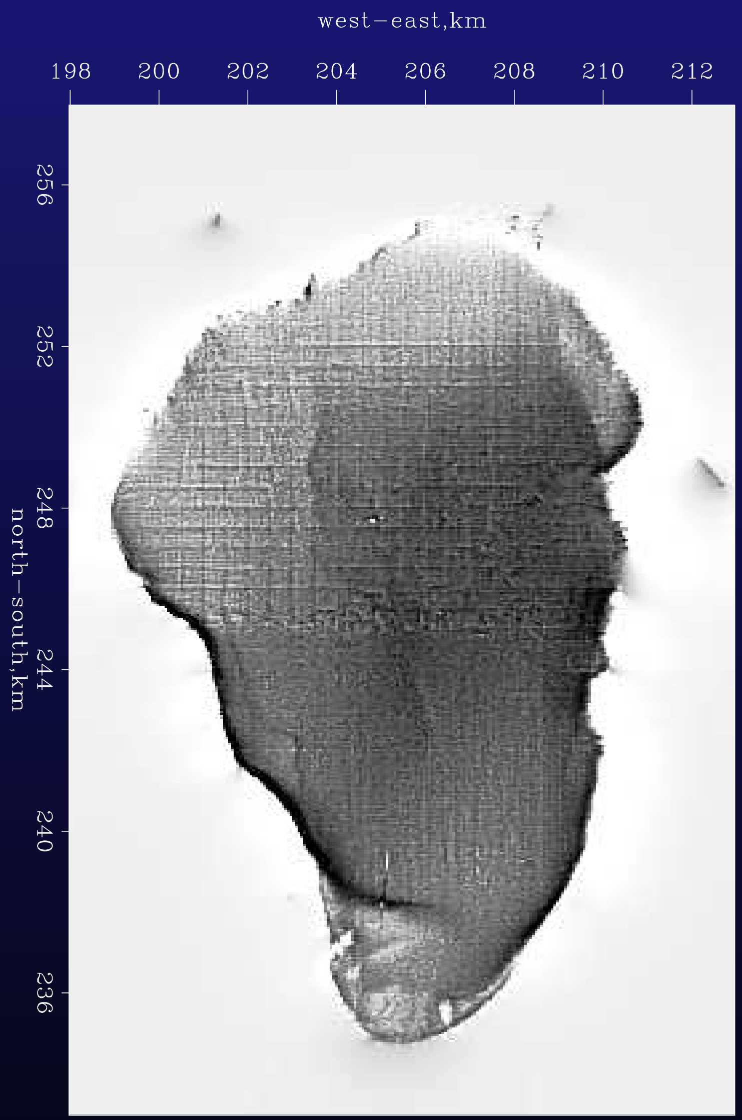
Adjust $h(x, y)$ and drift, to minimize the residual

$$0 \quad \approx \quad \mathrm{bin}^T\, h(x, y) \; + \; \mathrm{drift}(s) \; - \; \mathrm{data}(s)$$

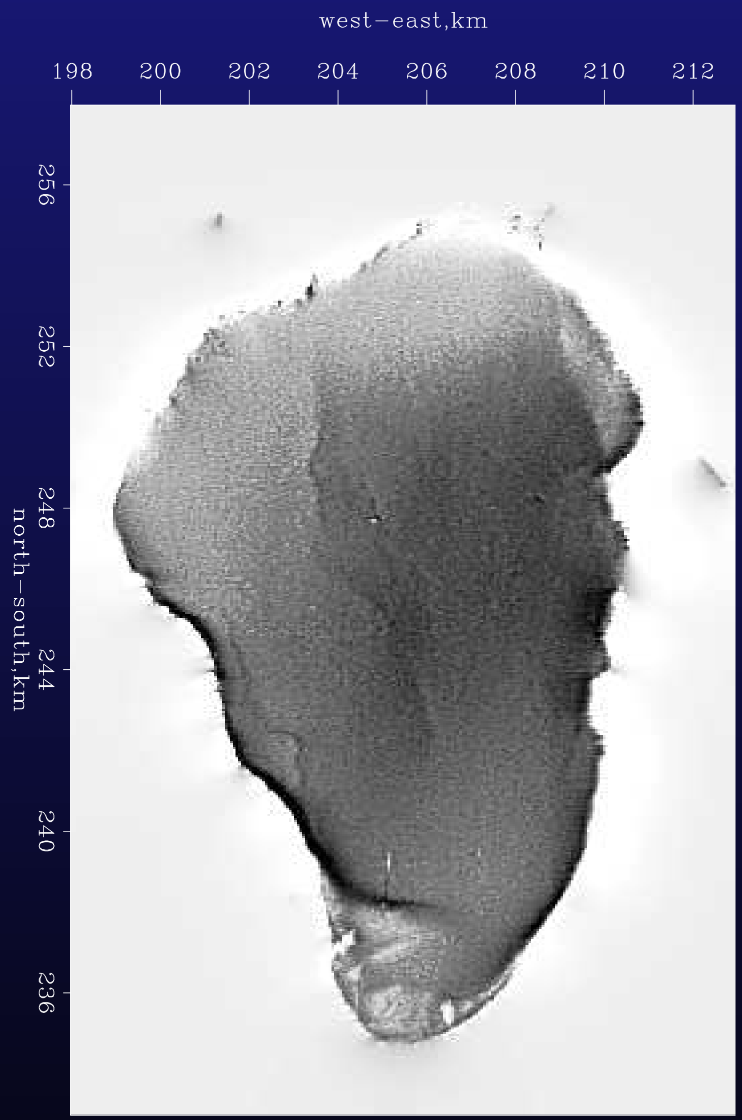
The free variables in the conjugate gradient iterations are $p(x, y)$ and the random numbers (into the low-pass filter on s).

Both free variable sets require regularization.

Modeling drift on track



Ignoring acquisition drift



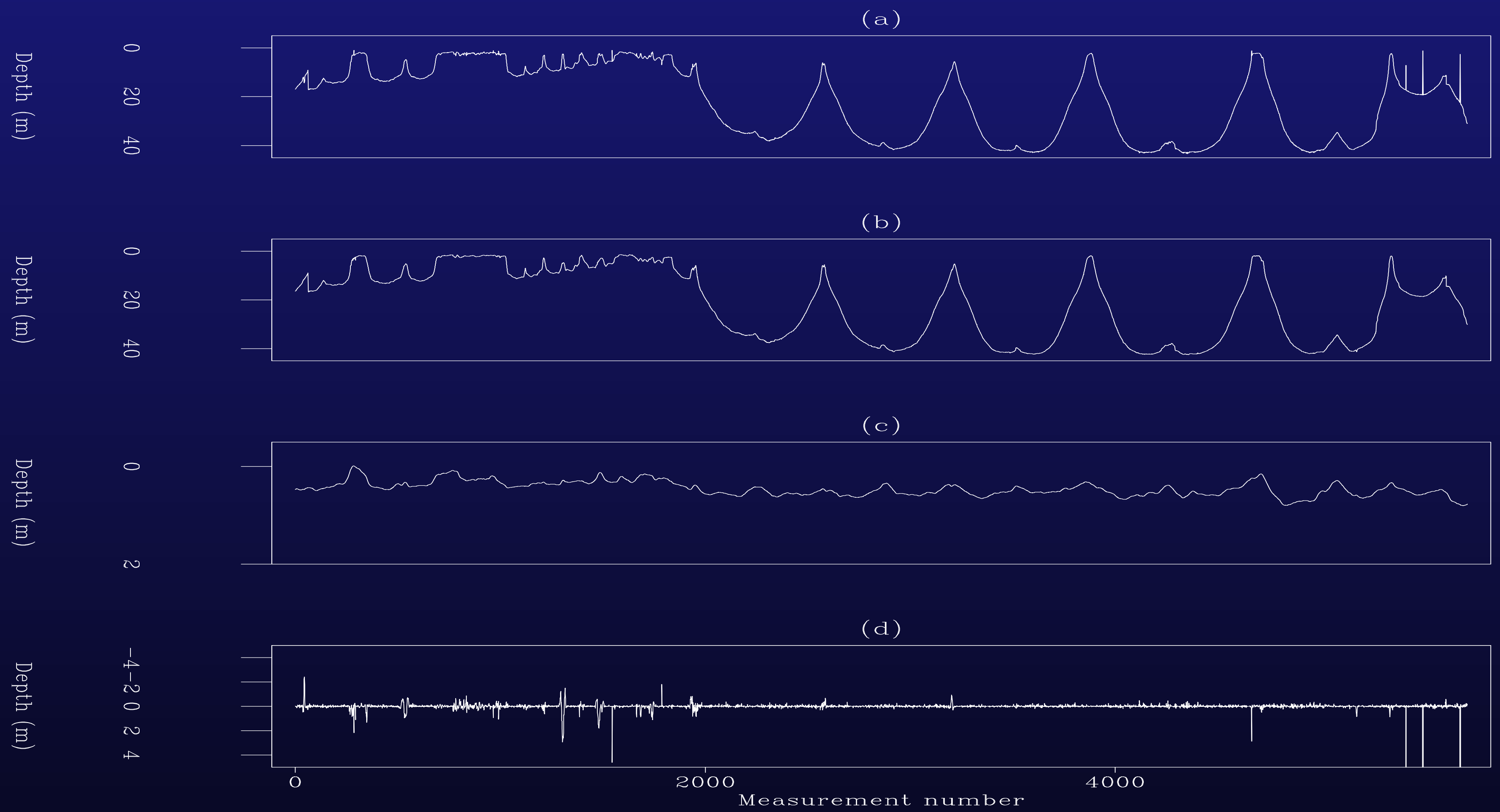
Fitting drift

I could stop here and take questions,

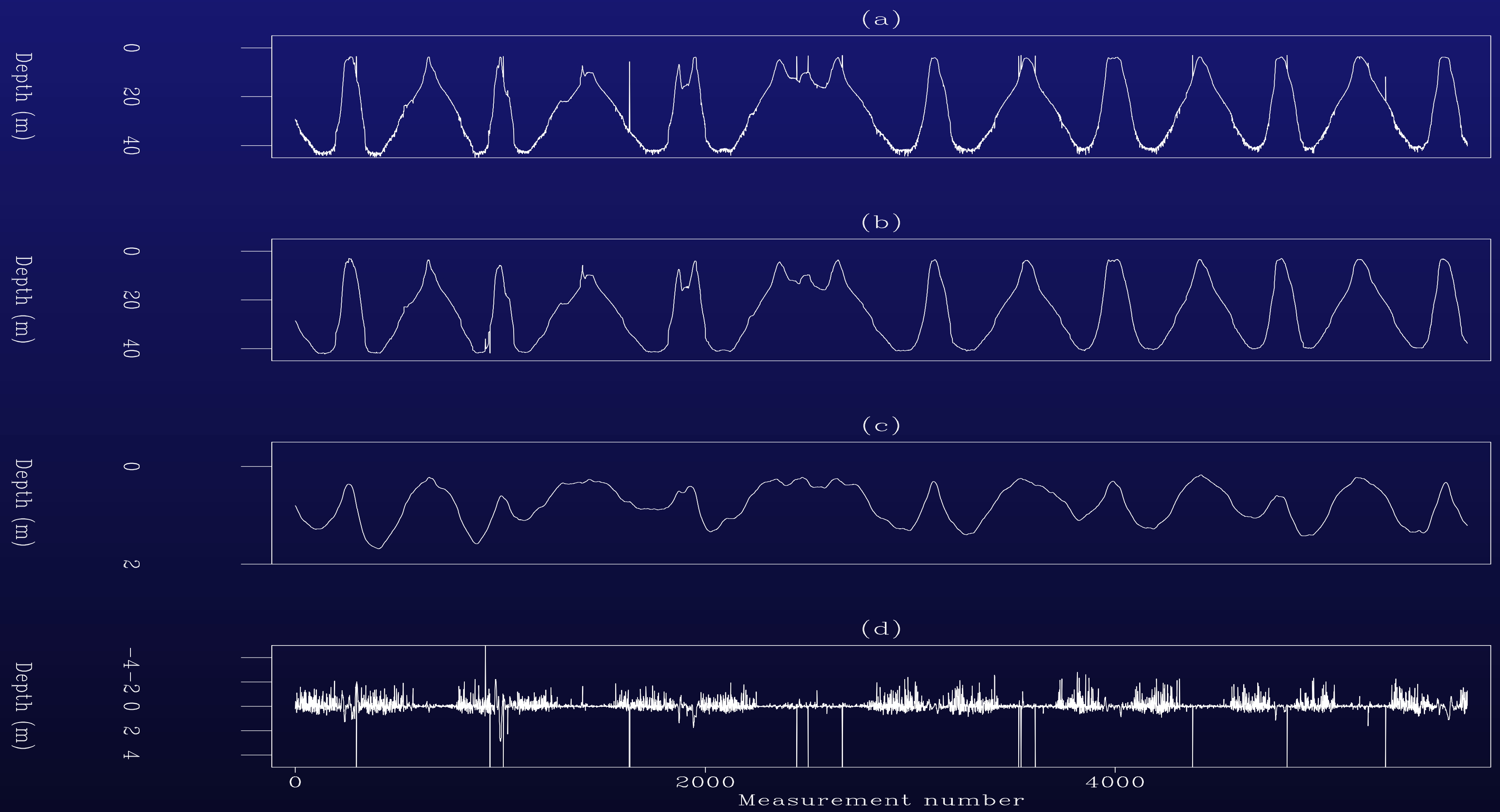
or examine the residuals

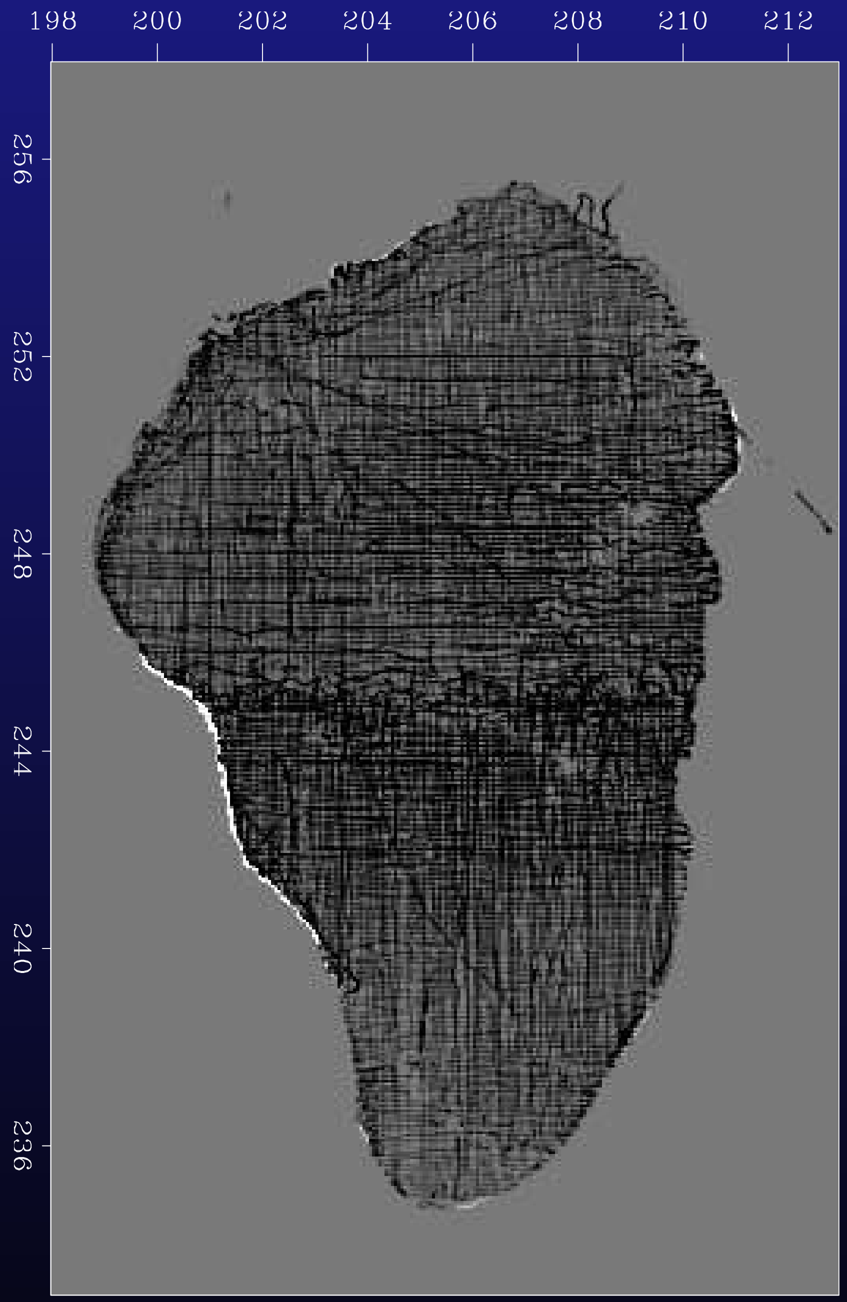
and see some failed ideas.

Data subset: raw, modeled, drift, residual

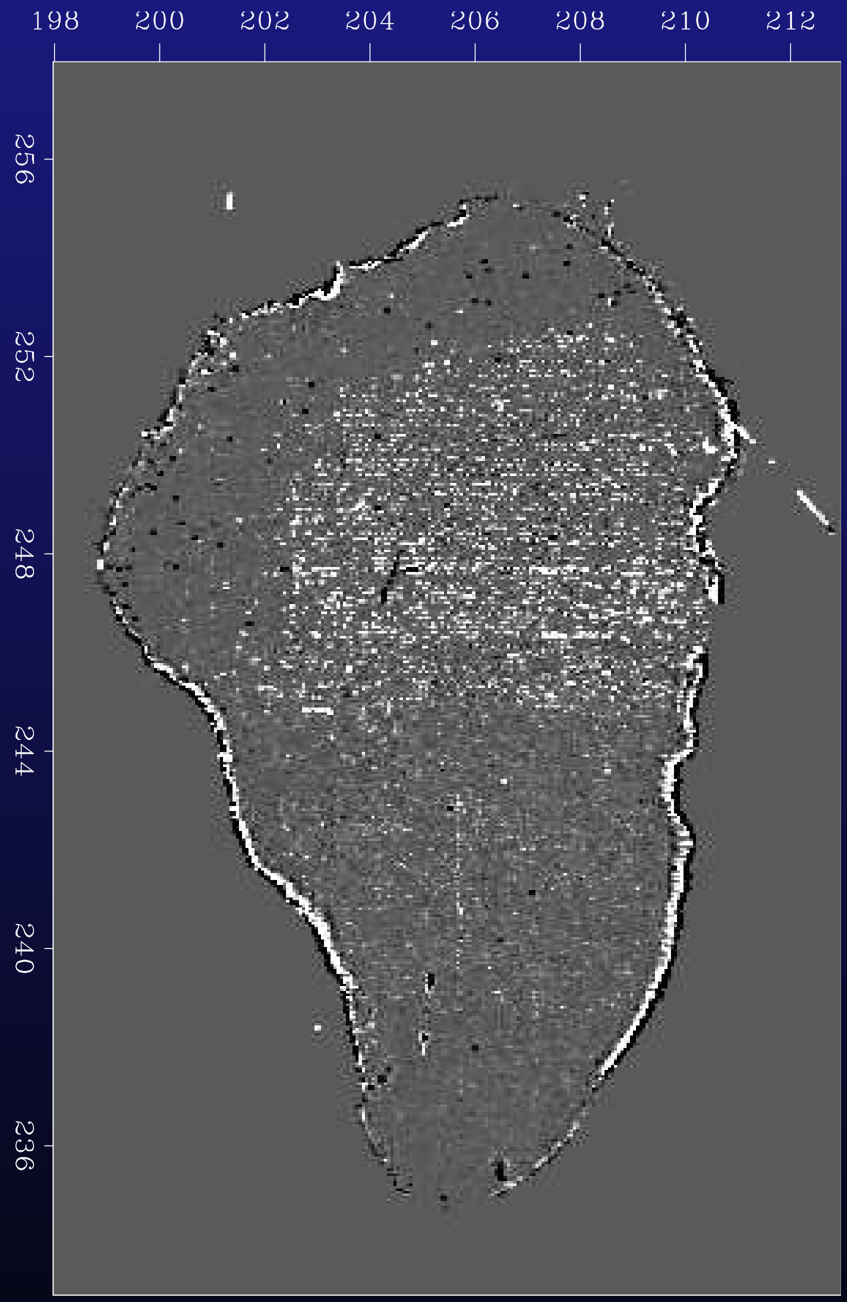


Data subset: raw, modeled, drift, residual



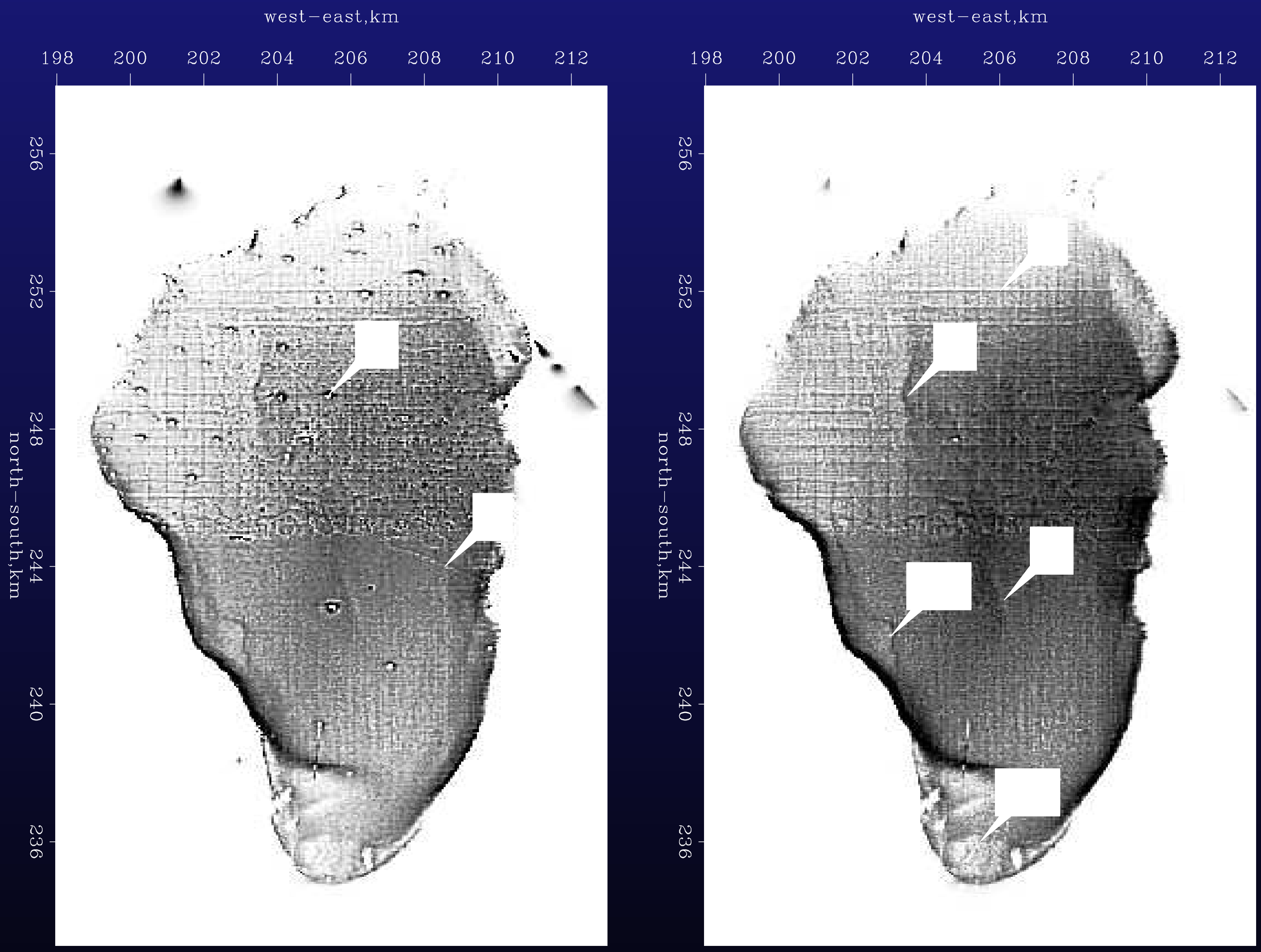


Data drift in model space

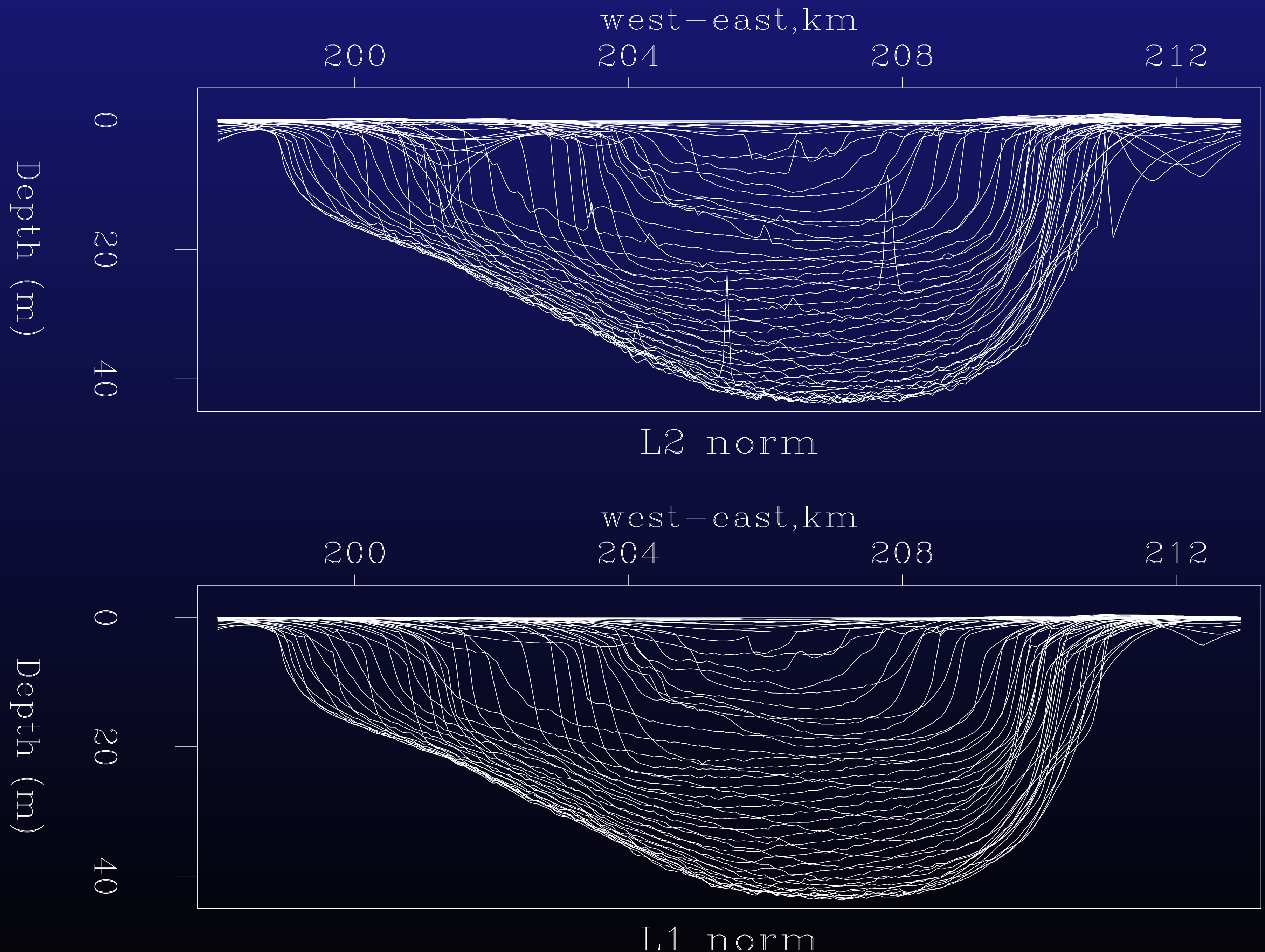


Data residual in model space

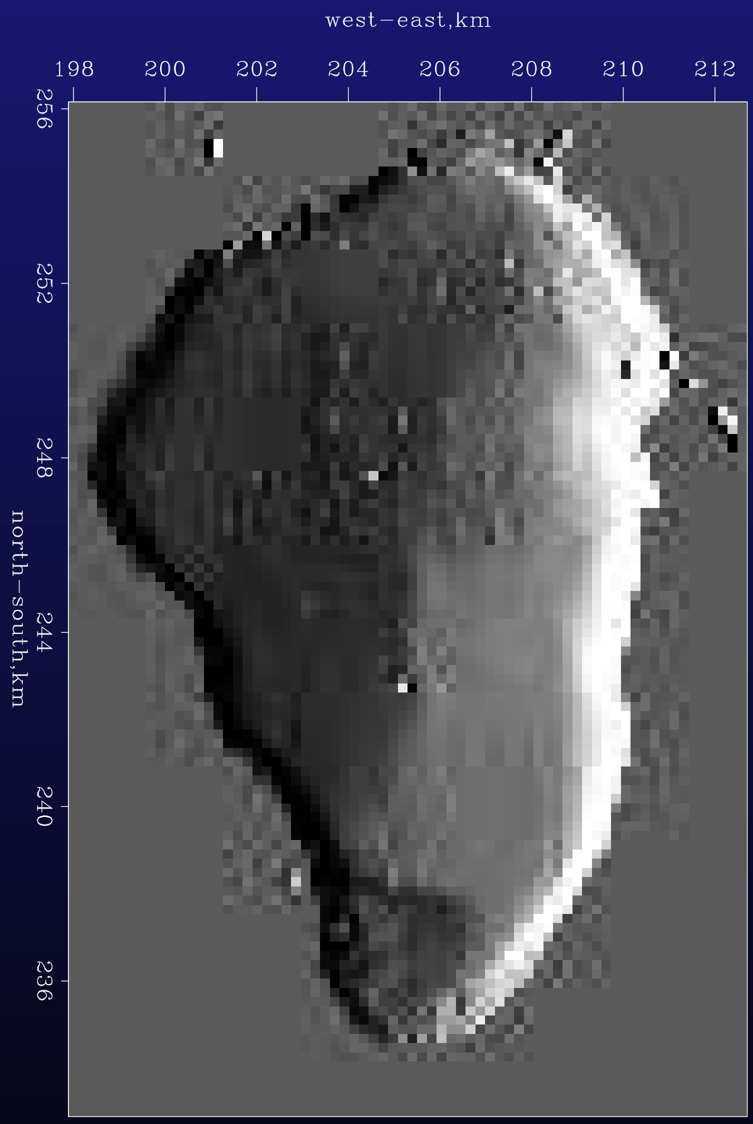
L2 and L1 norms



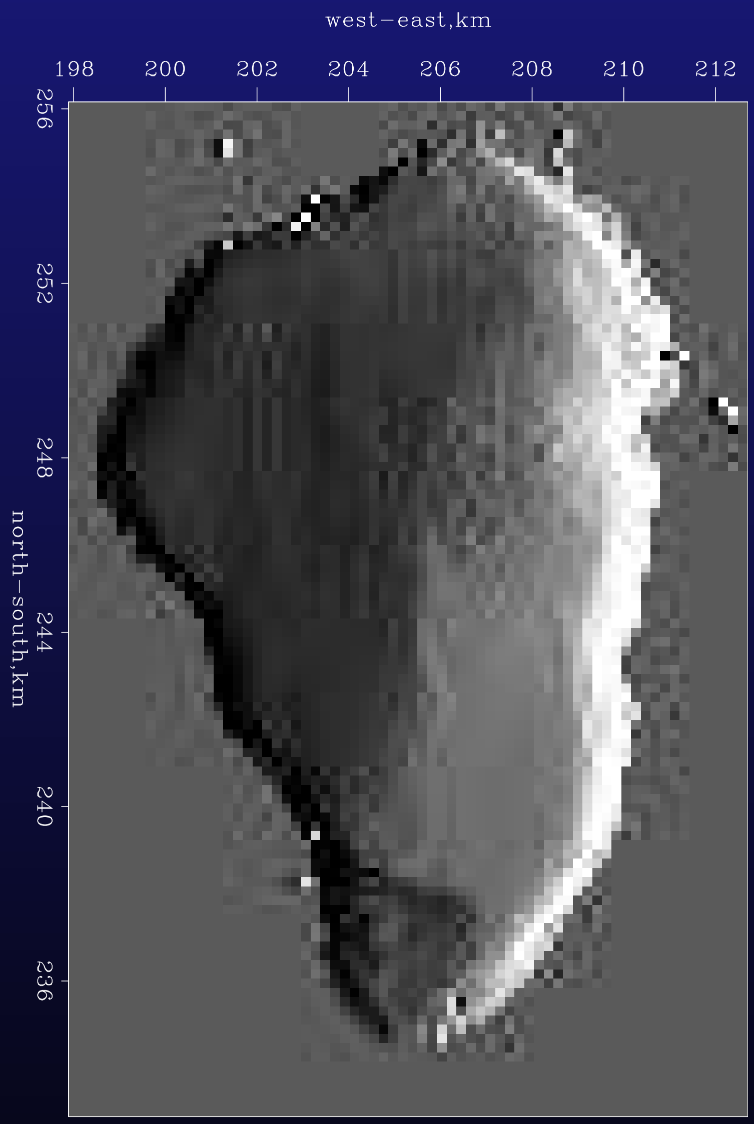
L2 and L1 norms



Median stack in each bin

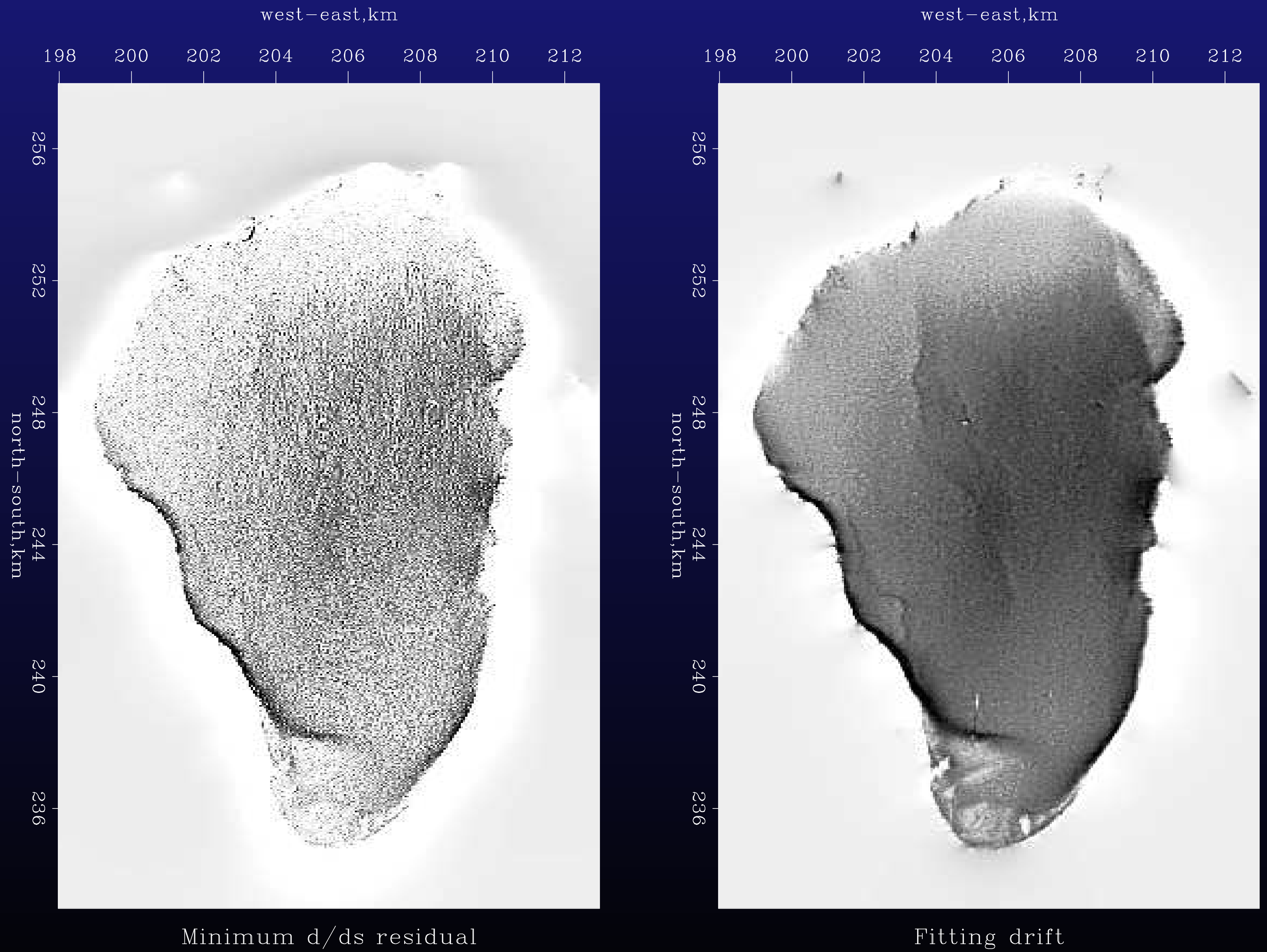


Mean bin and roughen



Median bin and roughen

Failure: Minimize d/ds residual



That's all folks!

More details on-line.

Google for "Claerbout" to find free book.