Two strategies for sparse data interpolation

Morgan Brown[1]

ABSTRACT

I introduce two strategies to overcome the slow convergence of least squares sparse data interpolation: 1) a 2-D multiscale Laplacian regularization operator, and 2) an explicit quadtree-style upsampling scheme which produces a good initial guess for iterative schemes. The multiscale regularization produces an order-of-magnitude speedup in the interpolation of a sparsely sampled topographical map. The quadtree method produces an initial guess which leads to similar speedups for iterative methods.

INTRODUCTION

Iterative methods of sparse data interpolation are prone to slow convergence because the small eigenvalues of most regularization operators correspond to slowly-varying trends in the unknown model. Any approach to improve convergence of iterative techniques must improve the condition number of the normal equations. SEP researchers have used recursive filter preconditioning (Fomel et al., 1997) to overcome this problem.

I introduce two strategies to overcome the slow convergence of these interpolation problems. Firstly, I implement a composite regularization operator which applies a 2-D Laplacian at different spatial scales. Use of the multiscale operator to regularize the least squares interpolation of a sparsely sampled topographical map produces an order-of-magnitude speedup in convergence, compared to the case of regularizing with the single-scale 2-D Laplacian. Secondly, I implement a quadtree-style scheme to explicitly interpolate sparsely sampled data. The quadtree method is fast, and as shown on the topographical intepolation example, produces reasonable results itself, and may also be used as an initial guess for inversion schemes.

BACKGROUND

Many types of geophysical data consist of measurements of a given quantity, collected at arbitrary locations near the earth's surface. The problem is to infer the value of this quantity at *all* locations in the study area - in other words, to estimate the earth model of the quantity that gave rise to the collected data. These ideas are embodied in the following simple linear

[1] **email:** morgan@sep.stanford.edu

relationship

$$\mathbf{B}\mathbf{m} = \mathbf{d}. \tag{1}$$

d and **m** are the measured data and estimated model, respectively, while **B** is a linear operator that carries out the "experiment" by sampling the model at the measurement locations and mapping these values to the data vector. However, the reverse – an estimate of the model, given the data – is usually more useful. In this case, **B** must be "inverted" in some sense:

$$\hat{\mathbf{m}} = \mathbf{B}^{\dagger}\mathbf{d}. \tag{2}$$

The simplest choice is $\mathbf{B}^{\dagger} = \mathbf{B}^{T}$, but any choice so that the model honors the known data has a measure of validity. $\mathbf{B}^{\dagger}$ may also be cast as a least squares inverse. Depending on the acquisition geometry of the experiment, the least squares problem may be underdetermined, overdetermined, or more commonly, both (Menke, 1989). For purely overdetermined problems, the least squares inverse is $\mathbf{B}^{\dagger} = (\mathbf{B}^{T}\mathbf{B})^{-1}\mathbf{B}^{T}$, but if some model points are undetermined, $\mathbf{B}^{T}\mathbf{B}$ is singular.

Regularization

One method of solving so-called "mixed-determined" problems is to force the problem to be purely overdetermined by applying regularization, in which case Equation (1) becomes

$$\begin{bmatrix} \mathbf{B} \\ \epsilon\mathbf{A} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \tag{3}$$

A is the regularization operator; usually convolution with a compact differential filter. ϵ is a scaling factor. The least squares inverse is then

$$\mathbf{B}^{\dagger} = (\mathbf{B}^{T}\mathbf{B} + \epsilon^{2}\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{B}^{T}. \tag{4}$$

The regularization term, $\epsilon^{2}\mathbf{A}^{T}\mathbf{A}$, is nonsingular with positive eigenvalues, so it stabilizes singularities in $\mathbf{B}^{T}\mathbf{B}$, but it is poorly-conditioned for many common choices of **A**, i.e., Laplacian or gradient. The smallest eigenvalues of $\epsilon^{2}\mathbf{A}^{T}\mathbf{A}$ correspond to smooth (low-frequency) model components, so iterative methods of solving equation (4), including the conjugate-direction method used in this paper, require many iterations to obtain smooth estimates of the model (Shewchuk, 1994).

Preconditioning

We can precondition equation (3) by making a simple change of variables:

$$\mathbf{m} = \mathbf{S}\mathbf{x}. \tag{5}$$

Analogous to equation (4), we can write the least squares inverse for the preconditioned model **x**:

$$\mathbf{B}^{\dagger} = (\mathbf{S}^{T}\mathbf{B}^{T}\mathbf{B}\mathbf{S} + \epsilon^{2}\mathbf{S}^{T}\mathbf{A}^{T}\mathbf{A}\mathbf{S})^{-1}\mathbf{B}^{T}. \tag{6}$$

If $\mathbf{S}$ is the left inverse of $\mathbf{A}$ ($\mathbf{SA} = \mathbf{I}$), then equation (6) reduces to the classic damped least squares problem (Menke, 1989):

$$\mathbf{B}^{\dagger} = (\mathbf{S}^T\mathbf{B}^T\mathbf{BS} + \epsilon^2\mathbf{I})^{-1}\mathbf{B}^T. \tag{7}$$

If $\mathbf{A}$ is a differential operator, $\mathbf{S}$ is then a smoothing operator, and it follows that the smallest eigenvalues of $\mathbf{S}^T\mathbf{B}^T\mathbf{BS}$ correspond to the complex (high frequency) model components. In contrast to equation (4), smooth, useful models will appear in early iterations of the preconditioned problem of equation (7), although absolute rate of convergence to the same final result should not change.

Spectral factorization (Sava et al., 1998) and the Helix transform (Claerbout, 1998) permit multidimensional, recursive, approximate inverse filtering, so it is indeed possible to compute $\mathbf{S} \approx \mathbf{A}^{-1}$ for many choices of $\mathbf{A}$. One downside of recursive filter preconditioning is that the operator is difficult to parallelize. For large problems, the cost of a single least squares iteration may be considerable, so the parallelization issue should be kept in mind.

Multiscale regularization

Another approach to combat the slow convergence of least squares sparse data interpolation is to design a regularization operator that works at multiple scales simultaneously. Starting with equation (3), we replace the regularization operator, $\mathbf{A}$, with a composite regularization operator (for the two-scale case):

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{AD}_k \end{bmatrix} \tag{8}$$

$\mathbf{D}_k$, mnemonic for *downsampling*, is a normalized binning operator which subsamples a vector of size n to a vector of size n/k, implicitly smoothing it in the process. Replacing $\mathbf{A}$ in equation (3) with this new regularization operator gives

$$\begin{bmatrix} \mathbf{B} \\ \epsilon_1\mathbf{A} \\ \epsilon_2\mathbf{AD}_k \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{9}$$

ϵ_1 and ϵ_2 are scaling factors. In the fashion of equation (4), we can write the least squares inverse corresponding to the system of equation (9):

$$\mathbf{B}^{\dagger} = (\mathbf{B}^T\mathbf{B} + \epsilon_1^2\mathbf{A}^T\mathbf{A} + \epsilon_2^2\mathbf{D}_k^T\mathbf{A}^T\mathbf{AD}_k)^{-1}\mathbf{B}^T \tag{10}$$

Applying the downsampling operator $\mathbf{D}_k$ to the model vector attenuates high-frequency components while boosting low-frequency components, thus we infer that the eigenvalue spectrum of $\mathbf{A}^T\mathbf{A} + \mathbf{D}_k^T\mathbf{A}^T\mathbf{AD}_k$ is better balanced than that of $\mathbf{A}^T\mathbf{A}$ alone, which speeds convergence to a smooth model.

Claerbout (1999) presents a very similar multiscale methodology with one important difference: the filters, not the data, are upscaled from one scale to the next. Crawley (2000) applies this methodology to interpolating seismic data with nonstationary prediction error filters (PEF). The PEF is more readily upscaled, since it is normally conceptualized as a dip annhilator, and it annhilates the same dips at all scales. Unfortunately, other filters, like the Laplacian finite difference filter used in this paper, do not have the self-similarity property of the PEF, so explicitly expanding the filter is a dangerous proposition.

Quadtree Pyramid Interpolation

The quadtree decomposition is an adaptive sampling scheme which seeks to divide a digital image into regions of nearly homogeneous pixel value. Intuitively, the block size is smallest where the image changes most rapidly, and largest where the pixel values are constant over large areas.

A similar intuition applies to the interpolation of sparse data. Given point data collected on the Earth's surface, equation (1) relates how the data is placed into the discrete computational grid. If data is collected at perfectly regular intervals on the Earth's surface, it is possible to choose a bin size such that one and only one measurement falls in each bin. On the other hand, if the data sampling is irregular, two problems may arise: 1) more than one datum may fall into a given bin, and the values averaged, implying information loss, and 2) no data may fall into a given bin, leaving a "hole" in the model.

Applied to interpolation, the quadtree methodology seeks to adaptively sample the model such that 1) where data are closely spaced, the bin size is small, to minimize averaging of adjacent data and 2) where data are sparsely distributed, the bin size is large, to avoid introducing holes in the model.

First assume that there exists a regular bin size such that binning the data produces a model with no holes. From here, we regard "bin size" as equivalent to "scale" - where scale goes from coarsest (largest bins) to finest (smallest bins). Also assume that at each scale, the bins which contain one or more data values are known.

- $\mathbf{m}_i$ - Model at scale i; $i = 0$ is coarsest scale, $i = n$ is finest scale.
- $\mathbf{B}_i$ - Bin data onto grid of scale i; $i = 0$ is coarsest scale, $i = n$ is finest scale.
- $\mathbf{K}_i$ - Known data mask at scale $i - 1$ for bins which contain data, 0 otherwise; $i = 0$ is coarsest scale, $i = n$ is finest scale.
- $\mathbf{U}_i$ - Upsampling operator. Upsample from scale $i - 1$ to scale i. $i = 0$ is coarsest scale, $i = n$ is finest scale. Adjoint to the downsampling operator in the multiscale regularization discussion. For instance, if the downsampling operator sums four input bin locations into an output location and averages, the upsampling operator takes the averaged value and places it back into the four bins.
- $\mathbf{d}$ - data.

The algorithm proceeds as follows.

```
Compute $\mathbf{m}_0 = \mathbf{B}_0^T \mathbf{d}$.
Loop over scale: $i = 1 \rightarrow n$.
    Upsample $\mathbf{m}_i^u = \mathbf{U}_i \mathbf{m}_{i-1}$.
    Compute $\mathbf{m}_i^b = \mathbf{B}_i^T \mathbf{d}$.
    Where $\mathbf{K}_i = 1, \mathbf{m}_i = \mathbf{m}_i^b$. Otherwise, $\mathbf{m}_i = \mathbf{m}_i^u$.
End Loop.
```

This approach is quite similar to the *Multigrid*-style method employed by Crawley (1995), but there is no inversion involved – my method is totally explicit. Interestingly, Luettgen et al. (1994) show that for the underdetermined optical flow determination problem of image processing, use of an explicit quadtree scheme gives results identical by some measures to the result obtained by solving a regularized inverse problem.

RESULTS

In this paper, I use a particularly simple example dataset - a 10-meter-resolution digital elevation map (DEM) of the area surrounding Fallen Leaf Lake, obtained freely from USGS[2]. This map, along with some relevant landforms, is shown in Figure 1. To simulate an experiment, I sampled the map randomly at 2250 points: 1000 points in the northern region, 1000 in the southern region, and only 250 in the central region. The model grid is 256x256 points, giving a 40-meter output resolution. Figure 2 shows the experiment's fold, which varies from 0 to 3.

Figures 3 through 8 show the results of applying various estimation schemes to fill the holes in the acquisition. Starting from upper-left and moving clockwise, each of the four figures shows a) the "answer," i.e., the 1024x1024 topographical surface, subsampled by a factor of four, b) the estimated model, c) The error in the estimate clipped to a common value and overlain by the the 2250 known data locations, and d) a crossplot of the estimated model and answer at 4000 randomly chosen spatial locations.

Figure 3 shows 100 iterations with Laplacian regularization. Obviously, convergence has not been achieved. Figure 4 shows 1000 iterations with Laplacian regularization. The crossplot is very tight, making this result tough to beat. Figure 5 shows 100 iterations of Laplacian regularization with preconditioning. This result is disappointing: Convergence to this result occured in only 10-20 iterations, but the result itself is not desirable. Although not the subject of this paper, the "ice-cream-cone" nature of this result is alarming and merits further investigation. Figure 6 shows 100 iterations of multiscale Laplacian regularization. This result is quite similar to the 1000 iterations Lapacian result, but the crossplot is not as tight. Also, we expect this result to be a bit smoother than the pure Laplacian, but it is difficult to see if it is. Figure 7 shows the explicit Quadtree Pyramid interpolation. The quadtree structure of the

[2] `http://edcwww.cr.usgs.gov/doc/edchome/ndcdb/ndcdb.html`

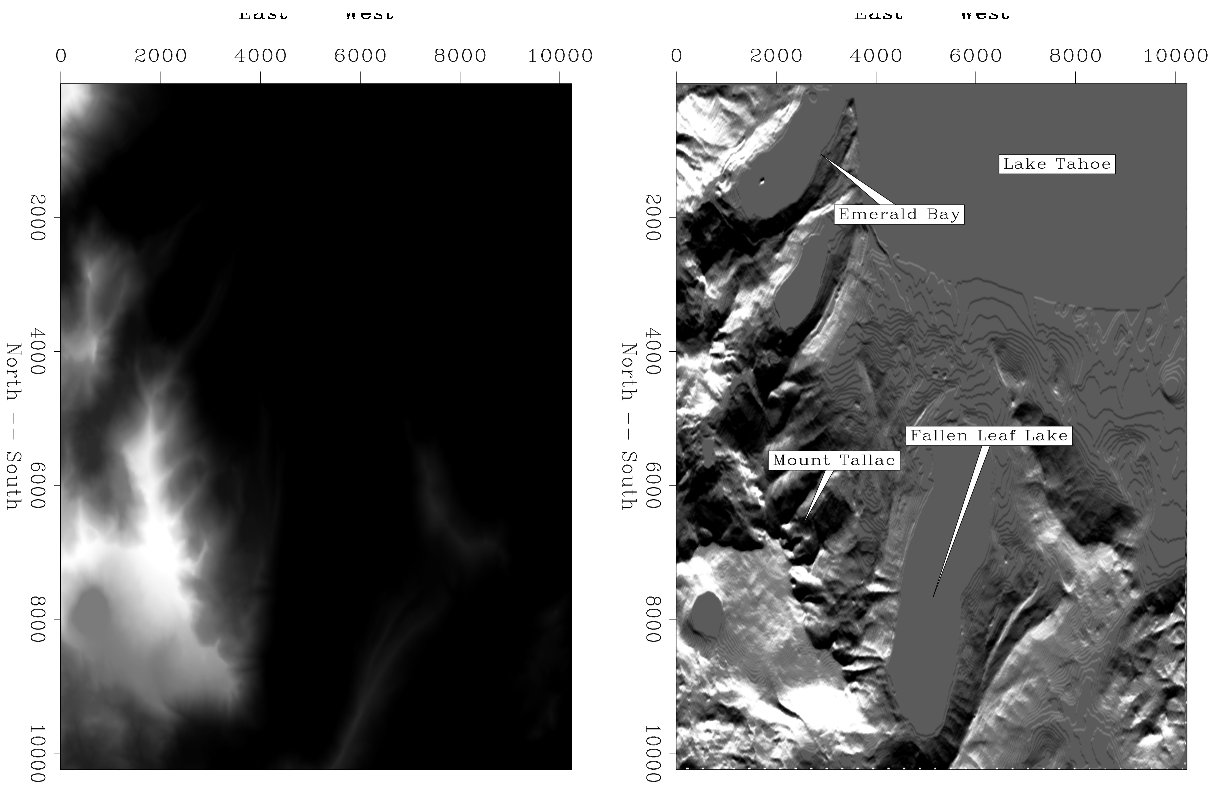

Figure 1: Left: Digital elevation map of the vicinity around Fallen Leaf Lake, California. Resolution is 10 meters. Windowed down from an original size of roughly 1400x1200 points to 1024x1024 points. Right: North-south derivative applied to topography to highlight landforms. morgan1-topo2 [ER]

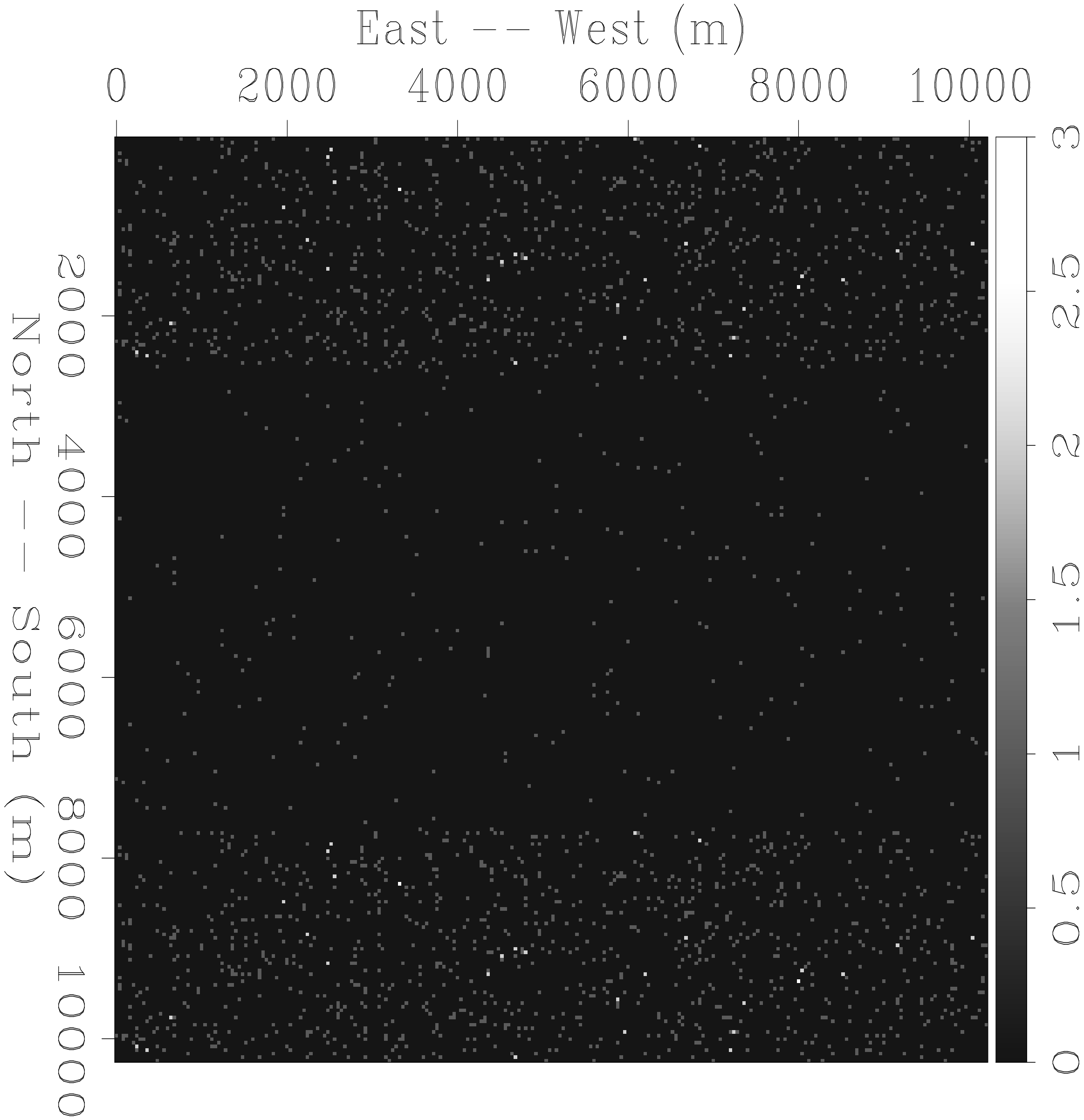

Figure 2: Experimental fold. 2250 random samples on the 1024x1024 map are the “data”. Model grid is 256x256. morgan1-topo2-fold [ER]

result is quite apparent. Although the interpolated map is not smooth spatially, the general structure of the topography are reproduced quite convincingly. Figure 8 shows 100 iterations of Laplacian regularization, with the explicit Quadtree Pyramid interpolation (Figure 7) used as a starting guess. This result is quite similar to the 1000 iterations Laplacian result; also the 100 iterations with the multiscale Laplacian.

As the "answer" is known in this problem, we can quantitatively check the accuracy of an estimated model. The statistics below speak for themselves. Taking the standard deviation of the error to be a measure of goodness, we can rank the results from best to worst:

1. Laplacian regularization, 1000 iterations.
2. Multiscale Laplacian regularization, 100 iterations.
3. Laplacian regularization, 100 iterations, quadtree pyramid interpolation used as starting guess.
4. Quadtree pyramid direct interpolation.
5. Preconditioned Laplacian regularization, 100 iterations.
6. Laplacian regularization, 100 iterations.

Laplacian regularization, 100 iterations

```
                   | Min. Val. | Max. Val. | Mean Val. |  Median   | Std. Dev. |
 -------------------------------------------------------------------------------
     True values:    -42.10075   253.25171    33.82695    12.50521    48.19902
Estimated values:      0.00000   246.00000    44.96923    18.00000    54.02688
          Errors:    -70.55768   235.33752    11.14226     0.87845    29.06191
 Absolute errors:      0.00000   235.33752    12.89230     2.21318    28.32898
 Relative errors:      0.00000  1206.58191     3.02346     0.06742    33.63662
```

Laplacian regularization, 1000 iterations

```
                   | Min. Val. | Max. Val. | Mean Val. |  Median   | Std. Dev. |
 -------------------------------------------------------------------------------
     True values:      3.19786   232.39223    45.01379    20.00496    53.70287
Estimated values:      0.00000   246.00000    44.96923    18.00000    54.02688
          Errors:    -68.31208    76.44979    -0.04465    -0.00007     5.77580
 Absolute errors:      0.00000    76.44979     2.51392     0.80804     5.20023
 Relative errors:      0.00000     1.00000     0.06820     0.02891     0.10538
```

Preconditioned Laplacian regularization, 100 iterations

```
                   | Min. Val. | Max. Val. | Mean Val. |  Median   | Std. Dev. |
 -------------------------------------------------------------------------------
     True values:      2.55815   220.00000    38.08480    18.64038    45.06718
Estimated values:      0.00000   246.00000    44.96923    18.00000    54.02688
          Errors:    -61.61859   174.82333     6.88441     0.54753    19.72276
 Absolute errors:      0.00000   174.82333     8.05753     1.45995    19.27327
 Relative errors:      0.00000    43.94756     0.26195     0.07687     1.28934
```

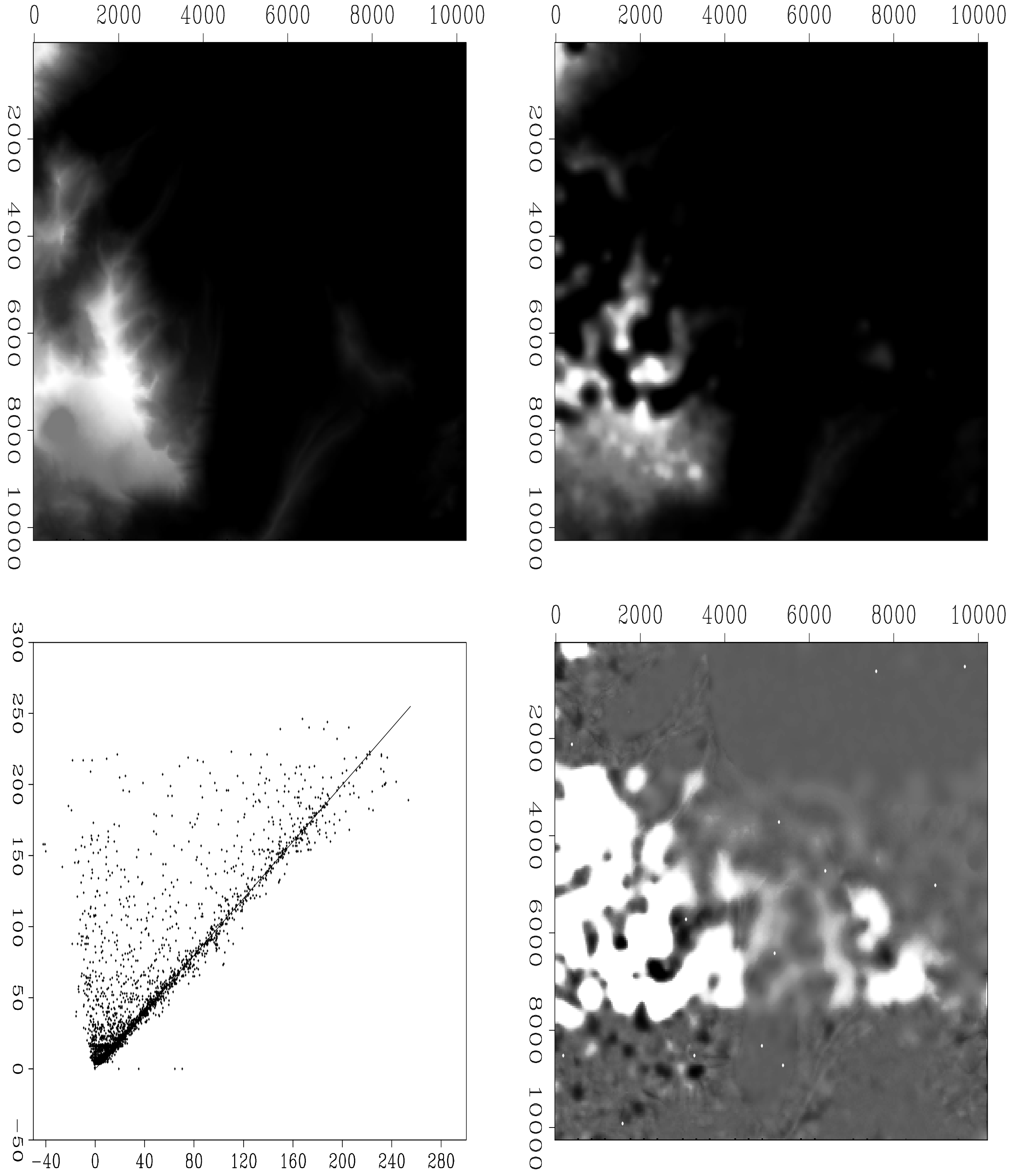

Figure 3: Simple Laplacian regularization, 100 iterations. Clockwise from upper-left: True model; Estimated model; Error; Crossplot of true versus estimated model. morgan1-topodata2-lap100-show [ER]

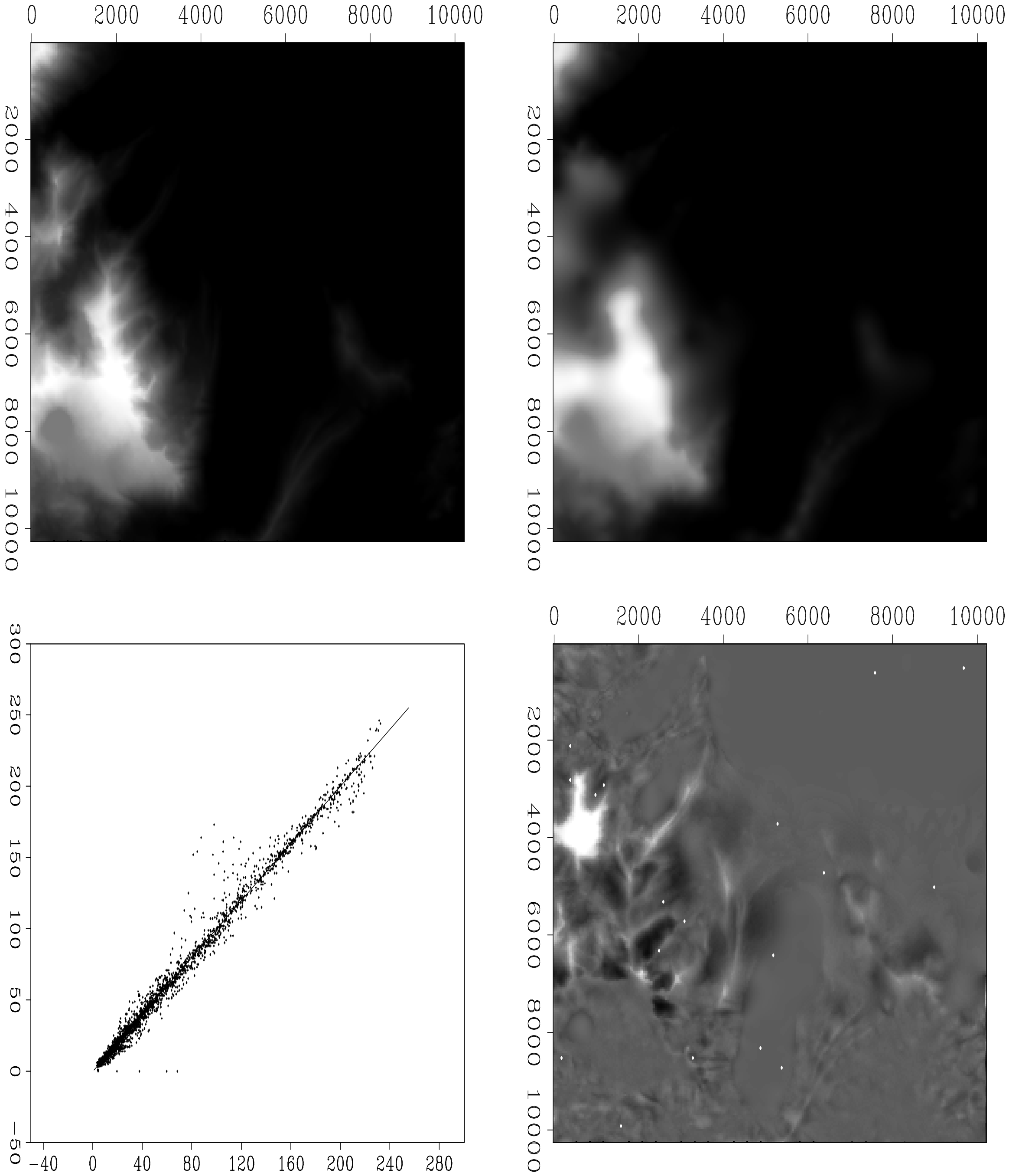

Figure 4: Laplacian regularization, 1000 iterations. Clockwise from upper-left: True model; Estimated model; Error; Crossplot of true versus estimated model. morgan1-topodata2-lap1000-show [ER]

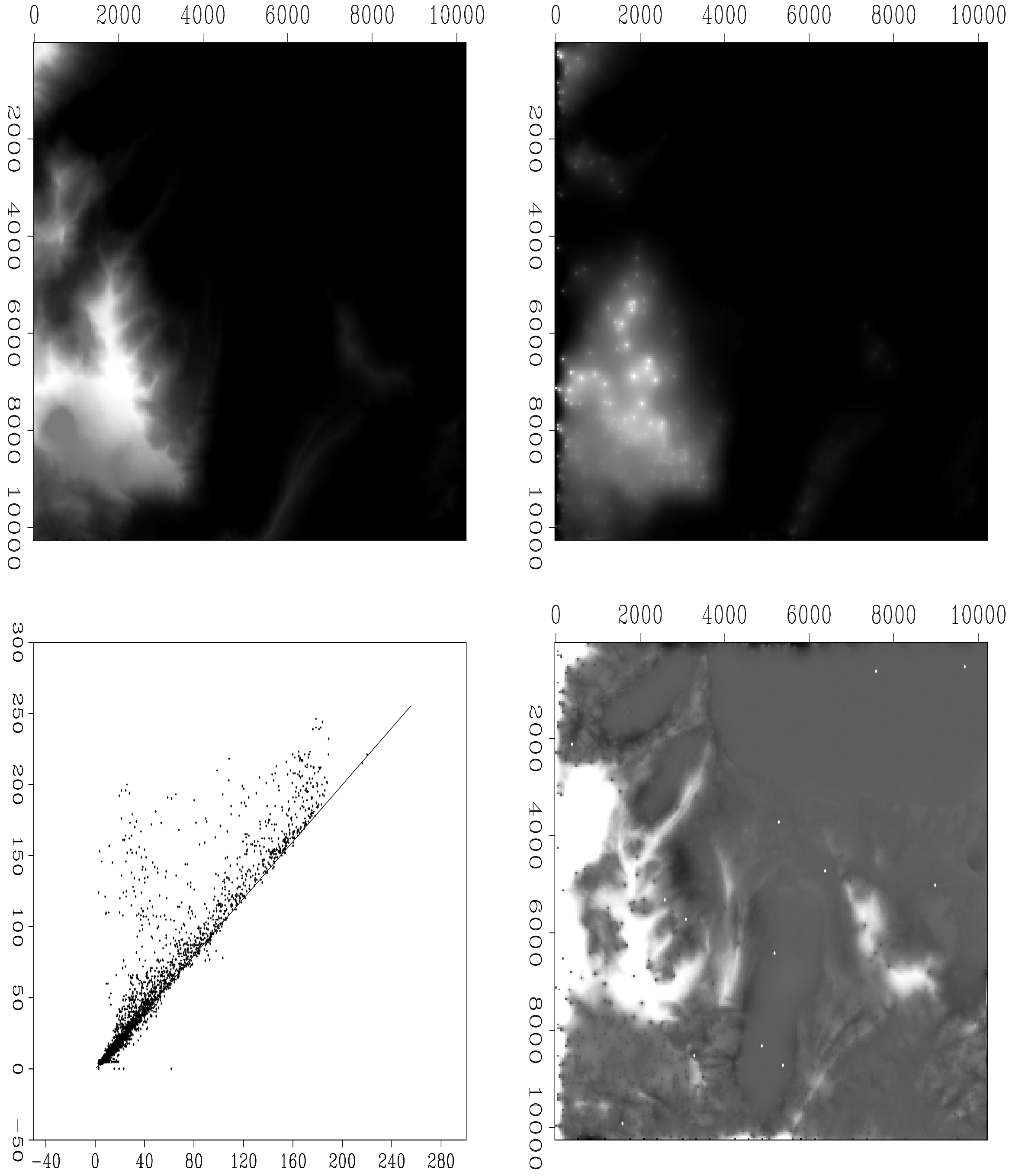

Figure 5: Preconditioned Laplacian regularization, 100 iterations. Clockwise from upper-left: True model; Estimated model; Error; Crossplot of true versus estimated model. morgan1-topodata2-prec100-show [ER]

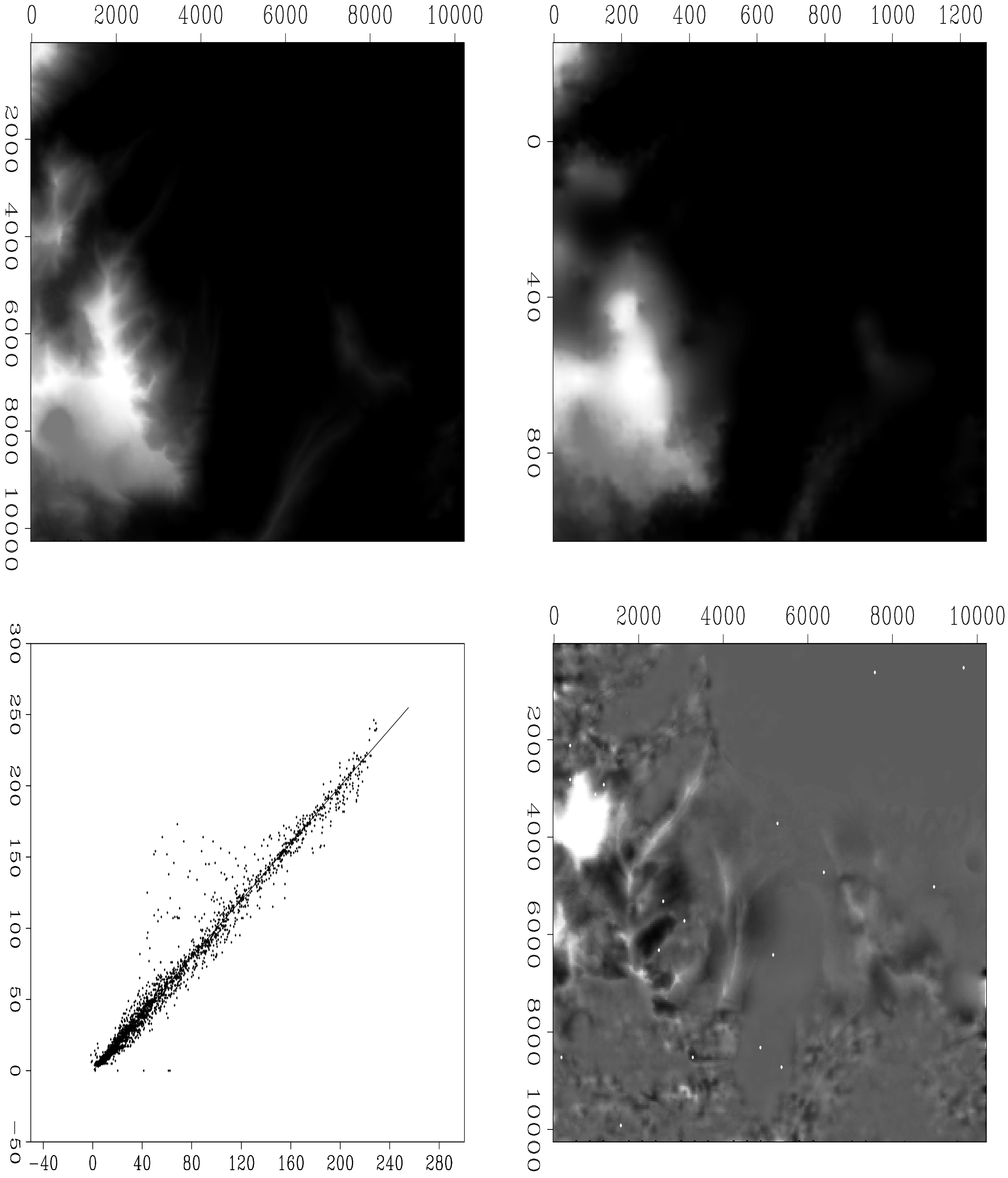

Figure 6: Multiscale Laplacian regularization, 100 iterations. Clockwise from upper-left: True model; Estimated model; Error; Crossplot of true versus estimated model. morgan1-topodata2-ms100-show [ER]

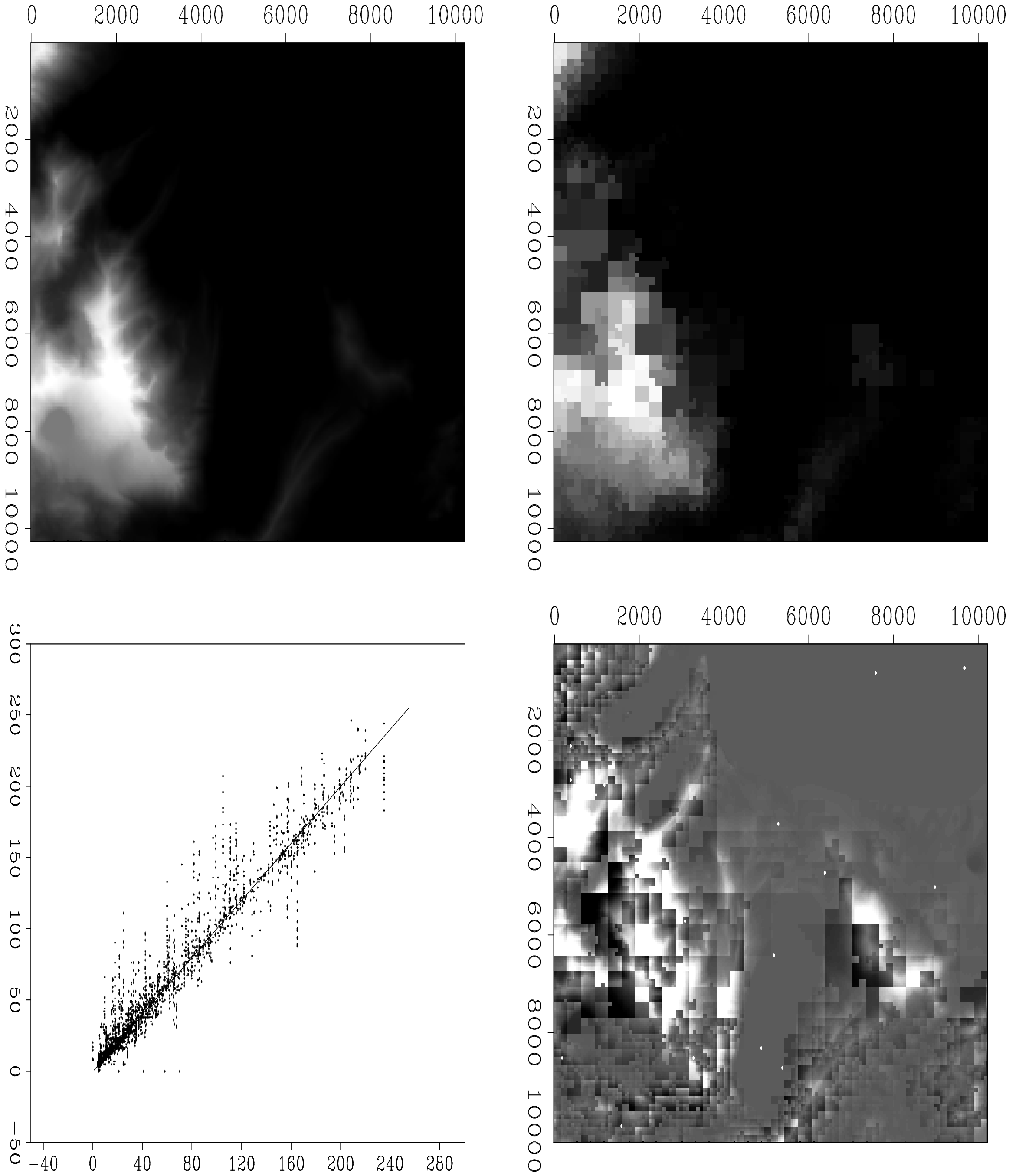

Figure 7: Quadtree pyramid interpolation result. Clockwise from upper-left: True model; Estimated model; Error; Crossplot of true versus estimated model. morgan1-topodata2-pyramid-show [ER]

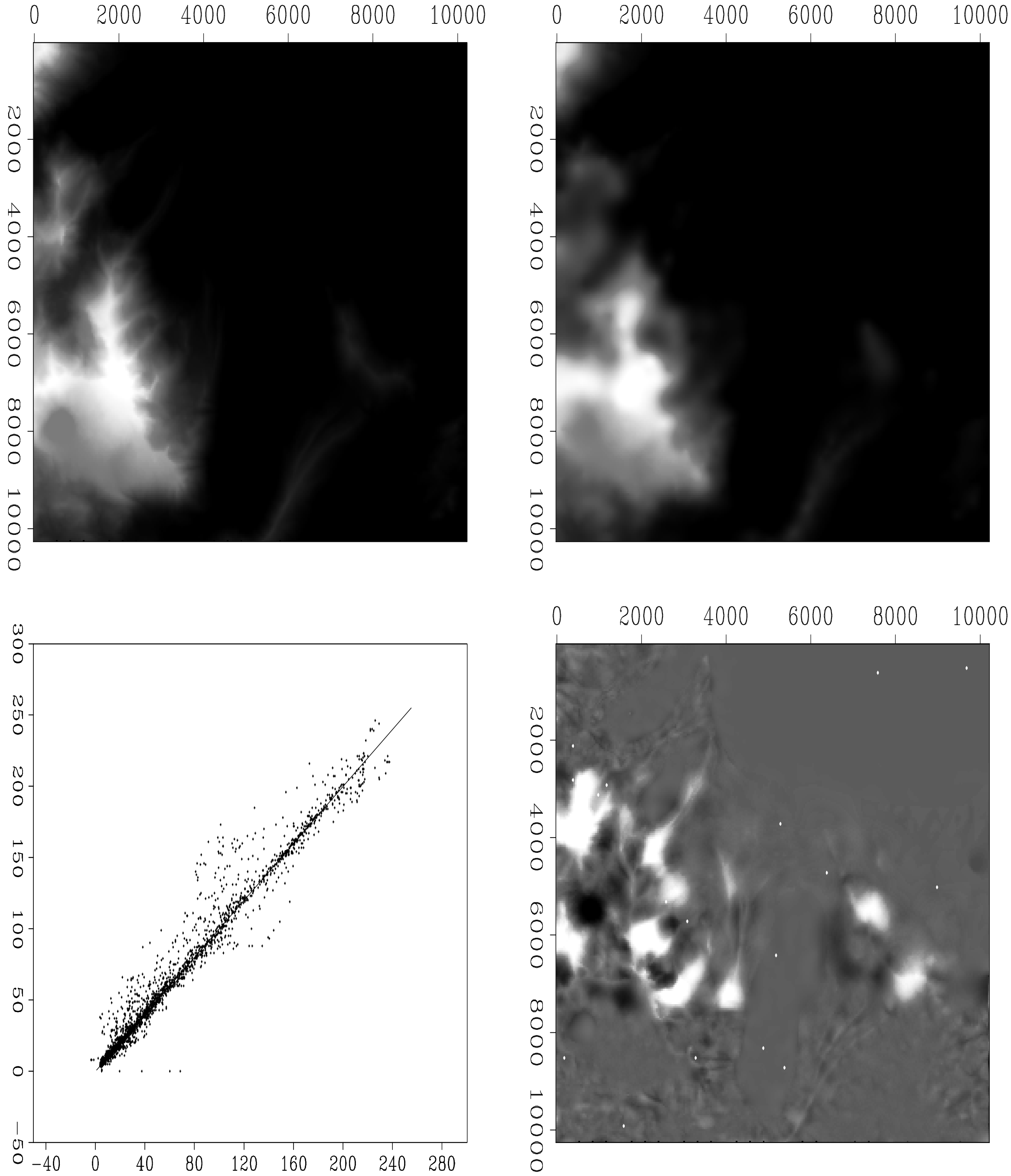

Figure 8: Laplacian regularization, 100 iterations. Clockwise from upper-left: True model; Estimated model; Error; Crossplot of true versus estimated model. morgan1-topodata2-pyrlap100-show [ER]

Multiscale Laplacian regularization, 100 iterations

```
                   | Min. Val. | Max. Val. | Mean Val. |  Median   | Std. Dev. |
 ------------------------------------------------------------------------------
     True values:     -1.53884   229.31610    44.82654    20.00943    53.62170
Estimated values:      0.00000   246.00000    44.96923    18.00000    54.02688
          Errors:    -62.51950   107.75475     0.14272    -0.00067     8.23634
 Absolute errors:      0.00000   107.75475     3.38367     1.06108     7.51047
 Relative errors:      0.00000     7.89127     0.09558     0.03850     0.23027
```

Quadtree pyramid interpolation

```
                   | Min. Val. | Max. Val. | Mean Val. |  Median   | Std. Dev. |
 ------------------------------------------------------------------------------
     True values:      0.00000   235.00000    42.54507    18.00000    52.90804
Estimated values:      0.00000   246.00000    44.96923    18.00000    54.02688
          Errors:    -77.00000   102.00000     2.42414     0.00000    12.79912
 Absolute errors:      0.00000   102.00000     6.17795     1.00000    11.46838
 Relative errors:      0.00000     5.82759     0.18037     0.04881     0.40353
```

Laplacian regularization, 100 iterations, Quadtree pyramid starting guess

```
                   | Min. Val. | Max. Val. | Mean Val. |  Median   | Std. Dev. |
 ------------------------------------------------------------------------------
     True values:     -3.58206   237.56772    43.53872    18.33733    53.00369
Estimated values:      0.00000   246.00000    44.96923    18.00000    54.02688
          Errors:    -68.67423    71.94943     1.43058     0.00038     8.61643
 Absolute errors:      0.00000    71.94943     3.64835     0.85361     7.93589
 Relative errors:      0.00000     9.47873     0.10503     0.03159     0.31080
```

DISCUSSION

The example presented in this paper does not exemplify the interesting subject of *scale-dependant* phenomena; seismic wave propogation being one of them. In this case, the scaling is generally in terms of temporal frequency. Specifically, estimates of an earth property obtained by inversion of seismic data collected at different scales will generally be different (Mavko et al., 1998). Small-scale measurements (with limited spatial coverage) of various earth properties are obtainable from well logs, while larger-scale measurements with good spatial coverage are provided by surface seismic data. In theory, the physics governing the scale dependance of the problem is known; I believe that a stronger form of multiscale regularization can effectively constrain a joint inversion of surface seismic and well log data.

The quadtree pyramid interpolation method I presented is common sensical, but has some strong theoretical justifications as well. The operator **B** of equation 1 which conducts the experiment has a very simple form, and a similarly simple nullspace. The nullspace is simply all linear combinations of the unknown model points. Least squares regularization constrains the nullspace with a priori assumptions. The quadtree pyramid is slightly different. Recall that there always exists a bin size such that **B** of equation 1 is invertible; the model at this scale is unique. Going to the next finest scale, we can simply fill the holes with the model estimate at the coarsest scale. In this sense, we fill the nullspace optimally in the sense that the filling is done with “known” data. Analogously, in spline theory, it can be shown (Ahlberg et al., 1967) that the best estimator of the spline coefficients on a fine mesh are the coefficients on a coarser mesh.

CONCLUSIONS

I presented two multiscale methods for sparse data interpolation problems: multiscale regularization and the quadtree pyramid. Multiscale regularization produced an order-of-magnitude speedup (100 iterations versus 1000) in convergence for least squares interpolation of sparsely sampled topographical data, compared to regularization with a simple Laplacian. The quadtree pyramid produces a result which is of decent quality, with essentially no cost – roughly one iteration. When used as a starting guess for iterative solutions (simple Laplacian regularization), the quadtree pyramid result leads to a good result for ten times fewer iterations.

ACKNOWLEDGMENTS

The program used to compute statistics about the estimated models was adapted from one written by Sergey Fomel.

REFERENCES

Ahlberg, J., Nilson, E., and Walsh, J., 1967, The theory of splines and their applications: Academic Press, Inc.

Claerbout, J., 1998, Multidimensional recursive filters via a helix: Geophysics, **63**, no. 05, 1532–1541.

Claerbout, J., 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project, http://sepwww.stanford.edu/sep/prof/.

Crawley, S., 1995, Multigrid nonlinear SeaBeam interpolation: SEP–**84**, 279–288.

Crawley, S., 2000, Seismic trace interpolation with nonstationary prediction-error filters: SEP–**104**.

Fomel, S., Clapp, R., and Claerbout, J., 1997, Missing data interpolation by recursive filter preconditioning: SEP–**95**, 15–25.

Luettgen, M. R., Karl, W. C., and Willsky, A. S., 1994, Efficient multiscale regularization with applications to the computation of optical flow: IEEE Transactions on Image Processing, **3**, no. 1, 41–64.

Mavko, G., Mukerji, T., and Dvorkin, J., 1998, The rock physics handbook: Cambridge University Press.

Menke, W., 1989, Geophysical data analysis: discrete inverse theory: Academic Press.

Sava, P., Rickett, J., Fomel, S., and Claerbout, J., 1998, Wilson-Burg spectral factorization with application to helix filtering: SEP–**97**, 343–351.

Shewchuk, J. An introduction to the conjugate gradient method without the agonizing pain:. http://www.cs.cmu.edu/afs/cs/project/quake/public/papers/painless-conjugate-gradient.ps, 1994.