Stanford Exploration Project, Report 97, July 8, 1998, pages 207–216

Stanford Exploration Project, Report 97, July 8, 1998, pages 207–216

Madagascar revisited: A missing data problem

Jesse Lomask[1]

keywords: *least squares, interpolation, helix, gravity*

ABSTRACT

The Madagascar satellite data set provides images of a spreading ridge off the coast of Madagascar. This data set has two regions: the southern half is densely sampled and the northern half is sparsely sampled. The sparsely sampled region presents a missing data problem. I am using prediction-error filters estimated on the dense southern half to fill data on the sparse half. The prediction-error filters effectively spread the texture of the spreading ridge to the sparse tracks.

INTRODUCTION

To a certain extent, the surface of the ocean is a gravitational equipotential surface. Mountains and ridges beneath cause bulges in the sea surface which reflect the topograghy beneath. With each pass, the GEOSAT satellite measures thin swaths of the height of the sea-level above some reference ellipsoid. Numerous passes can be combined to create a map of the ocean floor topography.

The altimetry of the sea surface is also influenced by tidal fluctuations and currents that overwhelm the high frequency bulges that we are interested in imaging. Any two adjacent or crossing tracks are mismatched by a low or zero frequency shift.

The satellite data consists of four different sets, each of which is a one-dimensional array of tracks connected end to end. There are two sets of sparsely sampled tracks, one north flying and one south flying. There are also two sets of densely sampled tracks, again one north flying and the other south flying. The sparse tracks cover the same region as the dense tracks plus a region of equal size to the north. Figure 1 shows the geometry of the data separated into sparse tracks and dense tracks.

Using a weighted least squares approach, Ecker and Berlioux (1995) imaged the dense southern region. They used a derivative along the data tracks to remove the shifts between each pass.

I applied a similar weighted least squares approach to the entire merged data set and have begun to address the missing data problem of the sparse tracks. I

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

found that when prediction-error filters (PEFs) are estimated on the dense data and applied to the sparse data, many of the linear features are carried into the missing data, illustrating that PEFs are effectively able to fill in between the tracks.

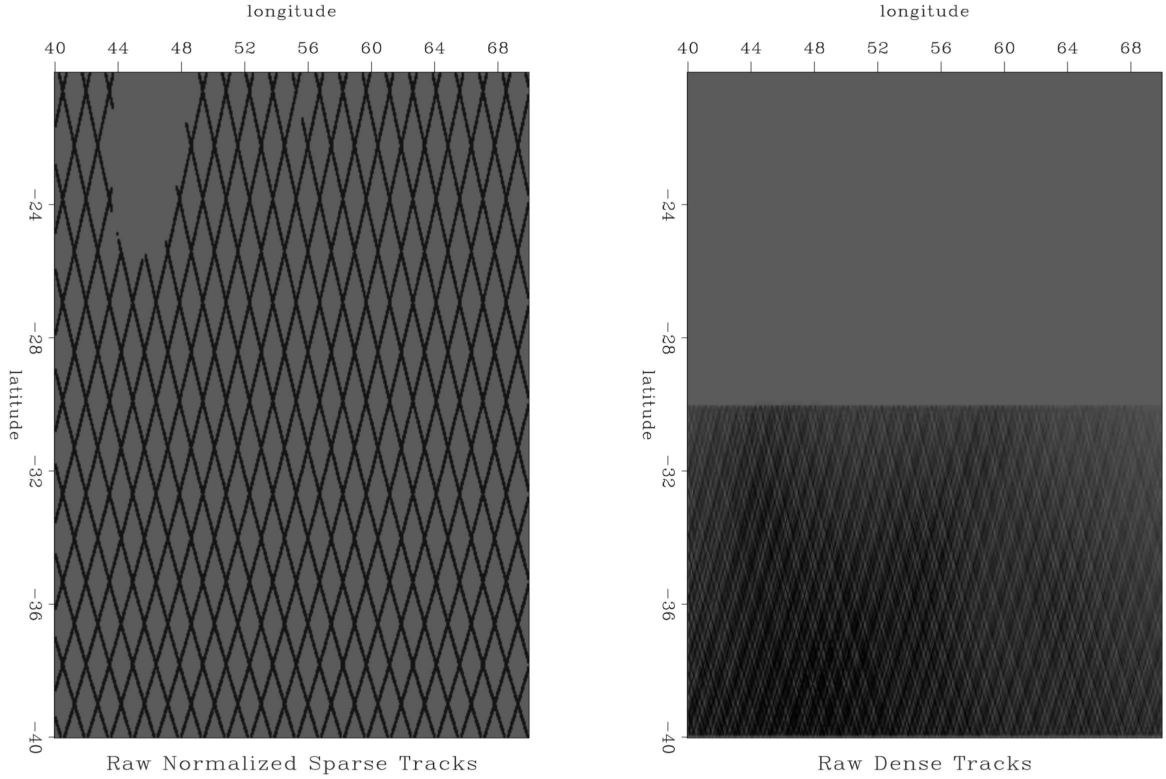

Figure 1: The raw binned data. lomask1-tracks [ER]

FITTING GOALS

All four data sets were merged into one and sorted into quadruplets of (x, y, z, w). The x and y components are longitude and latitude in degrees, respectively. The z value is sea-surface height above a reference ellipsoid in millimeters. Lastly, the w component is a weighting value that is nonzero for any suspicious z value and at the track-ends.

I used fitting goals that are essentially the same as those applied to the Sea of Galilee (Claerbout, 1997b).

$$\mathbf{W}\frac{d}{dt}[\mathbf{Lm} - \mathbf{d}] \approx \mathbf{0} \tag{1}$$
$$\epsilon\mathbf{Am} \approx \mathbf{0} \tag{2}$$

The first goal is to find a map, $\mathbf{m}$, that when sampled into data-space by linear interpolation, has a derivative that is equal to the derivative along the tracks of the observed data, **d**. The weighting term **W**, will throw out any derivative values influenced by noisy values or track-ends.

The next goal applies a regularization operator, **A**, which insures that the model with infilled missing data is smooth. I regularized with the 2D gradient, (∇_x, ∇_y), and the 2D Laplacian, ∇^2. Finally, I began testing 2D prediction-error filters (PEFs), which were estimated on the dense data.

The models in this paper have 400×400 bins which seem to have enough resolution to make detailed images while not slowing down the solver too much. The entire merged data set consists of 537979 samples.

Why take the derivative along the track?

In using a first derivative along the track, we assume that the tracks are shifted relative to each other by a constant value. If the satellite was traveling slow enough then tidal changes could be causing continuously changing shifts along the length of individual tracks. If this were the case, then a higher order derivative would be needed instead of the first derivative.

Observation of the convergence of the data-space residual revealed that the residual does shrink everywhere, which means that the model at least begins to match the data. However after enough iterations when the residual converges to a constant value, there still remain clumps of residual which move around in data-space with each additional iteration. Several possibilities could account for this. It could be the result of inaccuracies in the acquisition of the data or it could mean that the first derivative along the track is not the best function to use.

If the shifts between tracks were ramps rather than plateaus, then using the first derivative as described in equation (1) would create a model whose data-space derivative along track could not closely match that of the observed data. The difference between the derivative of the sampled model and the derivative along the observed data would not be shifted plateaus.

Jon Claerbout suggested calculating the 1D prediction-error filter on the rediduals to reveal the nature of the differences. If the differences were shifted plateaus, the PEF would be a first derivative. If the differences were shifted ramps, the PEF would be a second derivative. If the differences were curves, the PEF would be a higher order derivative, and so on.

To test this, I applied the above data fitting goal to get a model, **m**. I then calculated 1D prediction-error filters of different lengths along **r**=**W**[**Lm** -**d**]. The results were:

PEF of length 2: 1.000 -.997

PEF of length 4: 1.000 -.997 -.001 .003

PEF of length 10: 1.000 -.998 .0001 .373 -.372 .005 .122 -.123 -.003 -.002

These PEFs are, to high accuracy, first derivatives, proving that the first derivative along the track is the desired function to use in the fitting goal (1).

Preliminary Maps

Figures 2 and 3 display the best merged map using the Laplacian, ∇^2, and 2D gradient, (∇_x, ∇_y), as regulators.

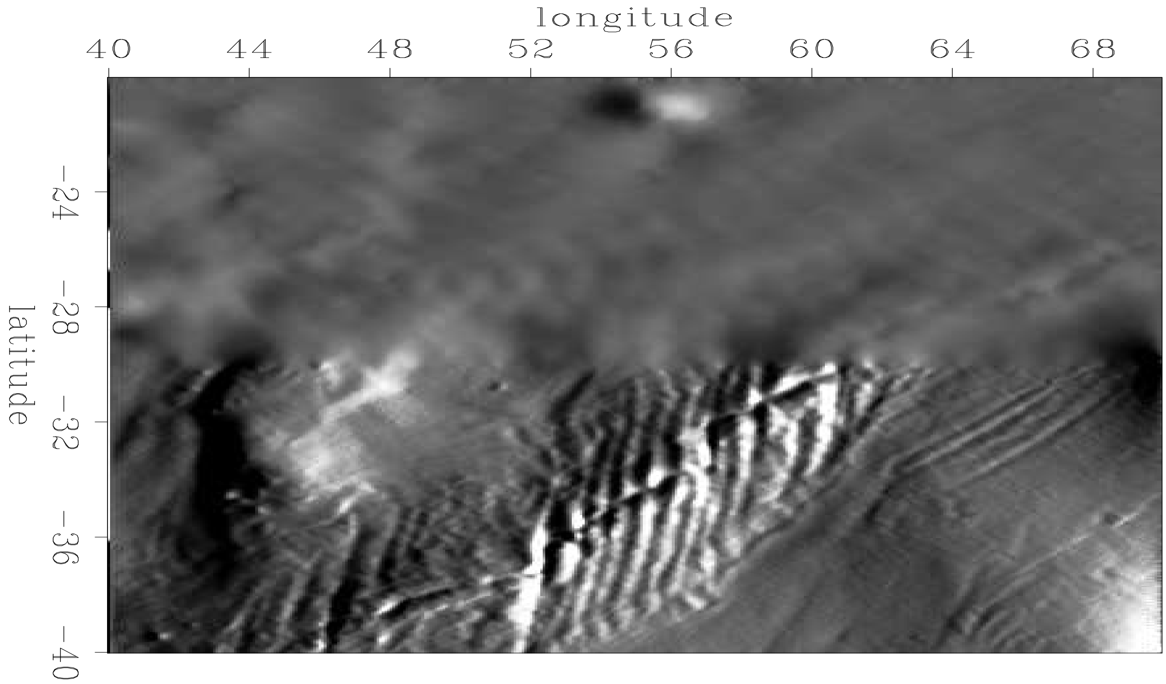

Figure 2: Dense and sparse tracks combined using the 2D Laplacian as regularization, EW roughened by gradient. lomask1-laplac [CR]

As expected, the northern half is too smooth, but the major trends are captured in both examples. The general shapes of the major ridge and the western dome features continue into the northern half. It is possible that this model with two different data densities needs two different regularization parameters, ϵ. In Figure 2, the southern half is clearly being blurred by the regularization fitting goal. The ϵ used in this case was the ϵ that created a reasonable image in the northern half. In the southern half, the best ϵ to use would actually be 0, because on this grid there is no missing data in the southern half.

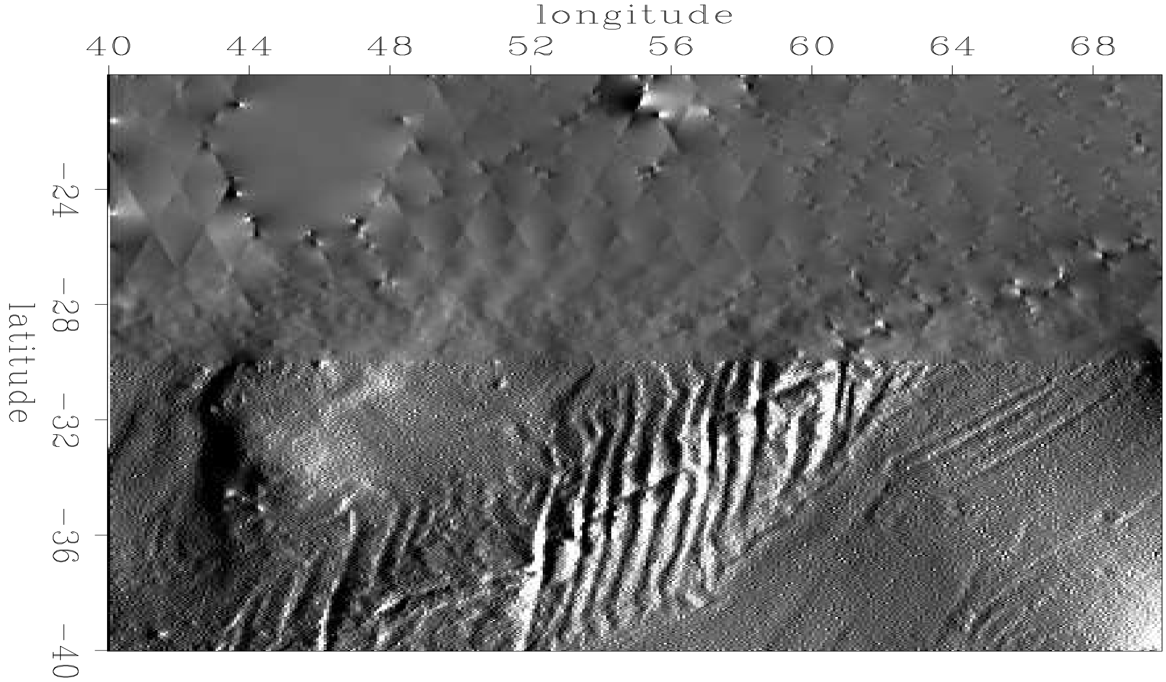

Figure 3: Dense and sparse tracks combined using the 2D gradient as regularization, EW roughened by gradient. lomask1-igrad2 [CR]

APPLYING PEFS TO FILL IN MISSING DATA

As expected, the results of using the 2D Laplacian and 2D gradient as regularizations were far too smooth to realistically represent the ocean floor between the sparse tracks. In order to capture the texture of the ridge feature in the dense tracks, we decided to estimate a prediction-error filter on the dense region and apply it using the helix as a 2D convolutional regularization.

Rather than creating a PEF on the lower portion and comparing its application to the entire sparse data set, a simpler test would be to create the PEF on the lower dense portion of the entire merged data set and apply it to the lower sparse tracks. This would permit a very straight forward comparison: the smooth model created from all the data within the dense region compared with the model from the same region using the sparse tracks only.

By looking at the smooth model constructed from the dense tracks in Figure 4, it seems that this data has two general types of texture. There is the rough lineated texture of the spreading ridge and there is the smooth texture of the ocean plane everywhere else. To compare different prediction-error filters, I created one PEF on the rough ridge feature, one PEF on the smooth areas, and one PEF over the entire dense region.

The vertical artifacts along the southern boundary are a result of polynomial division on the helix. They could be removed by padding. Both of the models

Figure 4: EW roughened template for PEF estimation, southern densely sampled area. lomask1-peftemp [ER]

Figure 5: Model, regularized by PEF, created from southern sparsely sampled data, EW roughened by gradient. lomask1-allpef [ER]

created from PEFs estimated on the ridge, Figure 6, and over the entire template, Figure 5, seem to capture some of the lineations of the spreading ridge, although the former has slightly more continuity.

The model created from the PEF estimated on the smooth area, Figure 7, does a poor job on the ridge as expected. Further, in the smooth areas, it is only slightly better than the others. This is a not a surprise because PEFs work best at spreading linear features.

CALCULATING THE REGULARIZATION PARAMETER

It is the purpose of the regularization parameter, ϵ, to weight the regularization residual so that the iterative solver does not focus on one goal while ignoring the other. For example, an ϵ that is too large will insure that the missing data is filled but it may be too smooth. On the other hand, an ϵ that is too small will not fill in much data and will tend to leave the acquisition footprint behind.

Figure 6: Model, regularized by ridge PEF, created from southern sparsely sampled data, EW roughened by gradient. lomask1-ridgepef [ER]

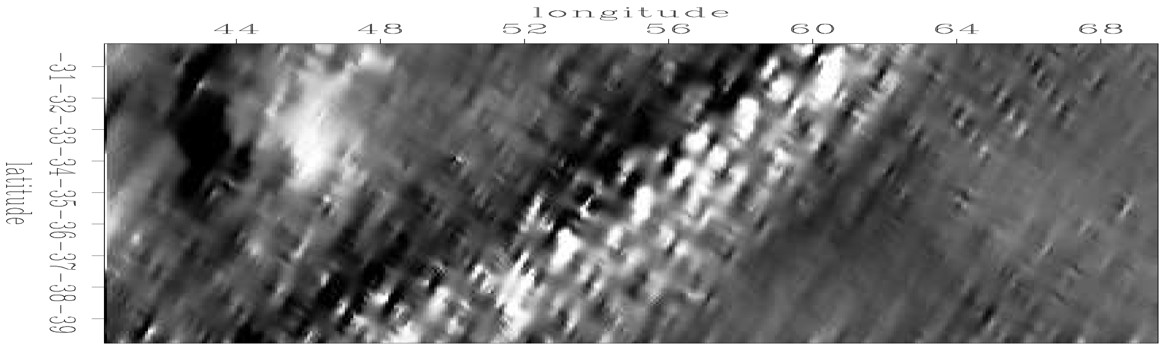

Figure 7: Model, regularized by smooth PEF, created from southern sparsely sampled data, EW roughened by gradient. lomask1-smoothpef [ER]

ϵ is used in fitting goal (2) to balance the two goals. For now, I applied Jon Claerbout's idea (1991). Within the conjugate solver routine, the gradient determines in which direction to minimize the residual.

$$\mathbf{\Delta m} = \mathbf{L}'\frac{d}{dt}\mathbf{W}'\mathbf{r_d} + \epsilon\mathbf{A}'\mathbf{r_m} \tag{3}$$

Finding an ϵ which balances both goals we try:

$$\epsilon = \frac{|\mathbf{L}'\frac{d}{dt}\mathbf{W}'\mathbf{r_d}|}{|\mathbf{A}'\mathbf{r_m}|} \tag{4}$$

In initial tests, I placed these equations in the solver and calculated a new ϵ for each iteration with the first ϵ value equal to 1. After about 15 iterations it converged to an almost constant value. This calculated ϵ was slightly lower than the ϵ that I found by trial and error. Figure 6 was generated using an ϵ of 0.4 whereas the calculated ϵ for that figure, returned by the above equation, was approximately 0.3.

The calculated ϵ does not seem to work on the entire merged data set probably as a result of different data densities. The northern sparsely sampled region needs a different ϵ than the southern densely sampled region. In this case, a scalar ϵ value is not sufficient.

FUTURE WORK

The next step to properly evaluating the PEFs is to pad the model so that the artifacts from helical polynomial division do not interfere with the ridge features. Once this is completed, the size and shape of the PEF can be tested.

Patching, creating and applying individual PEFs to sections of the model, and multigridding, incrementally decreasing the bin size to capture finer and finer features, may prove to be useful in filling in the missing data. Ultimately, a combination of the two may be even better.

If the calculated ϵ does prove to work sufficiently, it may be a good idea to implement a ϵ vector which could apply different ϵ values to different parts of the model.

Finally, since the helix is already being used to apply the regularization, preconditioning would reduce convergence time.

CONCLUSIONS

The application of weighted least squares using the first derivative effectively removes the shifts from one track to the next to create a realistic looking map of the ocean floor in the dense data. The use of PEFs to fill in the missing data looks promising as the test cases indicate.

When PEFs are applied to the sparse tracks, the calculated ϵ proposed by Jon Claerbout seems to return values close to those that I arrived at through trial and error. However, it does not seem to work well when applied to the entire merged data set when the southern region is sampled much more than the northern region. Nontheless, the calculated ϵ may provide a good starting point for choosing ϵ.

ACKNOWLEDGEMENTS

I would like to thank Jon Claerbout for his encouragement and endless supply of suggestions. I would also like to thank SEP students in general for helping me get started, in particular Bob Clapp and Morgan Brown.

REFERENCES

Claerbout, J. F., 1991, Earth Soundings Analysis: SEP–**71**, 1–304.

Claerbout, J. F., 1997, Geophysical exploration mapping: Environmental soundings image enhancement: Stanford Exploration Project.

Ecker, C., and Berlioux, A., 1995, Flying over the ocean southeast of Madagascar: SEP–**84**, 295–306.