Flattening with geological constraints

Jesse Lomask and Antoine Guitton

SUMMARY

In areas with faults and poor signal/noise ratio, where reflectors can be discontinuous from place to place, a dip-based flattening technique might not be able to appropriately track sedimentary layers. To aid the flattening algorithms, one or a few reflectors can be picked. This information can be then incorporated in our algorithms as geological constraints. In a first method, we add a model mask to a time domain solution using a Gauss-Newton approach that incorporates an initial solution. In a second method, we set the lower and upper bounds of a constrained optimization algorithm called limited memory BFGS with bounds (L-BFGS-B). Having incorporated the geological information, the flattening algorithms can accurately pick reflectors in 3D for a noisy field data example. This method is also able to pick across faults requiring a minimal amount of interpretation. Preliminary performance tests indicate that the Gauss-Newton method converges faster than the L-BFGS-B method.

INTRODUCTION

Flattening, as described here, is an efficient full-volume automatic dense-picking method for flattening seismic data. In all of the flattening methods presented thus far (Bienati et al., 1999; Lomask and Claerbout, 2002; Lomask, 2003a; Lomask et al., In press; Guitton et al., 2005) a key selling point is that they require no picking. This would be fine if all data sets had reasonably accurate estimated dips, but in the real world flattening without picking can produce results that are not perfect. Noise, both coherent and otherwise, can overwhelm the dip estimation causing reflectors in those areas to not be flat. Although certain faults with tip-lines (terminations) encased within the data cube can be flattened if a fault model is provided, faults that cut across the entire data cube cannot. Consequently, it would be useful to have the ability to add some geological constraints to restrict the flattening result in areas of poor data quality while allowing it to efficiently tackle other areas where the dips are accurate. Additionally, these constraints can be used to correlate across faults that do not terminate within the datacube.

Here, we present two flattening methods with hard constraints. The hard constraints can be manually picked horizons or individual picks. Both methods require regularization in time (or depth). In one method the hard constraints are implemented as a masked initial model within the inversion. In the other method, the hard constraints are set as lower and upper bounds of a Limited-memory BFGS with Bounds (L-BFGS-B) algorithm (Zhu et al., 1997). We envision a tool that interpretors can run once completely unconstrained, then quality control the results. The interpreter can then adjust some horizons and then run the flattening method again honoring their changes. The algorithm is fast enough so that this process can be repeated several times. In the future, computational and algorithmic improvements can result in a flattening method that is so efficient that the flattening process can be run between picks. This method also has the potential of combining other information into the flattening such as well log picks.

METHOD

The flattening method described by Lomask et al. (In press) creates a time-shift (or depth-shift) field $\boldsymbol{\tau}(x,y,t)$ such that its gradient approximates the dip $\mathbf{p}(\mathrm{x},\mathrm{y},\boldsymbol{\tau})$. The dip is a function of $\boldsymbol{\tau}$ because for any given horizon, the appropriate dips to be summed are the dips along the horizon itself. We use the dip estimation technique developed by Fomel (2002). Using the gradient operator ($\nabla = [\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y}]^T$) and the estimated dip ($\mathbf{p} = [\mathbf{p}_x \quad \mathbf{p}_y]^T$), our regression is

$$\nabla\boldsymbol{\tau}(x,y,t) \quad = \quad \mathbf{p}(x,y,\boldsymbol{\tau}). \tag{1}$$

To add regularization in the time direction, we apply a 3D gradient operator with a residual weight $\mathbf{W}_\epsilon$ that controls the amount of vertical regularization defined as

$$\mathbf{W}_\epsilon \nabla = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\epsilon} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial t} \end{bmatrix} \quad , \tag{2}$$

where $\mathbf{W}_\epsilon$ is a large block diagonal matrix consisting of two identity matrices $\mathbf{I}$ and a diagonal matrix $\boldsymbol{\epsilon}$=$\epsilon\mathbf{I}$.

For simplicity, we implicitly chain this weight operator to the gradient operator to create a new operator now defined as a 3D gradient with an adjustable weighting parameter ϵ as

$$\nabla_\epsilon \quad = \quad \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \epsilon\frac{\partial}{\partial t} \end{bmatrix} \quad . \tag{3}$$

The residual is defined as

$$\mathbf{r} \quad = \quad \nabla_\epsilon \boldsymbol{\tau} - \mathbf{p} \quad = \quad \begin{bmatrix} \frac{\partial \boldsymbol{\tau}}{\partial x} \\ \frac{\partial \boldsymbol{\tau}}{\partial y} \\ \epsilon\frac{\partial \boldsymbol{\tau}}{\partial t} \end{bmatrix} \quad - \quad \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{0} \end{bmatrix}. \tag{4}$$

We solve this using a Gauss-Newton approach by iterating over equations (5)-(7), i.e.,

iterate {

$$\mathbf{r} \quad = \quad [\nabla_\epsilon \boldsymbol{\tau}_k - \mathbf{p}(x,y,\boldsymbol{\tau}_k)] \tag{5}$$

$$\Delta\boldsymbol{\tau} \quad = \quad (\nabla_\epsilon^T \nabla_\epsilon)^{-1} \nabla_\epsilon^T \mathbf{r} \tag{6}$$

$$\boldsymbol{\tau}_{k+1} \quad = \quad \boldsymbol{\tau}_k + \Delta\boldsymbol{\tau} \tag{7}$$

} ,

where the subscript k denotes the iteration number.

We solve equation (6) using a 3D version with adjustable regularization of a method described by Ghiglia and Pritt (1998) for 2D phase unwrapping as

$$\Delta\boldsymbol{\tau} \approx \mathrm{DCT}_{3\mathrm{D}}{}^{-1} \left[\frac{\mathrm{DCT}_{3\mathrm{D}} \left[\nabla_\epsilon{}^T \mathbf{r} \right]}{J} \right], \tag{8}$$

where $\mathrm{DCT}_{3\mathrm{D}}$ is the 3D discrete cosine transform and

$$J = -2(\cos(w\Delta x) + \cos(w\Delta y)) + 2\epsilon^2(1 - \cos(w\Delta t)) + 4. \tag{9}$$

J is the real component of the Z-transform of the 3D finite difference approximation to the Laplacian with adjustable regularization.

We wish to add a model mask $\mathbf{K}$ to prevent changes to specific areas of an initial $\boldsymbol{\tau}_0$ field. This initial $\boldsymbol{\tau}_0$ field can be picks from any source. In general, they may come from a manually picked horizon or group of horizons. These initial constraints do not have

to be a continuous surfaces but instead could be isolated picks, such as well-to-seismic ties.

To apply the mask we follow the same the development as Claerbout (1999) as

$$\begin{aligned} \mathbf{0} &\approx \nabla_\epsilon \boldsymbol{\tau} - \mathbf{p} && (10)\\ \mathbf{0} &\approx \nabla_\epsilon (\mathbf{K} + (\mathbf{I} - \mathbf{K}))\boldsymbol{\tau} - \mathbf{p} && (11)\\ \mathbf{0} &\approx \nabla_\epsilon \mathbf{K}\boldsymbol{\tau} + \nabla_\epsilon (\mathbf{I} - \mathbf{K})\boldsymbol{\tau} - \mathbf{p} && (12)\\ \mathbf{0} &\approx \nabla_\epsilon \mathbf{K}\boldsymbol{\tau} + \nabla_\epsilon \boldsymbol{\tau}_0 - \mathbf{p} && (13)\\ \mathbf{0} \approx \mathbf{r} &= \nabla_\epsilon \mathbf{K}\boldsymbol{\tau} + \mathbf{r}_0 - \mathbf{p}. && (14) \end{aligned}$$

Our resulting equations are now

iterate {

$$\begin{aligned} \mathbf{r} &= \nabla_\epsilon \mathbf{K}\boldsymbol{\tau}_k - \mathbf{p}(\boldsymbol{\tau}_k) + \mathbf{r}_0 && (15)\\ \Delta\boldsymbol{\tau} &= (\mathbf{K}^T \nabla_\epsilon^T \nabla_\epsilon \mathbf{K})^{-1} \mathbf{K}^T \nabla_\epsilon^T \mathbf{r} && (16)\\ \boldsymbol{\tau}_{k+1} &= \boldsymbol{\tau}_k + \Delta\boldsymbol{\tau} && (17) \end{aligned}$$

} .

Because the $\mathbf{K}$ in equation (16) is non-stationary we cannot solve it with equation (8). Consequently, we solve it in the time domain using preconditioned conjugate gradients with equation (8) as the preconditioner(Ghiglia and Pritt, 1998) .

WEIGHTED SOLUTION

In dealing with noise and certain geological features such as faults and angular unconformities, it is necessary to apply a weight to our method. This weight is applied to the residual to ignore fitting equations that are affected by the bad dips estimated at faults. In the case of angular unconformities, it can be used to disable the vertical regularization in locations where multiple horizons converge. Our resulting weighted and constrained Gauss-Newton equations are now

iterate {

$$\begin{aligned} \mathbf{r} &= \mathbf{W}\nabla_\epsilon \mathbf{K}\boldsymbol{\tau}_k - \mathbf{W}\mathbf{p}(\boldsymbol{\tau}_k) + \mathbf{W}\mathbf{r}_0 && (18)\\ \Delta\boldsymbol{\tau} &= (\mathbf{K}^T \nabla_\epsilon^T \mathbf{W}_\epsilon^T \mathbf{W} \nabla_\epsilon \mathbf{K})^{-1} \mathbf{K}^T \nabla_\epsilon^T \mathbf{W}_\epsilon^T \mathbf{r} && (19)\\ \boldsymbol{\tau}_{k+1} &= \boldsymbol{\tau}_k + \Delta\boldsymbol{\tau} && (20) \end{aligned}$$

} .

Again, we solve equation (19) efficiently using preconditioned conjugate gradients with equation (8) as the preconditioner.

ALTERNATIVE METHOD

Alternatively, we can use the L-BFGS-B algorithm for imposing very tight constraints on the picked values of the tau field. The L-BFGS-B algorithm seeks to find a vector of model parameters τ such that we minimize

$$\min f(\tau) \text{ subject to } \tau \in \Omega, \qquad (21)$$

where

$$\tau \in \Omega = \{\tau \in \Re^N \mid l_i \le \tau_i \le u_i\}, \qquad (22)$$

with l_i and u_i being the lower and upper bounds for the model τ_i, respectively. In this case, l_i and u_i are called simple bounds. For the flattening technique, we want to minimize (Lomask, 2003b; Guitton et al., 2005; Lomask et al., In press)

$$f(\tau) = \int\int \left[\left(p_x(x,y,z;\tau) - \frac{\partial \tau}{\partial x} \right)^2 + \left(p_y(x,y,z;\tau) - \frac{\partial \tau}{\partial y} \right)^2 \right] dx\, dy, \qquad (23)$$

The L-BFGS-B algorithm combines a quasi-Newton update of the Hessian (second derivative) with a trust-region method. It has been successfully applied for flattening (Guitton et al., 2005).

Incorporating the initial τ field is trivial with the L-BFGS-B method: we simply set the bounds where an a-priori value exists:

$$\begin{aligned} l_i &= \tau_{0_i} - \alpha \times \tau_{0_i} && (24)\\ u_i &= \tau_{0_i} + \alpha \times \tau_{0_i}, && (25) \end{aligned}$$

where α is a small number (≈ 0.001). Note that the L-BFGS-B algorithm allows us to optionally activate the constraints for every point of the model space. Note that in equation 23, the objective function incorporates a smoothing in the vertical direction of the τ field as well.

RESULTS

We conducted tests of these modified flattening methods on two 3D data sets from the Gulf of Mexico and the North Sea. We only show the results obtained with the Gauss-Newton approach, the L-BFGS-B results being almost identical.

In Figure 1 is a 3D North Sea data set provided by Total. Several horizons that result from unconstrained flattening are displayed. Although we are only displaying five horizons, we have actually tracked all of the horizons in the cube. This is a key feature of flattening. Although some of the displayed horizons are well tracked, the lowest two are not. In Figure 2, we picked the lowest horizon and passed it into the flattening method as a hard constraint. We also passed the upper portion of the unconstrained flattening result in Figure 1 as constraints. Notice that the fourth picked reflector more accurately tracks its respective event.

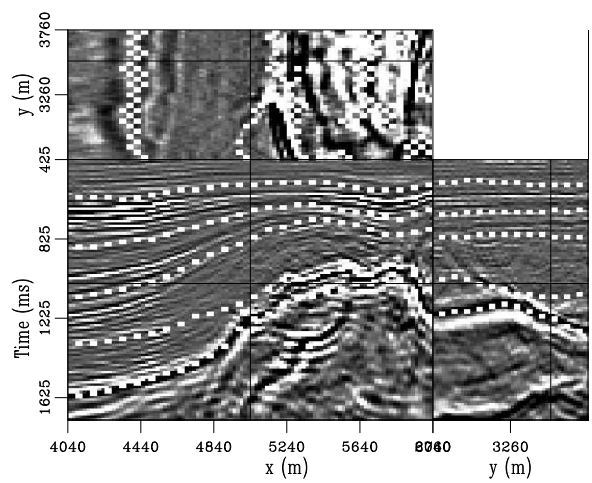

Figure 1: Image of a 3D North Sea data set displaying five tracked horizons. Although we are only displaying five horizons, these flattening methods track all of the horizons in the data cube at once. This is an unconstrained solution. Notice that the fourth displayed horizon from the top is inaccurate.

In Figure 3 is a 3D Gulf of Mexico data set provided by Chevron. The white dashed lines represent every 25th tracked horizon of the Gauss-Newton constrained flattening method. There are two faults in this data cube identified in the figure. Fault 1 has part of its tip-line encased within the cube as can be observed by its termination in the time slice. Fault 2, on the other hand, does not terminate within the data cube. Because Fault 1 terminates within the data cube, no constraints need to be provided to flatten across it, however, Fault 2 requires some picking. In this case, we picked one

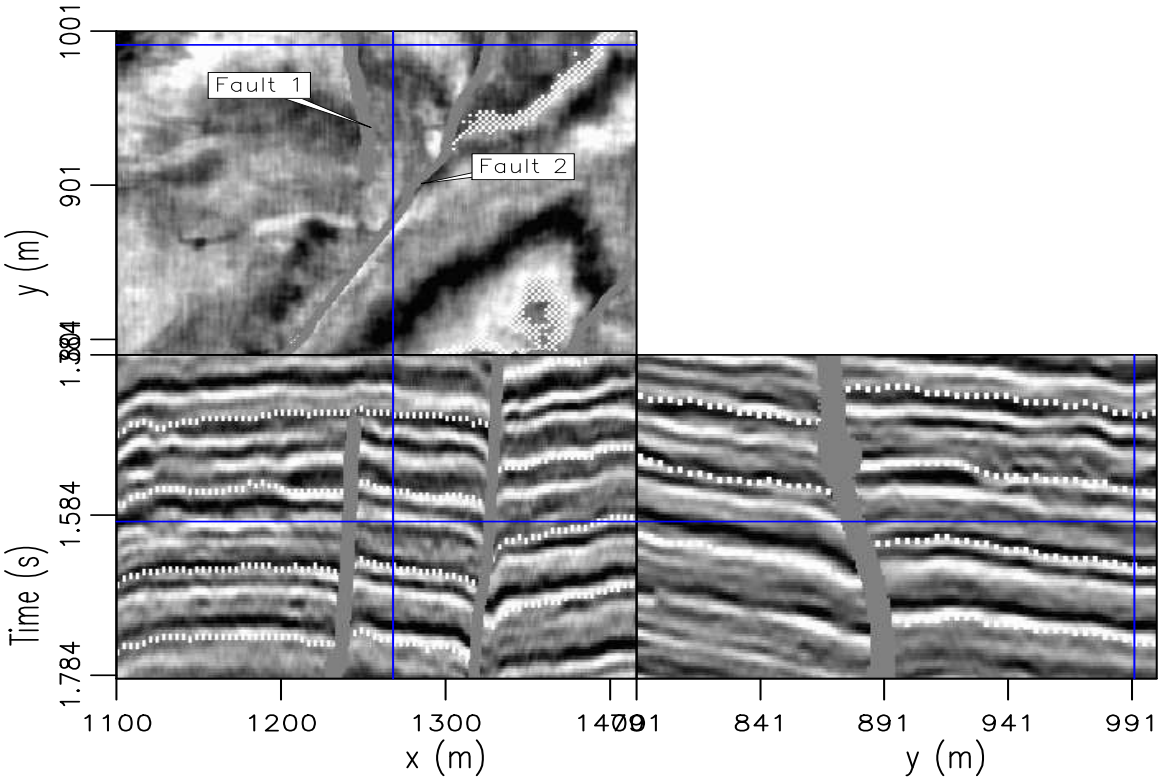

Figure 3: The dark straight lines superimposed onto the orthogonal sections identify the location of these sections: a time slice, an in-line section, and a cross-line section. The white dashed lines represent every 25th tracked horizon of the Gauss-Newton constrained flattening method.

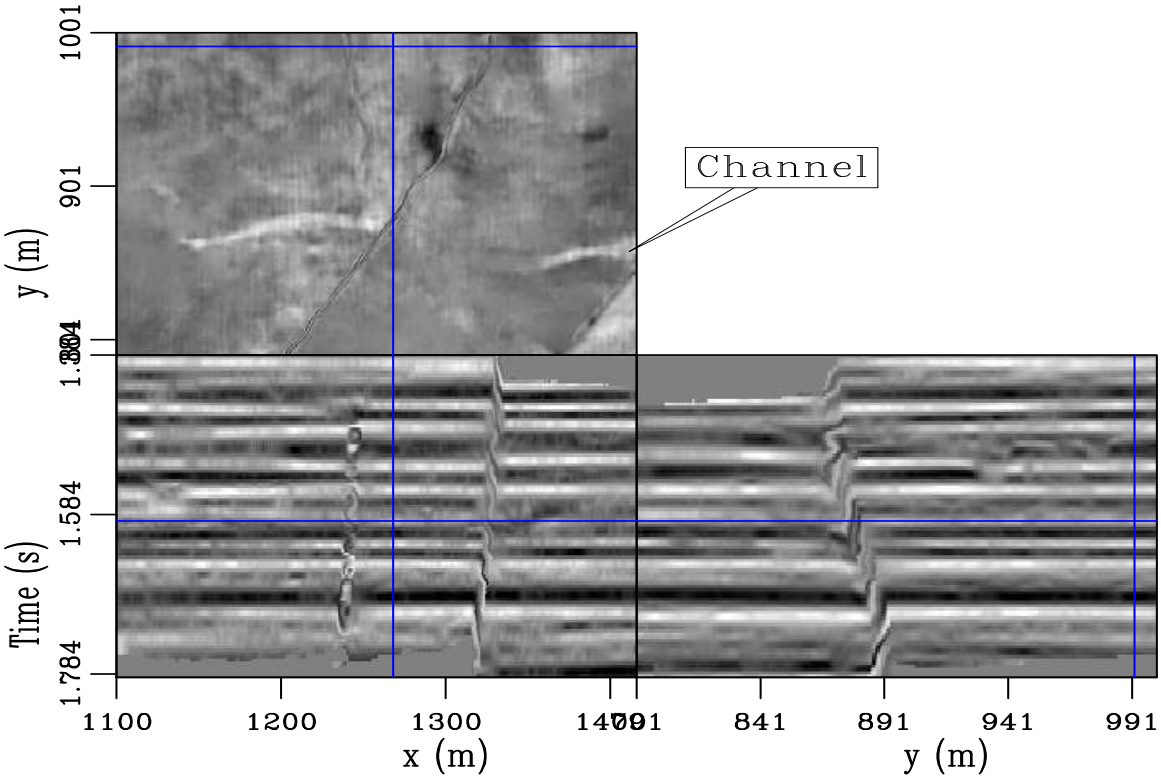

Figure 4: As Figure 3 only after flattening. Notice that reflectors on both sides of both faults are properly reconstructed. Also, notice a sinusoidal channel that is annotated on the horizon slice.

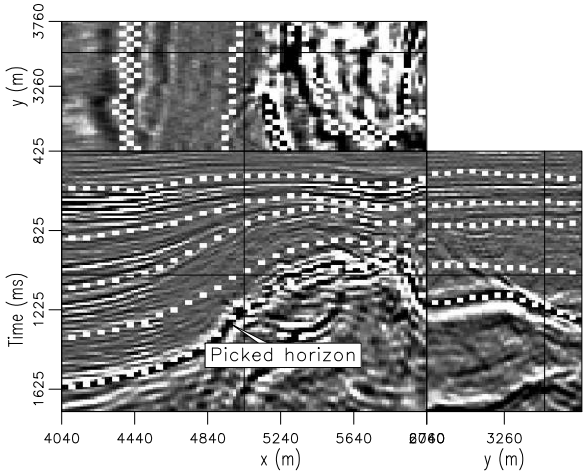

Figure 2: As Figure 1 only after flattening with constraints. The region of the top three picked reflectors in Figure 1 and the annotated picked horizon were passed as constraints to get this result. Notice that the fourth horizon from the top tracks more accurately.

vertical pair of traces across Fault 2. Then we applied the weighted Gauss-Newton method with the weight $\mathbf{W}$ being the picked fault model and the mask $\mathbf{K}$ and initial model $\boldsymbol{\tau}_0$ being the picked pair of traces correlating across Fault 2. Notice the overlain horizons track their respective events across both faults.

Figure 4 is flattened volume of the data in Figure 3. Notice the horizons are reconstructed across both faults. In addition to a fault model, the only picks required were from a single trace of correlations across Fault 2. Also, notice the faint outline of a channel that is annotated on the figure.

In short, we reconstructed this 3D volume by correlating a single pair of traces across a fault. It should be pointed out that an automatic fault indicator could substitute for the fault model, reducing the amount of manual interpretation even further.

CONCLUSIONS AND FUTURE WORK

We have added constraints to two flattening methods so that human interpretation can be incorporated into the solutions. We have successfully demonstrated their effectiveness on two 3D field data sets. Both methods converge to the same solution, the most important difference is which method converges fastest with the least amount of memory usage. Preliminary tests indicate a similar memory usage for both techniques with an advantage to the Gauss-Newton algorithm in terms of speed.

In order to add constraints, regularization is required to enforce conformity between horizons. In many geological settings this is desirable, however there are some notable exceptions such as angular unconformities. If these unconformities can be identified, it may be necessary to add a residual weight to essentially disable the regularization at those regions of the data.

As demonstrated here, the ability to incorporate some picking allows the reconstruction of horizons across faults that cut across the entire data cube. An interpreter can pick a few points on a 2D line and then flatten the entire 3D cube. With computational improvements in both the algorithm and hardware, this method could be applied on the fly, as the interpreter adds new picks.

Also, as pointed out in (Lomask, 2003a), geological features can be interpreted on the flattened horizon sections then subsequently unflattened into their original structural shape to tie with wells and other data.

The ability to use flattening in an iterative scheme is still underexploited. Once a data cube has been flattened, dips can be re-estimated on the flattened cube and then flattened again using the statistics of the dip as a measure of flatness. It seems plausible that this approach has the potential to greatly improve the quality of the flattening process. Furthermore, better flattening results can be obtained by first flattening low-passed versions of the data and gradually adding in higher frequencies.

ACKNOWLEDGMENT

We would like to thank Brian Bergquist, Chevron, and Total for supplying the data used in this paper.

REFERENCES

Bienati, N., M. Nicoli, and U. Spagnolini, 1999, Automatic horizon picking algorithms for multidimensional data: 61st European Association of Geoscientists & Engineers Meeting, Helsinki, Extended Abstracts, Session:6030.

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

Fomel, S., 2002, Applications of plane-wave destruction filters: Geophysics, **67**, no. 6, 1946–1960.

Ghiglia, D. C. and M. D. Pritt, 1998, Two-dimentional phase unwrapping: Theory, algorithms, and software: John Wiley and Sons, Incorporated.

Guitton, A., J. M. Lomask, and S. Fomel, 2005, Non-linear estimation of vertical delayx: 75th Society of Exploration Geophysicists Meeting, Extended Abstracts.

Lomask, J. and J. Claerbout, 2002, Flattening without picking: SEP–**112**, 141–150.

Lomask, J., A. Guitton, S. Fomel, and J. Claerbout, In press, Flattening without picking: Geophycics.

Lomask, J., 2003a, Flattening 3D seismic cubes without picking: 73th Society of Exploration Geophysicists Meeting, Extended Abstracts, 1402–1405.

Lomask, J., 2003b, Flattening 3-D data cubes in complex geology: SEP–**113**, 247–260.

Zhu, H., R. H. Byrd, and J. Nocedal, 1997, Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization: ACM Transactions on Mathematical Software, **23**, 550–560.

EDITED REFERENCES

Note: This reference list is a copy-edited version of the reference list submitted by the author. Reference lists for the 2006 SEG Technical Program Expanded Abstracts have been copy edited so that references provided with the online metadata for each paper will achieve a high degree of linking to cited sources that appear on the Web.

REFERENCES

Bienati, N., M. Nicoli, and U. Spagnolini, 1999, Automatic horizon picking algorithms for multidimensional data: 61st European Association of Geoscientists & Engineers Meeting, Helsinki, Extended Abstracts, session 6030.

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

Fomel, S., 2002, Applications of plane-wave destruction filters: Geophysics, **67**, 1946–1960.

Ghiglia, D. C., and M. D. Pritt, 1998, Two-dimentional phase unwrapping: Theory, algorithms, and software: John Wiley and Sons, Incorporated.

Guitton, A., J. M. Lomask, and S. Fomel, 2005, Non-linear estimation of vertical delays: 75th Annual International Meeting, SEG, Extended Abstracts, 841–844.

Lomask, J., 2003a, Flattening 3D seismic cubes without picking: 73rd Annual International Meeting, SEG, Extended Abstracts, 1402–1405.

———, 2003b, Flattening 3-D data cubes in complex geology: SEP, **113**, 247–260.

Lomask, J., and J. Claerbout, 2002, Flattening without picking: SEP, **112**, 141–150.

Zhu, H., R. H. Byrd, and J. Nocedal, 1997, Algorithm 778: LBFGS-B, FORTRAN routines for large scale bound constrained optimization: Transactions on Mathematical Software, **23**, 550–560.