Flattening 3-D seismic cubes without picking

Jesse Lomask, Stanford University

SUMMARY

As a seismic interpretation aid, I present different methods to convert an entire 3D data cube into a cube of horizon slices without picking. These methods essentially sum local dip estimates into total time-shifts. Thus far, an efficient Fourier-based method that sums the dips along time slices to flatten unfaulted data has been developed and tested on a simple field dataset. Another method in the time-space (T-X) domain that flattens vertically faulted data, again by summing along slices, has been developed and tested on a simple sythetic data. This T-X domain method requires a fault model as input. A computationally efficient and practical way of globally flattening 3-D data volumes in the presense of noise, non-vertical faults, and unconformities can be developed.

INTRODUCTION

In spite of numerous advances in computational power in recent years, interpretation still requires a lot of manual picking. One of the main goals of interpretation is to extract from the seismic data geological and reservoir features. One commonly used interpretation technique that helps with this effort is to flatten data on horizons [e.g. Lee (2001)]. This procedure removes structure and allows the interpreter to see geological features as they were emplaced. For instance, after flattening seismic data, an interpreter can see in one image an entire flood plain complete with meandering channels. However, in order to flatten seismic data, a horizon needs to be identified and tracked throughout the data volume. If the structure changes often with depth, then many horizons need to be identified and tracked. This picking process can be time consuming and expensive.

I present a method for automatically flattening entire 3-D seismic cubes with minimal picking, if any. My method involves first calculating dips everywhere in the data. For unfaulted data, the local dips are resolved into time shifts via a least-squares problem that is solved quickly in the Fourier domain. For faulted data, weights are applied and the least-squares problem is solved in time-distance (T-X) space. The data is subsequently shifted according to the travel times to output a flattened volume. Bienati et al. (1999a,b); Bienati and Spagnolini (2001, 1998) use a similar approach to numerically resolve the dips into time shifts for the purpose of auto-picking horizons and flattening gathers, yet without flattening the full volume at once. As with amplitude based auto-pickers, amplitude variation also affects the quality of the dip estimation, and will, in turn, impact the quality of this flattening method. However the effect will be less significant because this method flattens the entire data cube at once, globally, in a least-squares sense, minimizing the effect of questionable dip information. Additionally, this should make the method more robust in noisy data or complicated structures. Once a seismic volume is flattened, automatic horizon tracking becomes a trivial matter. If necessary, horizons can then be unflattened and tied to wells.

METHODOLOGY

The basic idea is similar to phase unwrapping (Claerbout, 1999) but instead of summing phase differences to get total phase, dips are summed to get total time shifts that are then used to flatten the data. To apply the shifts, the central trace is held constant as a reference and all other traces are shifted vertically to match it.

The first step is to calculate dips everywhere in the 3-D seismic cube. Thus far only a simple dip estimation method has been used. Dip can be easily calculated using a plane-wave destructor as described in Claerbout (1992) although more advanced dip estimations techniques (Fomel, 2001) will later be employed. For each point in the data cube two components of dip, p_x and p_y, are estimated in the x direction and y direction, respectively. These can be represented everywhere on the a mesh as vectors as $\mathbf{p}_x$ and $\mathbf{p}_y$.

The most basic flattening approach is to integrate dips on each time-slice in the data cube to get total time shifts ($\mathbf{t} = t(x,y,z)$). The gradient ($\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$) of the time shifts can be related to the estimated dip ($\mathbf{p} = (\mathbf{p}_x, \mathbf{p}_y)$) in an overdetermined system with the following regression:

$$\nabla \mathbf{t} \quad = \quad \begin{bmatrix} \frac{\partial \mathbf{t}}{\partial \mathbf{x}} \\ \\ \frac{\partial \mathbf{t}}{\partial \mathbf{y}} \end{bmatrix} \approx \begin{bmatrix} \mathbf{p}_x \\ \\ \mathbf{p}_y \end{bmatrix}. \tag{1}$$

The dips are summed to find a total time shift vector using:

$$\nabla^2 \mathbf{t} \quad \approx \quad \nabla' \mathbf{p} \tag{2}$$

where ∇^2 is the Laplacian and ∇' is the divergence. Solving this equation for time-slices in both the Fourier domain and time-space domain is the basic flattening method described in the following sections.

UNFAULTED DATA

Continuous reflections that are invariant with depth, represent the most basic challenge to flattening. All the information required to flatten such seismic cubes is contained within each time-slice because the dip is not changing with depth. If the dip does vary with depth then each time-slice does not contain all the necessary information required to flatten the data the data and, as a result, the flattening process may need to be applied again.

Integrating dips in the Fourier domain

Using the integration method described below (Lomask and Claerbout, 2002), I first apply the divergence (∇') to the dip ($\mathbf{p}$). Then I convert to Fourier space where I integrate twice by dividing by the Laplacian. Then I convert back to the time domain. The resulting $\mathbf{t}$ can be thought of as the time shifts to apply to each point in the data to flatten it.

The flattening corrections for a 3-D volume can be generated in two different ways. In 2-D integration, the dips across each time-slice are integrated separately. In 3-D integration, the dips for all slices are integrated at once.

Integrating time slices independently: Beginning with input dip information across each horizon I have:

$$\mathbf{p} \quad = \quad \left(\mathbf{p}_x, \mathbf{p}_y\right). \tag{3}$$

The analytical solution to equation (2) is found with:

$$\mathbf{t} \quad \approx \quad \mathrm{FFT_{2D}}^{-1} \left[\frac{\mathrm{FFT_{2D}}\left[\nabla' \mathbf{p}\right]}{-Z_x^{-1} - Z_y^{-1} + 4 - Z_x - Z_y} \right] \tag{4}$$

where $Z_x = e^{iw\Delta x}$ and $Z_y = e^{iw\Delta y}$.

The chief stumbling block for this approach is that the zero frequency component is neglected from the denominator of equation (4). This means that each time slice has a constant shift applied to it. It works out that this shift is equal to the average absolute time value. The danger here is if there is any noisy dip values in one slice not present in adjacent slices, then the time correction to flatten a volume of data may actually juxtaposition data values. One way to prevent this is to integrate in 3-D (see below). Another way would be to adequately smooth the dip values.

Integrating all time slices at once: Beginning with my input dip data:

$$\mathbf{p} \quad = \quad (\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_t) \tag{5}$$

where $\mathbf{p}_t$ is all ones for smoothness in time (explained below).

The analytical solution is found with:

$$\mathbf{t} \quad \approx \quad \mathrm{FFT_{3D}}^{-1}\left[\frac{\mathrm{FFT_{3D}}\left[\nabla'\mathbf{p}\right]}{-Z_x^{-1}-Z_y^{-1}-Z_t^{-1}+6-Z_x-Z_y-Z_t}\right] \tag{6}$$

where $Z_x = e^{iw\Delta x}$, $Z_y = e^{iw\Delta y}$, and $Z_t = e^{iw\Delta t}$.

The denominator is the Z-transform of the 3-D Laplacian. The zero frequency term of the Z-transform of the denominator is neglected. This means that the resulting surface in space will have an unknown constant shift applied to it. However, by adding the t dimension and assuming the gradient in the t direction to be all ones, I am insuring that the integrated time varies smoothly in the t direction.

Integrating in three dimensions enforces vertical smoothness. Having a smooth output is beneficial if I plan to iterate on the result to remove any residual structure caused by dip changing with depth. The dip in the t direction is all ones. This can be thought of intuitively as imagining that the dip in the x direction is the derivative of x with respect to t. So dip in the t direction is the derivative of t with respect to t, therefore it is always one. By integrating in 3-D, I prevent my method from swapping sample positions in time. Unfortunately, by preventing swapping of sample values this method cannot flatten data with overturned reflections.

3-D field data - test case

Figure 1 is a field 3-D data cube plot from the Gulf of Mexico provided by ChevronTexaco. It consists of almost flat horizons that have been warped up around a salt piercement. Numerous channels can be seen in time slices. In the time slice at the top of Figure 1 a channel is observed snaking across along the south side.

Figure 2 shows the flattened output of the ChevronTexaco data using the method described in equation (6). Notice that the horizons are flatter than those of Figure 1. Also, notice that the horizon slice in Figure 2 does not have the low frequency banding prominent in the time slice in Figure 1. Lastly, notice that the salt dome appears to be more localized in the horizon slice of Figure 2. This indicates that the layers warped up by the salt have been made flatter.

FAULTS

To handle faults, I will have to leave the Fourier domain behind. The Fourier based method will estimate erroneous dips across faults. It will then try to honor these erroneous dips creating a result that behaves erratically. However, in the time-space domain, I should be able to handle all faults that have at least one half of the fault tipline within the data cube. My approach is to create a masking operator ($\mathbf{W}$) that will throw out dip estimates along faults. The method will try to remove all deformation except at the faults where it will allow complete slippage.

I want to find the time shifts $\mathbf{t}(x,y)$ such that their gradient is the dip $\mathbf{p}(x,y)$. This sums across time-slices and is similar to equation (4). A time-space equivalent of equation (6) has also been implemented. I assume the dip $\mathbf{p}(x,y)$ is not a function of the unknown $\mathbf{t}(x,y)$ and write the fitting goal:

$$\nabla\mathbf{t} \quad \approx \quad \mathbf{p}. \tag{7}$$

This is multiplied by the masking operator ($\mathbf{W}$) to throw out fitting equations at the faults as:

$$\mathbf{W}\nabla\mathbf{t} \quad \approx \quad \mathbf{W}\mathbf{p}. \tag{8}$$

The time shifts ($\hat{\mathbf{t}}$) can be found in a least-squares sense with:

$$\hat{\mathbf{t}} \quad = \quad (\nabla'\mathbf{W^2}\nabla)^{-1}\nabla'\mathbf{W^2}\mathbf{p}. \tag{9}$$

Synthetic - test case

As seen in Figure 3, the sythetic faulted model has a fault starting from the center. The dip is constant with depth. Surfaces within this model gradually climb in the clockwise direction. This model is flattened using a Fourier space method in Figure 3. Because of erroneous dips at the fault, it does a poor job of flattening. Figure 4 shows the results of applying the T-X space approach using conjugate gradients and a weight that throws out fitting equations at the fault as in equation (9). Notice it is now well flattened.

The weights applied to the residual, as in equation (8), to throw out fitting equations at faults and to weaken fitting equations with low dip semblance or high noise can be created from rough fault models or coherency cubes (Marfurt et al., 1999).

DISCUSSION

The methods presented here adequately flattened their respective test cases. The Fourier domain methods efficiently flatten unfaulted data. The T-X based method effectively flattened the vertically faulted model.

Angular unconformities present a major challenge for this methodology. Perhaps the simplest way to handle unconformities is to break up the cube into different cubes, flatten separately, and then recombine.

Although completely separated fault blocks present an obvious challenge to the T-X method because the dip is unknown at fault discontinuities, I have some plausible solutions. The fault slip displacement would be the perfect dip information at a fault. If you have a fault block that is bounded on all sides by faults within the 3-D cube, then there is really no way to remove the deformation across the faults without at least knowing two points that correlate on each side of the fault. From an interpretation standpoint it will be necessary to restore both sides of the fault to see the best geological picture. Under such circumstances it maybe necessary to first calculate the fault contours using a method similar to that described in Lomask (2002) and put the values of the fault slip into the dip cube as an input to the flattening process.

The present procedure for applying the shifts to cubes is very simplistic and needs enhancement. The current approach is to hold the central trace constant and shift all other traces vertically to match it. However, a procedure for actually applying the shifts in the presence of numerous non-vertical faults, over-turned beds, pinch-outs, and unconformities needs to be developed.

This method still can be integrated with automatic fault tracking schemes. The output from these programs can be passed to my flattening scheme and used as fault weights. The fault contours, which are an easy output from the flattening method, can be used to quality check and improve the automatic fault tracking output.

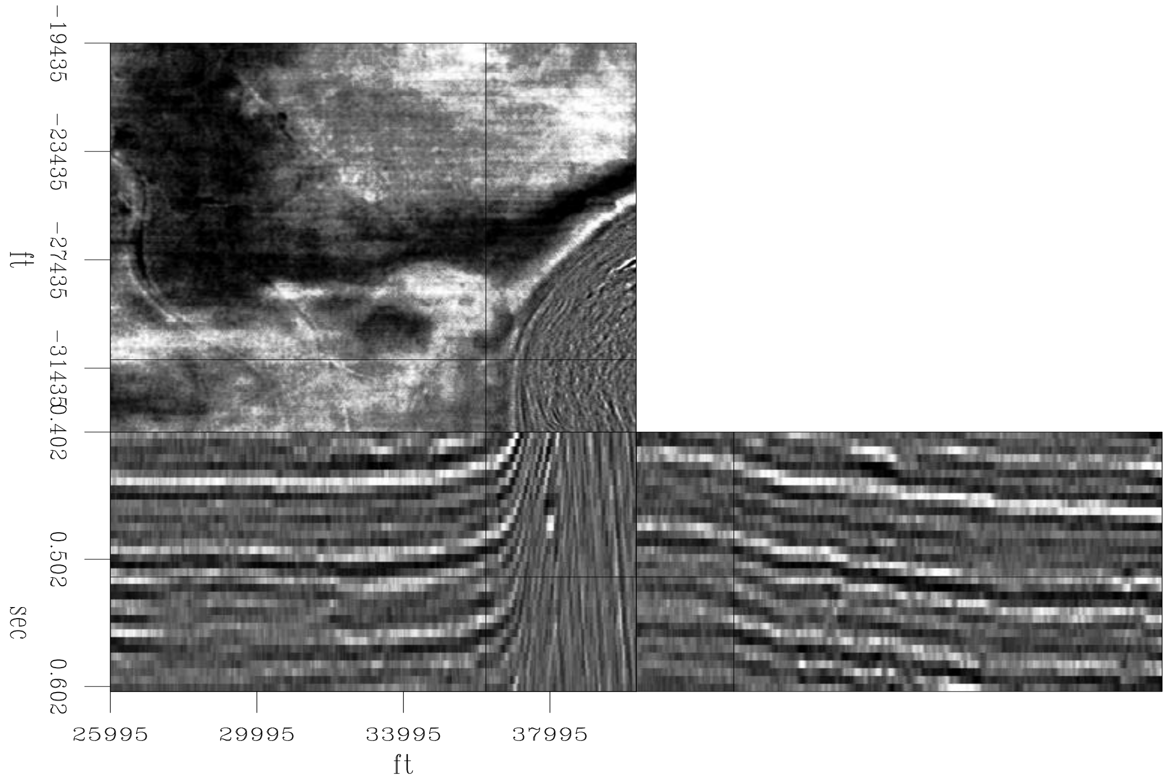

Figure 1: Unflattened ChevronTexaco Gulf of Mexico data.

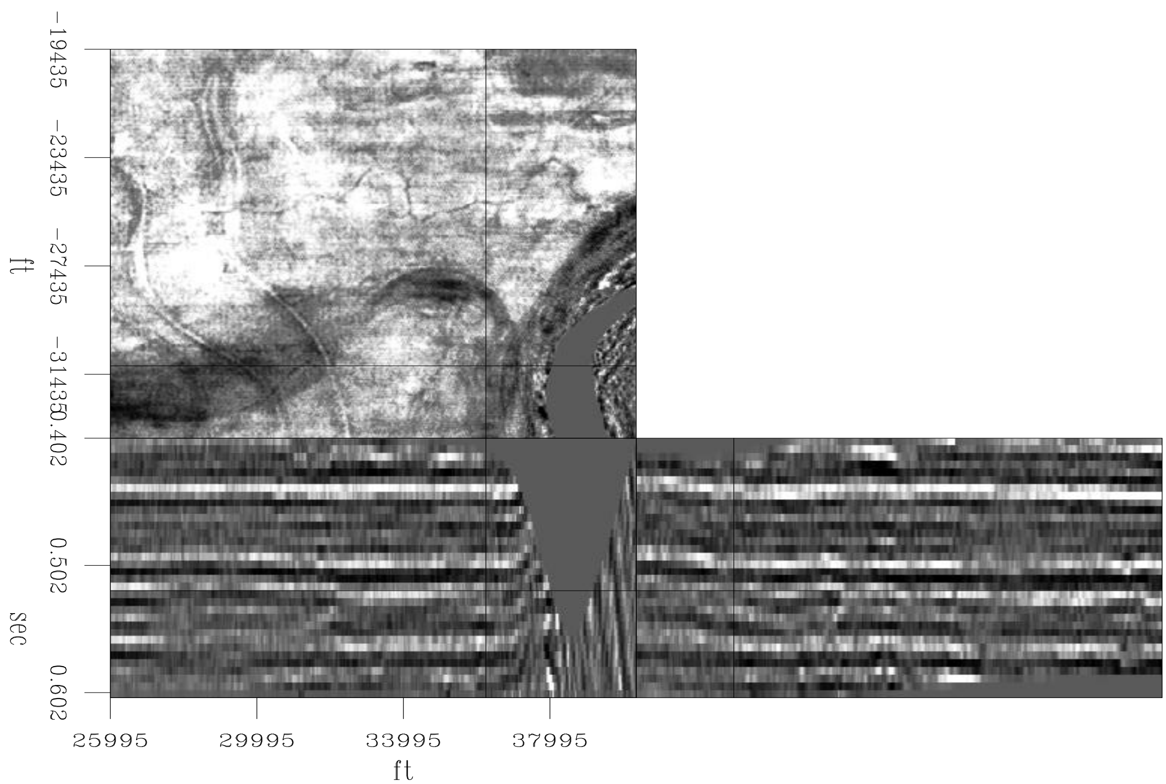

Figure 2: Result of flattening data in Figure 1.

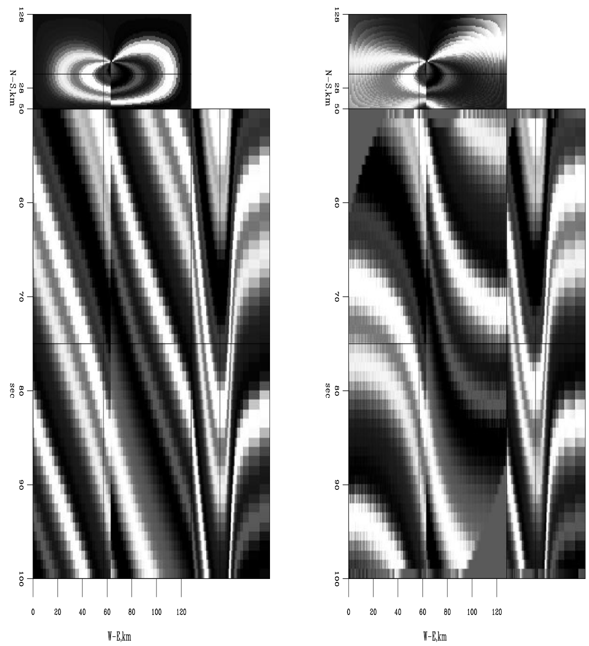

Figure 3: Faulted model. Left is unflattened. Right is flattened with the Fourier domain method. The Fourier domain method does a poor job because it uses erroneous dips at the fault.

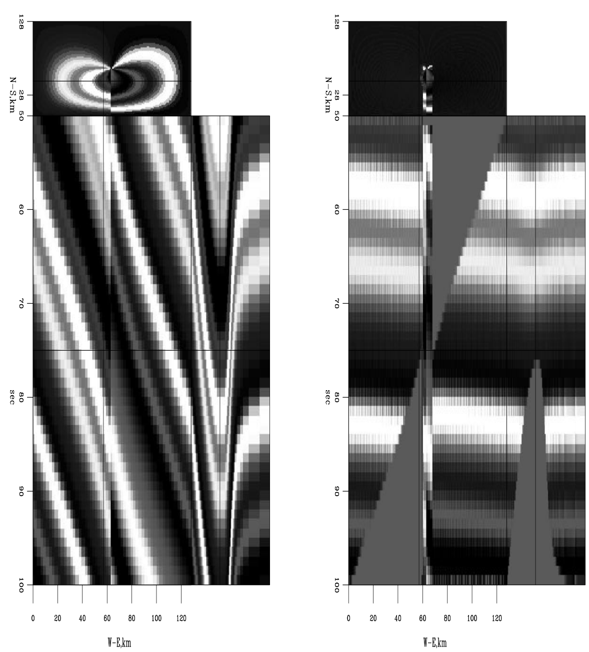

Figure 4: Left is unflattened. Right is flattened by the T-X space method rather than the Fourier method. This allows us to apply a weight to remove fitting equations corrupted by bad dips at the fault. Notice this method does a much better job at flattening.

ACKNOWLEDGEMENTS

I want to thank Jon Claerbout, Sergey Fomel, and Antoine Guitton for their helpful suggestions. I also want to express appreciation to ChevronTexaco for the Gulf of Mexico data and the SEP sponsors for their support.

REFERENCES

Bienati, N., and Spagnolini, U., 1998, Traveltime picking in 3d data volumes: 60th EAGE Meeting, Extended Abstracts.

Bienati, N., and Spagnolini, U., 2001, Multidimensional wavefront estimation from differential delays: IEEE Trans on Geoscience and Remote Sensing, , no. 3, 655–664.

Bienati, N., Nicoli, M., and Spagnolini, U., 1999a, Automatic horizon picking algorithms for multidimensional data: 61st EAGE Meeting, Helsinki, Extended Abstracts.

Bienati, N., Nicoli, M., and Spagnolini, U., 1999b, Horizon picking for multidimensional data: an integrated approach: Proc. 6th International Congress of the Brazilian Geophysical Society, Rio de Janeiro.

Claerbout, J. F., 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

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

Fomel, S., 2001, Three-dimensional seismic data regularization: Ph.D. thesis, Stanford University.

Lee, R., 2001, Pitfalls in seismic data flattening: The Leading Edge, **20**, no. 01, 160–164.

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

Lomask, J., 2002, Fault contours from seismic: SEP–**111**, 295–309.

Marfurt, K. J., Sudhaker, V., Gersztenkorn, A., Crawford, K. D., and Nissen, S. E., 1999, Coherency calulations in the presence of structural dip: Geophysics, **64**, no. 1, 104–111.