STRUCTURE ENHANCING FILTERING WITH THE STRUCTURE TENSOR

R. Morelatto and R. Biloti

email: *rodrigo.geof@gmail.com, biloti@ime.unicamp.br*

keywords: *Structure tensor, Slopes, Plane-wave destructor, Structure prediction*

ABSTRACT

The structure tensor is a very versatile tool. It can be used to detect edges, estimate coherency and local slopes. In this work we employ the structure tensor to estimate local slopes. We compare the slopes obtained with this tool with the slopes obtained by two different implementations of plane-wave destruction filters. Those three methods were tested against three different datasets, two synthetic and one real. The slopes detected through the structure tensor were reliable and comparable to the ones obtained with plane-wave destruction filters. Finally, we present an application for the slopes detected by the structure tensor. We show how to employ them to filter seismic data along structures.

INTRODUCTION

Determining local slopes is of great interest in seismic data analysis. They can be used to accomplish many of time-domain imaging tasks, like normal moveout and prestack time migration (Ottolini, 1983; Fomel, 2007c). Local slopes can also be used to interpolate data and filter along seismic structures (Fomel, 2002; Liu et al., 2010). In this work we compare the local slopes obtained via the well established method of plane-wave destruction (Claerbout, 1992) to the ones obtained using the structure tensor (Bakker, 2002).

The structure tensor was applied to seismic data analysis and filtering many times before. Bakker (2002) gives a very comprehensive description of the applications of structure tensors to seismic data filtering. They can also be used to identify and create clusters of areas of interest in seismic data (Faraklioti and Petrou, 2005) and to edge preserving smoothing by diffusion filtering of seismic data (Hale, 2009; Lavialle et al., 2007).

As noted by Bakker (2002), the amount of data to interpret has grown faster than the number of capable interpreters. Also, there are more pressure for quicker interpretation results, since risk management decisions are taken based on them. One way to ease the burden imposed on interpreters and to make automatic interpretation more reliable is to use structure oriented filtering. This procedure reduces noise and enhances reflector continuity. It also removes some subtle geological features, resulting in seismic sections easier to interpret.

Driven by those motivations, Fehmers and Höcker (2003) have proposed to use the structure tensor to perform structure oriented filtering by anisotropic diffusion. This procedure results in structure simplification and make the interpretation process more agile. Bakker (2002) also tried to address that problem by using orientation adaptive filtering and edge preserving filtering with the structure tensor. His work also features the use of the structure tensor to detect faults. In this paper we propose to study a third approach, by using structure prediction filtering (Liu et al., 2010). While Liu et al. (2010) advocate the use of plane-wave destruction to estimate dips, we propose to employ the dips detected by the structure tensor.

THE STRUCTURE TENSOR

The structure tensor is obtained by simple windowed smoothing operations and simple differentiation of the image. It is commonly used to detect lines and regions of interest in images. The structure tensor is known by different names depending on the application field: *gradient structure tensor*, *second-moment matrix*, *scatter matrix*, *interest operator* and *windowed covariance matrix* (Faraklioti and Petrou, 2005).

The first order structure tensor is obtained by a first order Taylor series expansion of the squared difference function. This function sums square differences of point-to-point image amplitudes between a fixed window W around the analysis point (x_0, t_0) and a window shifted by $(\Delta x, \Delta t)$. The squared difference function is defined as

$$E_{(x_0,t_0)}(\Delta x, \Delta t) \equiv \sum_{(i,j)\in W} w_{i,j} \left(P(x_i + \Delta x, t_j + \Delta t) - P(x_i, t_j)\right)^2, \tag{1}$$

where W is a window around (x_0, t_0), $w_{i,j}$ is are non-negative weights, and $P(x, t)$ is the image amplitude at the point (x, t). All the elements of the squared difference function are summarized in Figure 1.

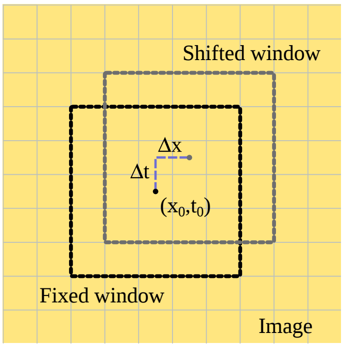


Figure 1: Parameters of the squared difference function. The black dashed square indicates the fixed window around the point (x_0, t_0), and the gray one represents the window shifted by $(\Delta x, \Delta t)$.

Function P Taylor series approximation is

$$P(x_i + \Delta x, t_j + \Delta t) = P(x_i, t_j) + \Delta x P_x + \Delta t P_t + \mathcal{O}(\|(\Delta x, \Delta t)\|^2), \tag{2}$$

where P_x and P_t are the first-order partial derivatives of P, evaluated at (x_i, t_J). For small shifts $(\Delta x, \Delta t)$, we keep only first order terms, giving rise to the approximation

$$P(x_i + \Delta x, t_j + \Delta t) - P(x_i, t_j) \approx \Delta x P_x + \Delta t P_t. \tag{3}$$

By squaring both sides of the previous equation, we have a first order approximation to the squared difference

$$\begin{aligned}(P(x_i + \Delta x, t_j + \Delta t) - P(x_i, t_j))^2 &\approx (\Delta x P_x + \Delta t P_t)^2 \\ &= (\Delta x, \Delta t) \begin{pmatrix} P_x^2 & P_x P_t \\ P_x P_t & P_t^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix}.\end{aligned} \tag{4}$$

By substituting equation (4) on equation (1), we finally obtain the first order approximation for the squared difference function

$$\tilde{E}_{(x_0,t_0)}(\Delta x, \Delta t) \equiv \sum_{(i,j)\in W} w_{i,j} (\Delta x, \Delta t) \begin{pmatrix} P_x^2 & P_x P_t \\ P_x P_t & P_t^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix}. \tag{5}$$

Since only the partial derivatives and the weights depend on (i, j), we can rewrite the previous equation as

$$\begin{aligned}\tilde{E}_{(x_0,t_0)}(\Delta x, \Delta t) &= \sum_{(i,j)\in W} (\Delta x, \Delta t) \begin{pmatrix} w_{i,j} P_x^2 & w_{i,j} P_x P_t \\ w_{i,j} P_x P_t & w_{i,j} P_t^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix} \\ &= (\Delta x, \Delta t) \begin{pmatrix} \langle P_x^2 \rangle & \langle P_x P_t \rangle \\ \langle P_x P_t \rangle & \langle P_t^2 \rangle \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix} \\ &= (\Delta x, \Delta t) \mathbf{M} (\Delta x, \Delta t)^T.\end{aligned} \tag{6}$$

The matrix $\mathbf{M}$ is known as the *structure tensor*. The symbol $\langle \cdot \rangle$ represents the average value produced by the smoothing procedure considering the weights $w_{i,j}$. The window size, in this case, is usually called *integration scale*. The local smoothing window size, used when the derivatives are calculated, is called *local scale*. The smoothing of the data is necessary to estimate reliable derivative values from noisy raw data (Faraklioti and Petrou, 2005).

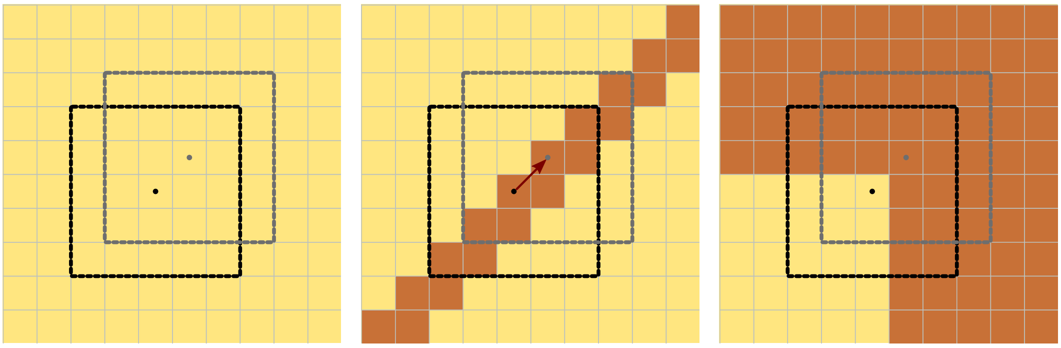
EIGENVALUES AND LOCAL IMAGE STRUCTURE

The structure tensor is clearly symmetric. It is also positive semidefinite, i.e., $(\Delta x, \Delta t)\mathbf{M}(\Delta x, \Delta t)^T \geq 0$, for all Δx and Δt. Indeed, from equations (4) and (6)

$$(\Delta x, \Delta t)\mathbf{M}(\Delta x, \Delta t)^T = \tilde{E}_{(x_0,t_0)}(\Delta x, \Delta t) \geq 0, \tag{7}$$

as long as $w_{i,t}$ are nonnegative.

Since $\mathbf{M}$ is symmetric and positive semidefinite, all its eigenvalues are real and nonnegative. The structure tensor's eigenvalues and eigenvectors can be used to detect lines, borders and regions with constant image intensity. All those scenarios, and the squared difference function, are sketched in Figure 2.



(**a**) Constant intensity. (**b**) Linear feature. (**c**) Corner.

Figure 2: Simplified version of possible image scenarios and its relation with the squared difference function. The black dashed squares represent the window centered at (x_0, t_0), while the gray dashed squares represent that window shifted by $(\Delta x, \Delta t)$. The squared difference function is the weighted sum of point-to-point image amplitude differences from the black to gray squares. In (a) $\tilde{E}(\Delta x, \Delta t) = 0$, in the vicinity of (x_0, t_0) in any direction. In (b), the red arrow indicates one direction where $\tilde{E}$ does not vary, since the amplitudes inside the regions delimited by both the black and the gray squares are the same, point-to-point. In (c) there is no direction across which both regions encompass the same amplitudes, point-to-point.

Let's start our discussion with the first case, when the image intensity is constant. In this case, we can move the gray window in any direction and the squared difference function will be close to zero due to image noise. This fact suggest that any nonzero vector $\mathbf{x} \equiv (\Delta x, \Delta t)^T$ is an eigenvector of $\mathbf{M}$, which implies that all of its eigenvalues are also close to zero.

The second case is when there is a linear feature in the image. As shown in Figure 2(b), there is only one possible direction where there is no variations of the squared difference function value. This direction is parallel to the linear feature. Recalling equation (6) again, and assuming $0 = E(\mathbf{x}) \approx \tilde{E}(\mathbf{x})$, for $\mathbf{x}$ in the direction parallel to the linear feature observed in the image we have

$$\mathbf{x}^T\mathbf{M}\mathbf{x} \approx 0. \tag{8}$$

It's possible to use that equation to show that $\mathbf{x}$ is in fact an eigenvector of $\mathbf{M}$. The first step is to consider the spectral decomposition of $\mathbf{M}$ as

$$\mathbf{M} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{9}$$

where $\mathbf{U}$ is an orthogonal matrix composed by normalized eigenvectors of $\mathbf{M}$ and $\mathbf{\Lambda}$ is a diagonal matrix, with the diagonal elements being the corresponding eigenvalues of $\mathbf{M}$. Since the tensor is a real symmetric positive semidefinite matrix, all its eigenvalues are real, so we can rewrite the decomposition as

$$\mathbf{M} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T, \tag{10}$$

where $\mathbf{\Lambda}^{\frac{1}{2}}$ denotes the element-wise square root of $\mathbf{\Lambda}$. Therefore,

$$\begin{aligned}\mathbf{x}^T\mathbf{M}\mathbf{x} &= \mathbf{x}^T\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x} \\ &= (\mathbf{x}^T\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}})(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x}) \\ &= (\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x})^T(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x}) \\ &= \left\|(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x})\right\|_2^2 \\ &\approx 0.\end{aligned} \tag{11}$$

Thus, the vector $\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x}$ is approximately null, since its Euclidean norm is near zero. Therefore,

$$\mathbf{M}\mathbf{x} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T\mathbf{x}) \approx 0. \tag{12}$$

Then it's possible to conclude that $\mathbf{x}$ is near to an eigenvector of $\mathbf{M}$, associated with an eigenvalue close to zero. The other eigenvalue is greater than zero, because it corresponds to the eigenvector orthogonal to the linear feature. In the corner case, one can not find a direction without variations of the squared difference function, as depicted in Figure 2(c). So, both eigenvalues will be much greater than zero. The expected behavior for each one of those scenarios is summarized in Table 1.

Local structure	Eigenvalues
constant intensity	$\lambda_1 \approx \lambda_2 \approx 0$
line	$\lambda_1 \gg \lambda_2 \approx 0$
corner	$\lambda_1 \gg 0, \lambda_2 \gg 0$

Table 1: Local structure conditions and expected relationships between eigenvalues of the structure tensor matrix (Faraklioti and Petrou, 2005).

The eigenvalues of the matrix $\mathbf{M}$ are the roots of the characteristic equation

$$\lambda^2 - \left(\left\langle P_x^2\right\rangle + \left\langle P_t^2\right\rangle\right)\lambda + \left\langle P_x^2\right\rangle\left\langle P_t^2\right\rangle - \left\langle P_xP_t\right\rangle^2 = 0. \tag{13}$$

Both eigenvalues can be easily found by solving the afore mentioned equation. Its solutions are given by

$$\lambda_{1,2} = \frac{1}{2}\left(\left\langle P_x^2\right\rangle + \left\langle P_t^2\right\rangle \pm \sqrt{\left(\left\langle P_x^2\right\rangle + \left\langle P_t^2\right\rangle\right)^2 - 4\left(\left\langle P_x^2\right\rangle\left\langle P_t^2\right\rangle - \left\langle P_xP_t\right\rangle^2\right)}\right). \tag{14}$$

From equation (14) we can note that both eigenvalues also satisfy the relation $\lambda_1 \geq \lambda_2 \geq 0$. In order to avoid loss of significant digits, it is wise to compute the eigenvalues as

$$\lambda_1 = \frac{1}{2}\left(\left\langle P_x^2\right\rangle + \left\langle P_t^2\right\rangle + \sqrt{\left(\left\langle P_x^2\right\rangle + \left\langle P_t^2\right\rangle\right)^2 - 4\left(\left\langle P_x^2\right\rangle\left\langle P_t^2\right\rangle - \left\langle P_xP_t\right\rangle^2\right)}\right) \tag{15}$$

and

$$\lambda_2 = \frac{\left\langle P_x\right\rangle\left\langle P_t\right\rangle - \left\langle P_xP_t\right\rangle^2}{\lambda_1}. \tag{16}$$

In our computational tests, for the eigenvalues calculation, the local and integration scales were windows with 5×5 samples each. The amplitudes squared differences were weighted by

$$w_{i,j} = \exp\left(-\frac{i^2 + j^2}{16}\right), \tag{17}$$

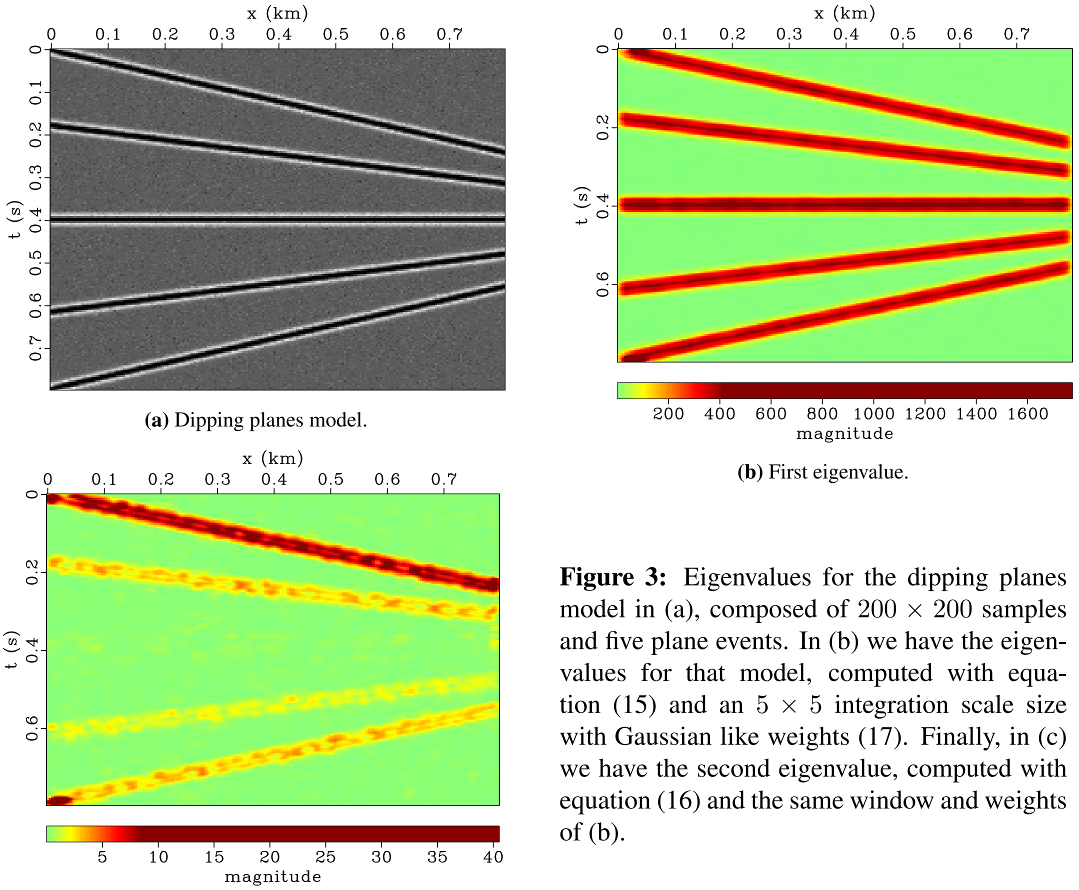
these weights have also been normalized before use.

We propose to study the tensor properties using three different datasets. The first one (Figure 3(a)) is composed of five plane events with different dips: 0.3, 0.17, 0.0, -0.17, and -0.3 s/km respectively. All planes have the same intensity and were composed of Ricker wavelets with central frequency of 40 Hz. This dataset resolution is 200×200 pixels, spaced by 4 m in the x axis and 4 ms in the t axis.

The second set is a synthetic sedimentary model (Figure 4(a)). Proposed by Claerbout (1992), this dataset is composed by 200×200 pixels, spaced by 8 m in the x axis and 4 ms in the t axis. Finally, the last dataset (Figure 5(a)) is a time-migrated seismic image from a historic Gulf of Mexico dataset (Claerbout and Green, 2010). It's composed by 250×876 pixels with the time sampling of 4 ms, and spacing between considered as unitary. The data was also filtered with an AGC filter using triangular weights and half-second window.

Before we obtain the slopes or other seismic attributes, we want to take a look at the tensor eigenvalues, given by equations (15) and (16). The eigenvalues are shown in figures 3(b) and 3(c) for the dipping planes dataset, figures 4(b) and 4(b) for the synthetic sedimentary model and figures 5(b) and 5(b) for the field data.

It is important to note that the values of λ_2 for the last dataset seem to be higher at the normal faults present in data (Figure 5(c)). This behavior is not totally unexpected. If we recall the relations between λ_1 and λ_2, listed on Table 1, we can see that the faults may be considered corner points. This behavior for the eigenvalues matches with common geological interpretation intuition, considering that the faults are registered as event terminations.



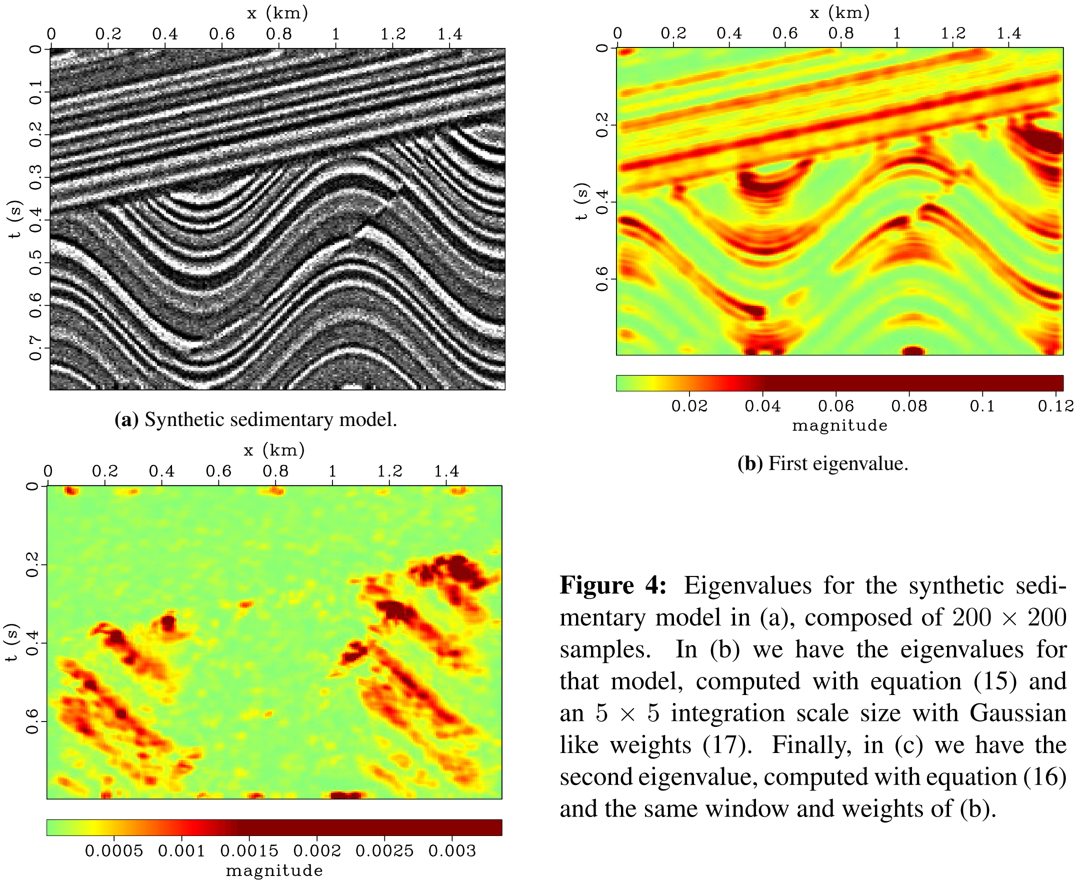
(a) Dipping planes model.

(b) First eigenvalue.

(c) Second eigenvalue.

Figure 3: Eigenvalues for the dipping planes model in (a), composed of 200×200 samples and five plane events. In (b) we have the eigenvalues for that model, computed with equation (15) and an 5×5 integration scale size with Gaussian like weights (17). Finally, in (c) we have the second eigenvalue, computed with equation (16) and the same window and weights of (b).

Let e_1 and e_2 be eigenvectors corresponding to eigenvalues λ_1 and λ_2. As discussed before, the eigenvector e_2 is parallel to the seismic image structures. We can estimate the data local slope by using the



(a) Synthetic sedimentary model.

(b) First eigenvalue.

(c) Second eigenvalue.

Figure 4: Eigenvalues for the synthetic sedimentary model in (a), composed of 200 $\times$ 200 samples. In (b) we have the eigenvalues for that model, computed with equation (15) and an 5 $\times$ 5 integration scale size with Gaussian like weights (17). Finally, in (c) we have the second eigenvalue, computed with equation (16) and the same window and weights of (b).

inclination of e_2 as

$$\sigma = \frac{\lambda_2 - \langle P_x^2 \rangle}{\langle P_x P_t \rangle}. \tag{18}$$

Since e_1 is orthogonal to e_2, we can also use it to estimate the local slope as

$$\sigma = -\frac{\langle P_x P_t \rangle}{\lambda_1 - \langle P_x^2 \rangle}. \tag{19}$$

COMPARISON OF LOCAL SLOPES ESTIMATION METHODS

In order to judge the quality of the slopes obtained with the structure tensor we need to compare its results with other methods. We choose to compare it with the well-known method of plane-wave destruction (Claerbout, 1992). We compare two different implementations of that method. For the sake of brevity we let to the reader to pursue further explanation of both methods.

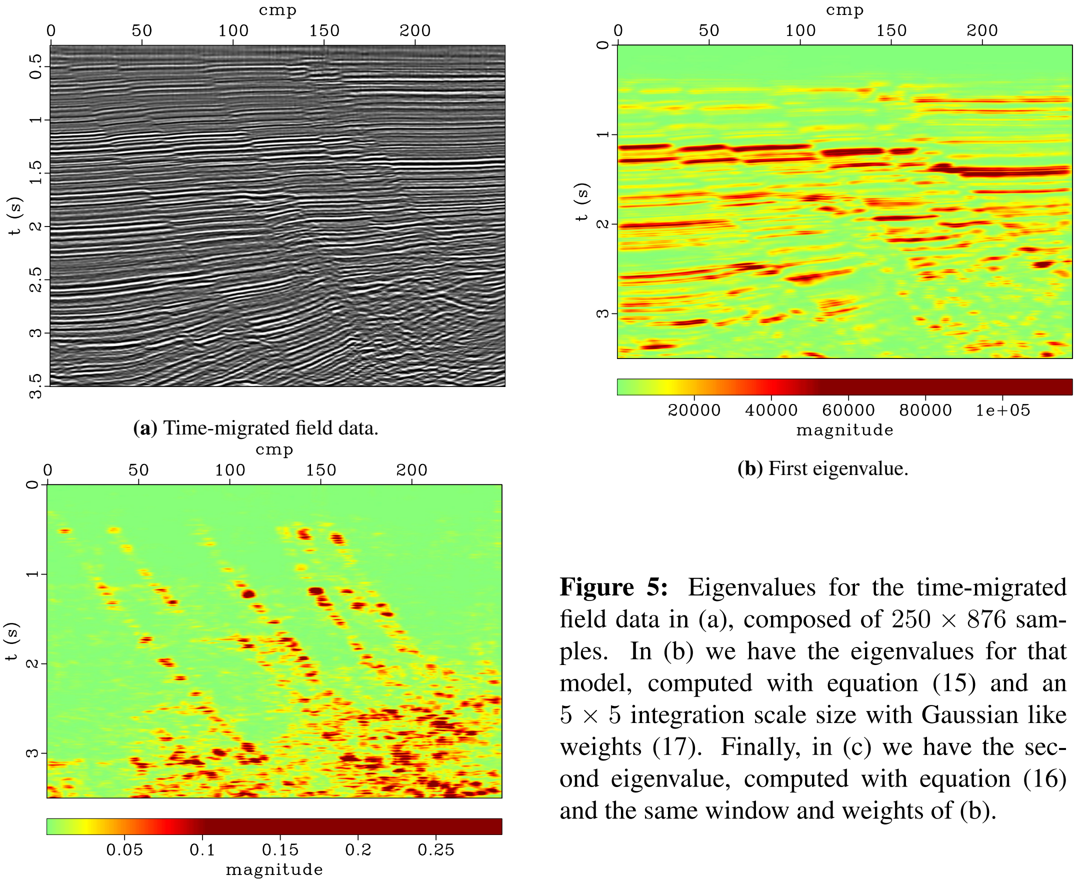
The first formulation was suggested by Fomel (2002), it treats the plane-wave filter as a time-distance (t-x) prediction-error filter. Let the local plane wave equation be given by

$$P_x + \sigma\, P_t = 0, \tag{20}$$

where P is the wave field. Assuming that the slope $\sigma(x,t)$ varies in both directions, one can design a local filter to propagate a trace to its neighbors. The filters are obtained with the help of an implicit finite-difference scheme for the local plane-wave equation.

Let the seismic section $\mathbf{s} = [\mathbf{s}_1 \ \mathbf{s}_2 \ \mathbf{s}_3 \ \ldots \mathbf{s}_n]^T$ be a collection of traces. The plane-wave destruction operation can be defined in a linear operator notation as

$$\mathbf{r} = \mathbf{D}(\sigma) s\ , \tag{21}$$



(a) Time-migrated field data.

(b) First eigenvalue.

(c) Second eigenvalue.

Figure 5: Eigenvalues for the time-migrated field data in (a), composed of 250 × 876 samples. In (b) we have the eigenvalues for that model, computed with equation (15) and an 5×5 integration scale size with Gaussian like weights (17). Finally, in (c) we have the second eigenvalue, computed with equation (16) and the same window and weights of (b).

where $\mathbf{r}$ is the destruction residual. $\mathbf{D}$ is the non-stationary plane-wave destruction operator. The previous equation leads to the system of equations

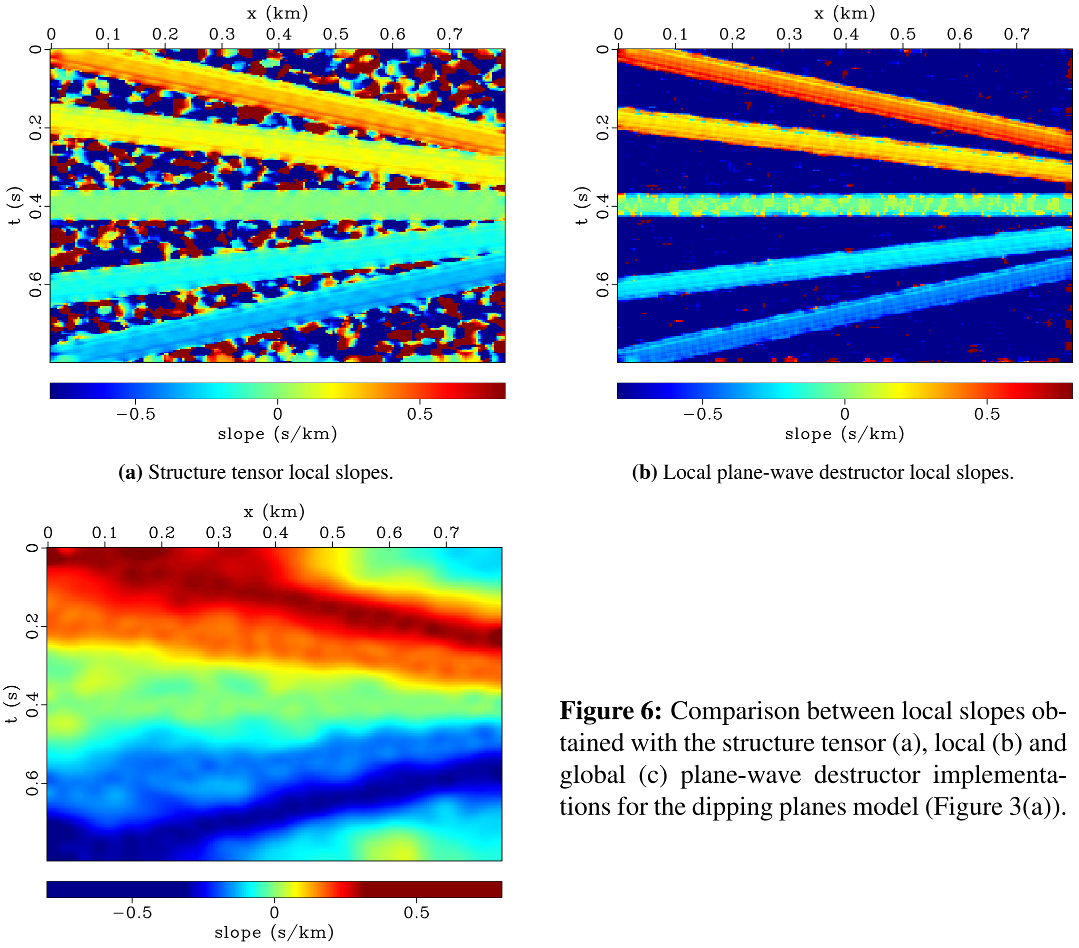
$$\begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \vdots \\ \mathbf{r}_N \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 & 0 & \cdots & 0 \\ -\mathbf{P}_{1,2}(\sigma_1) & \mathbf{I} & 0 & \cdots & 0 \\ 0 & -\mathbf{P}_{2,3}(\sigma_2) & \mathbf{I} & \ddots & 0 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & -\mathbf{P}_{N-1,N}(\sigma_{N-1}) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \\ \vdots \\ \mathbf{s}_N \end{bmatrix}, \qquad (22)$$

where $\mathbf{I}$ stands for the identity operator, σ_i is local dip pattern, and $\mathbf{P}_{i,j}(\sigma_i)$ is an operator for prediction of trace j from trace i. The destruction residual is minimized using regularized least-squares optimization. This is the essence of the method proposed by Fomel (2010) to estimate σ.

The second method, proposed by Schleicher et al. (2009), is based on a windowed fit of the data to the plane-wave equation using total least squares. The data derivatives are estimated for x and t, then the slope is obtained by fitting these derivatives to the plane wave equation inside a small window. The slope is considered to be constant in each window. In order to estimate the slope one just needs to use

$$\sigma = -\text{sign}\left(\sum_{(i,j)\in W} P_x(x_i,t_j)P_t(x_i,t_j)\right)\sqrt{\frac{\sum\limits_{(i,j)\in W} P_x^2(x_i,t_j)}{\sum\limits_{(i,j)\in W} P_t^2(x_i,t_j)}}, \qquad (23)$$

where i and j are the indices inside W window and σ is the estimated slope. The equation (23) is the total least-squares fit of equation (20) to the data derivatives inside the moving window W. The derivatives used are the same ones used for the structure tensor, both estimated using a smooth first derivative program of the Madagascar package (*sfsmoothder*) with standard parameters. On the test data, the window W is 5×5 samples in every test case. To improve the results for the local plane wave destruction implementation, its derivatives had been smoothed with the same Gaussian kernel of the structure tensor i.e. 5×5 samples suport and (17) weights.



(a) Structure tensor local slopes.

(b) Local plane-wave destructor local slopes.

(c) Global plane-wave destructor local slopes.

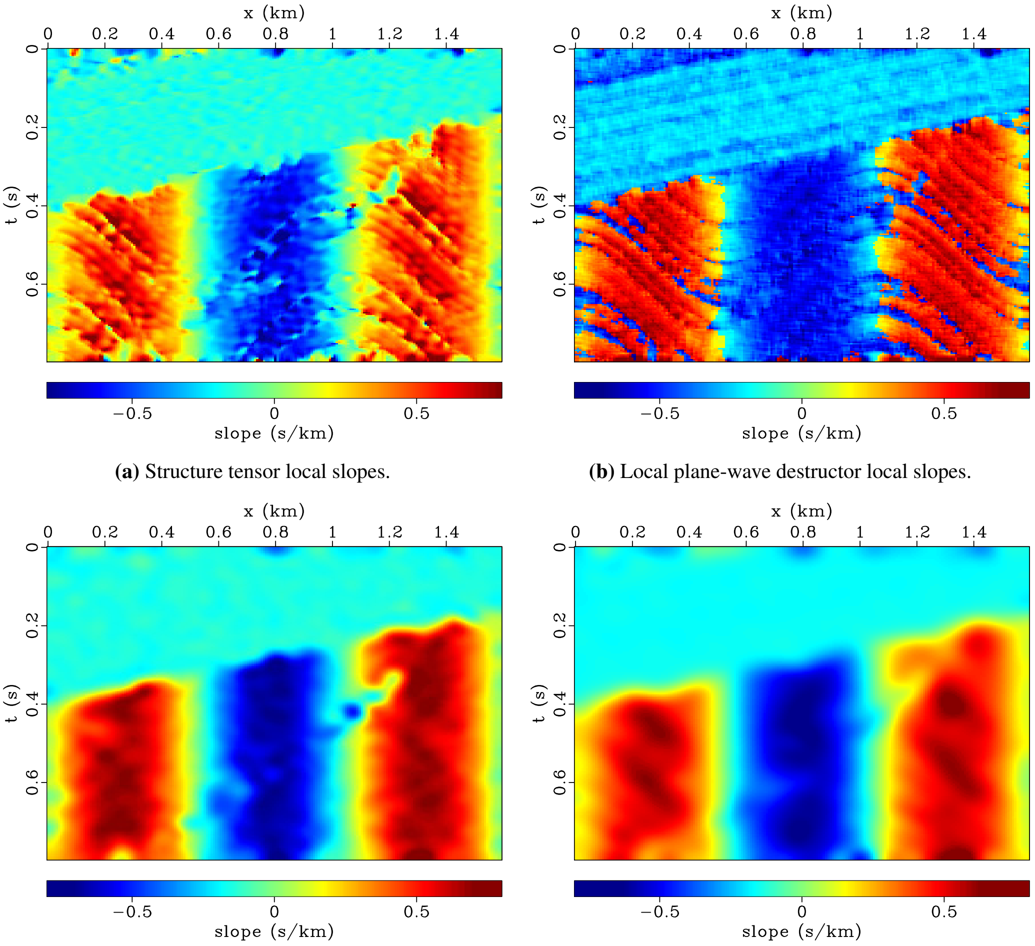
Figure 6: Comparison between local slopes obtained with the structure tensor (a), local (b) and global (c) plane-wave destructor implementations for the dipping planes model (Figure 3(a)).

In Figure 6 we compare the dips obtained by the three methods above for the dipping planes dataset. We can easily note that the smoother estimate is the one obtained by the global minimization method (Figure 6(c)). This lies on the fact that this method uses a global minimization with regularization, which forces smooth variations of dips estimations.

As seen in figures 6(b) and 7(b), the estimation based on the local implementation of the plane-wave destructor is almost visually equivalent to the structure tensor. This fact may be expected for simple test cases, since both methods have common characteristics as local support.

We also need to be careful on choosing an appropriated window size, because this method consider the events on each window as a single plane wave propagation. This may be a problem, if we want each pixel to represent just the slope of the event passing thought the center of the window. As seen in Figure 8(b), there are horizontal outlier values lines accross the figure. Tweaking the window size may solve this problem. Also, comparing figures 8(a) and 8(b), the structure tensor slope estimation seem to be a bit more smooth, without the afore mentioned outlier values lines.

One of the main advantages of the structure tensor is that each pixel corresponds mainly to the slope on the center of the window. This is due to the tensor construction, specially if we use a Gaussian like window. The structure tensor slope estimation was comparable to the global plane-wave destructor, as seen in Figure 6(a). With the advantage of being faster to run and simpler to implement.



(a) Structure tensor local slopes.

(b) Local plane-wave destructor local slopes.

(c) Global plane-wave destructor local slopes.

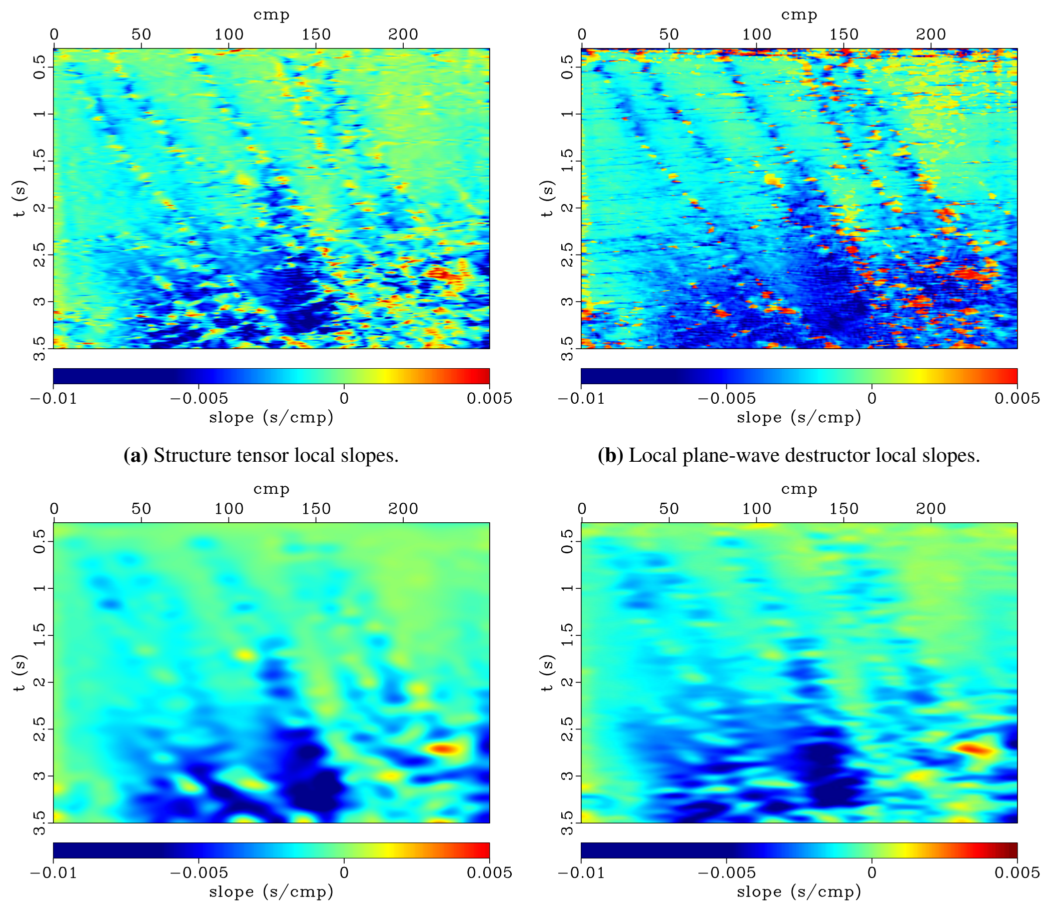
(d) Smoothed slopes estimated using the structure tensor.

Figure 7: Comparison between local slopes for the synthetic sedimentary model (Figure 4(a)). The slopes obtained with the structure tensor are shown in (a). A second version (d) of those slopes was obtained by smoothing (a) three times with a 15×15 triangular window. Finally, the local slopes for the same dataset using the local (b) and global (c) plane-wave destructor implementations.

The slopes were estimated using the first eigenvalue and the equation (18). This eigenvalue was obtained using equation (15). If calculated without proper care, λ_2 may suffer from loss of significance. By calculating it using equation (16) the dips obtained with equation (19) are equivalent to the ones obtained with λ_1 and equation (18).

By comparing figures 7(a) and 7(c), it's possible to conclude that the structure tensor slopes are a little less smooth than the slopes of the global implementation of plane-wave destruction. The slopes obtained with the structure tensor are based on sums over data derivatives. This derivatives can be a little noisy, even after applying smoothing procedures. A possible workaround is to further smooth the data before differentiation, taking care to not blur the reflector's edges too much.

Smoother slopes can be obtained by changing the local and integration scales. Greater local scale make the structure tensor ignore smaller details. The integration scale should reflect the characteristic size of the texture of interest (Weickert, 1999), in this case it should reflect the seismic events size. Instead of increasing the scale's size, we choose to smooth the slopes obtained three times with a triangular smoothing window of 15×15 samples, obtaining the slopes showed in Figure 7(d). The results of this procedure are almost visually identical to the results of the global plane wave destruction, as seen by comparing figures 7(d) and 7(c). This fact suggests that both estimations are almost equivalent, if proper smoothing is applied.



(a) Structure tensor local slopes.

(b) Local plane-wave destructor local slopes.

(c) Global plane-wave destructor local slopes.

(d) Smoothed slopes estimated using the structure tensor.

Figure 8: Comparison between local slopes for the time-migrated field data (Figure 5(a)). The slopes obtained with the structure tensor are shown in (a). A second version (d) of those slopes was obtained by smoothing (a) three times with a 15×15 triangular window. Finally, the local slopes for the same dataset using the local (b) and global (c) plane-wave destructor implementations.

STRUCTURE PREDICTION FILTERING

There are many ways to accomplish structure-enhancing filtering of a seismic image, like diffusion filtering of seismic data (Lavialle et al., 2007) or steering Gaussian elongated windows along local slope patterns (Haglund, 1991). For performance testing purposes, we choose to filter along the structures using plane-wave prediction (Liu et al., 2010). The filtering scheme is shown in Figure 9.

A trace can be predicted by shifting it according to the local seismic event slopes. Consider the prediction operator $\mathbf{P}_{i,j}(\sigma_i)$ as an operator for prediction of trace j from trace i, according to the local slope pattern σ_i (see e.g. Fomel (2002) and Fomel (2010) for further details). It's possible to predict a trace from a distant neighbor by simple recursion. So, predicting trace k from trace 1 is simply

$$\mathbf{P}_{1,k} = \mathbf{P}_{k-1,k} \cdots \mathbf{P}_{2,3}\,\mathbf{P}_{1,2}. \tag{24}$$

In this work we propose the use of the structure prediction with the dips estimated by the structure tensor, instead of using the ones estimated with plane-wave destruction. After estimating the slopes, we predict a trace from its neighbors and stack the predicted traces with the original one. In that way we accomplish the structure filtering (Liu et al., 2010).

We tested this filtering method with two datasets, the synthetic sedimentary dataset (Figure 4(a)), and the Gulf of Mexico dataset (Figure 5(a)). We used the smoothed slopes estimated with the structure tensor,

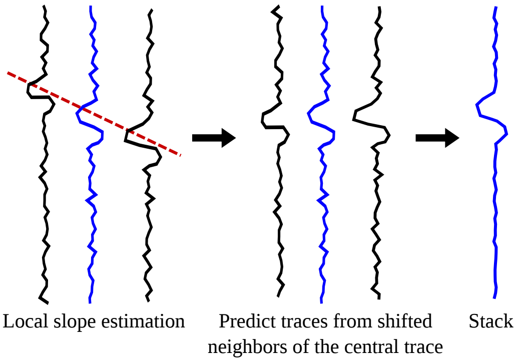
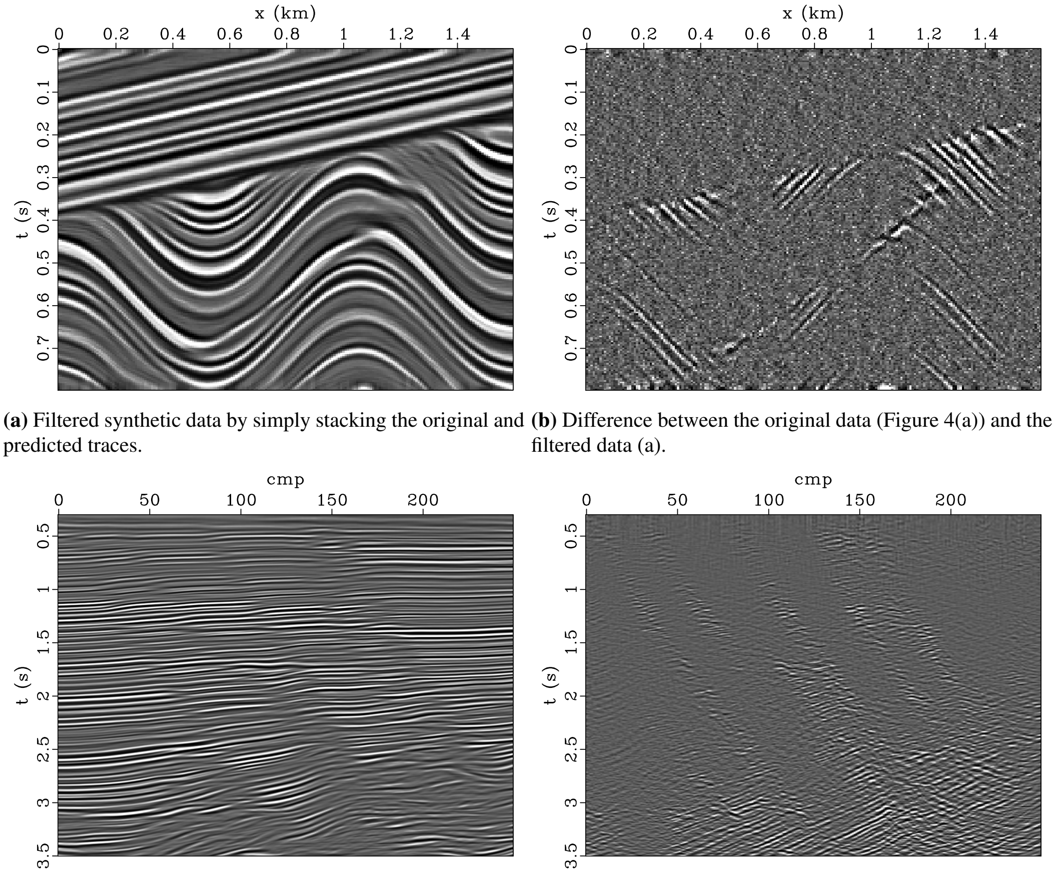


Figure 9: Prediction filtering scheme for the trace in blue. After estimate the local slopes for all points in data, the original trace can be predicted by shifting the neighbouring traces following the local slopes. In this figure, only the neighbours in the immediate vicinity are used. Distant neighbours can also be used by recursion. After the prediction step, all predicted traces are stacked with the original ones, accomplishing structure filtering.

showed in figures 7(d) and 8(d), to predict each trace from its seven nearest neighbors. At this point it is possible to accomplish structure filtering by simply stacking the predicted and original traces. By doing so, we have the filtered data showed in figures 10(a) and 10(c).



(a) Filtered synthetic data by simply stacking the original and predicted traces.

(b) Difference between the original data (Figure 4(a)) and the filtered data (a).

(c) Filtered field data by simply stacking the original and predicted traces.

(d) Difference between the original data (Figure 5(a)) and the filtered data (c).

Figure 10: Structure prediction filtering for the synthetic sedimentary and field datasets. First, each trace was predicted from its seven nearest neighbours. Then, all fourteen predicted traces and the original ones are stacked, generating the sections in (a) and (c) for the synthetic and real datasets. The difference between and the original data is shown in (b) and (d) for the synthetic and real datasets.

First, let us discuss the results concerning the synthetic dataset. The noise was clearly attenuated, but the fault and the interface between the folded layers and the plane layers was smeared. This effect is very clear when we calculate the difference between the original and filtered data (Figure 10(b)). There are also some small data loss at the ends of the folded layers. This may be due to the increased error in trace prediction, since steeper slopes may have bigger errors associated. This effect is also enhanced if the predicted traces are too far away from the original trace.

As seen in Figure 10(d), we can't see much of that effect on the real dataset. This may be due to the small slope of the seismic reflectors. What is very clear is the loss of information at the reflectors ends, blurring the normal faults present in data. Nevertheless, the results were satisfactory and the data coherence and reflectors continuity was improved, as shown in Figure 10(c). Again, this may be due to the data being compose mainly of planar like reflectors, which suits structure prediction filtering better.

SIMILARITY FILTERING WITH GAUSSIAN WEIGHTS

To prevent the blurring of data near faults and stratigraphic interfaces, we decided to improve the structure filtering by using similarity based filter weights for the stacking step (Liu et al., 2010). The basic filtering scheme is explained in Figure 11. For the similarity weights, we use the definition of local similarity proposed by Fomel (2007a). This version of local similarity is defined using shaping regularization and local correlation. Shaping regularization expands Tikhonov's regularization using a smoothing operator as the regularization operator (Fomel, 2007b). This formulation makes the similarity vary smoothly, being close to one when the two traces compared are locally similar and approaching zero when they differ.

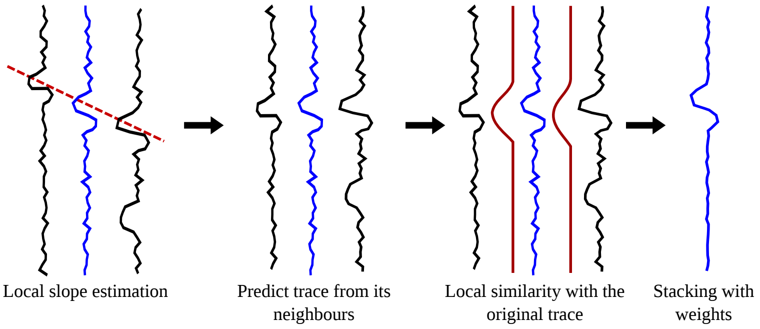
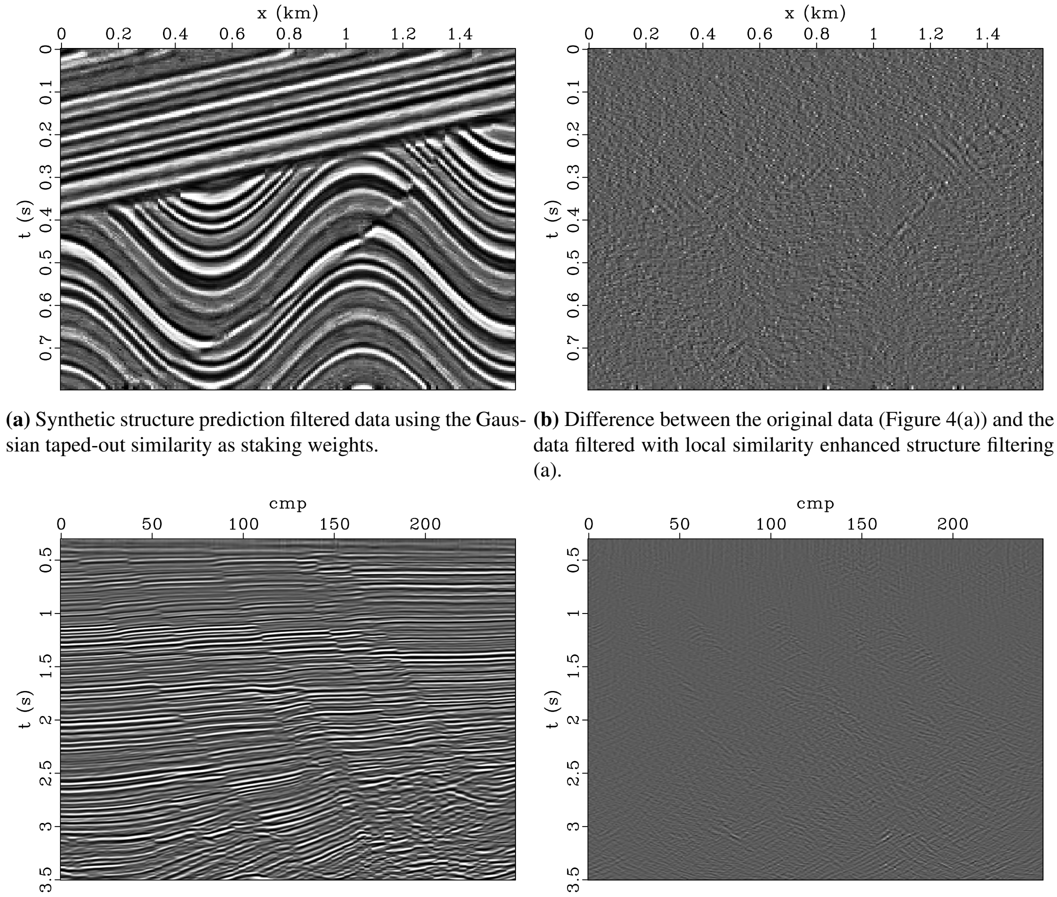

Local slope estimation
Predict trace from its neighbours
Local similarity with the original trace
Stacking with weights

Figure 11: Local similarity enhanced prediction filtering scheme for the trace in blue. After estimate the local slopes for all points in data, the original trace can be predicted by shifting the neighbouring traces following the local slopes. In this figure, only the neighbours in the immediate vicinity are used. Distant neighbours can also be used by recursion. After the prediction step, local similarity between each predicted trace and the original one is calculated. This similarity is used as stacking weights when all predicted traces are stacked with the original one. This procedure results in local similarity enhanced structure filtering.

To further improve the data staking we employed a Gaussian taper. This results lower weights in stacking for traces predicted from traces far from the original one, which diminishes some prediction errors in the stacking (Liu et al., 2010). We multiply each trace by

$$w_k = \exp\left(-\frac{k^2}{\zeta^2}\right), \tag{25}$$

where w_k is the Gaussian weight function, k is the index offset between the original and the predicted traces, i.e. for the original trace $k = 0$, for a trace predicted using an immediate neighbour $k = 1$. The parameter ζ alters the shape of the Gaussian. This approach is analogous to bilateral filtering (Tomasi and Manduchi, 1998), with the advantage of smooth variation of the similarity weights. Finally, the filtering using similarity stacking weights and $\zeta = 2$ is showed in Figure 12(a) for the synthetic data. In Figure 12(c), the same procedure was applied for the field data using $\zeta = 0.02$. We can see that the noise was attenuated, also there are very little smearing of the faults and other interfaces. This fact is further confirmed by the difference between the original data and the filtered data (figures 12(b) and 12(d)).



(a) Synthetic structure prediction filtered data using the Gaussian taped-out similarity as staking weights.

(b) Difference between the original data (Figure 4(a)) and the data filtered with local similarity enhanced structure filtering (a).

(c) Field structure prediction filtered data using the Gaussian taped-out similarity as staking weights.

(d) Difference between the original data (Figure 5(a)) and the data filtered with local similarity enhanced structure filtering (c).

Figure 12: Structure prediction filtering for the synthetic sedimentary data and field data. First, each trace was predicted from its seven nearest neighbours. Before the stacking step, the local similarity between the each predicted trace and the original ones is calculated. Those weights also receive the Gaussian taper (25), with $\zeta = 2$ for the synthetic dataset and $\zeta = 0.02$ for the field data. This procedure gives distant traces smaller weights. Finally, all fourteen predicted traces and the original ones are stacked using these weights, generating the section in (a) for the synthetic dataset and (b) for the real one. The difference between the original data and the filtered data is shown in (c) for the synthetic data and in (d) for the real dataset.

CONCLUSIONS

As discussed through this article, the structure tensor is highly correlated with the image local structure. It provides a fairly good and robust estimation for seismic data local slopes. The method worked pretty well on real data, as seen in Figure 8. The values obtained for the slopes are also very close to the ones obtained by the two plane-wave destruction methods tested. One point to keep in mind is that care should be taken on the smoothing prior to the data differentiation, to not blur features of interest, like reflector terminations. Also, one should be careful on choosing the integration scale. It should reflect the characteristic size of the features of interest (Weickert, 1999).

We also tested the slopes for structure oriented filtering. As seen in figures 12(a) and 12(c), the results were very positive, removing mostly noise from the data. These results assert the quality of the structure tensor based slopes, as also depicted in figures 8 and 7. One of the advantages of this method is the easy implementation and fast runtime, since it is basically composed of local sums over data and smoothing procedures. Those filtering results where further improved by the use of similarity weights (Figure 12) resulting in edge preserving structure oriented filtering. In the near future we intend to further test the structure tensor filtering capabilities. We intend to use different types of adaptive filters and compare those results with the results discussed above.

ACKNOWLEDGMENTS

This work was supported by the Instituto Nacional de Ciência e Tecnologia – Geofísica do Petróleo (INCT-GP), Brazil, and the sponsors of the WIT Consortium.

REFERENCES

Bakker, P. (2002). *Image structure analysis for seismic interpretation*. Ph.D. thesis, Delft University of Technology, Delft, Netherlands.

Claerbout, J. (1992). *Earth soundings analysis: Processing versus inversion*, volume 6. Blackwell Scientific Publications.

Claerbout, J. and Green, I. (2010). *Basic earth imaging*. Stanford University.

Faraklioti, M. and Petrou, M. (2005). The use of structure tensors in the analysis of seismic data. In Iske, A. and Randen, T., editors, *Mathematical Methods and Modelling in Hydrocarbon Exploration and Production*, volume 7 of *Mathematics in Industry*, pages 47–88. Springer Berlin Heidelberg.

Fehmers, G. and Höcker, C. (2003). Fast structural interpretation with structure-oriented filtering. *Geophysics*, 68(4):1286–1293.

Fomel, S. (2002). Applications of plane-wave destruction filters. *Geophysics*, 67(6):1946–1960.

Fomel, S. (2007a). Local seismic attributes. *Geophysics*, 72:A29–A33.

Fomel, S. (2007b). Shaping regularization in geophysical-estimation problems. *Geophysics*, 72(2):R29–R36.

Fomel, S. (2007c). Velocity-independent time-domain seismic imaging using local event slopes. *Geophysics*, 72(3):S139–S147.

Fomel, S. (2010). Predictive painting of 3d seismic volumes. *Geophysics*, 75(4):A25–A30.

Haglund, L. (1991). *Adaptive multidimensional filtering*. Ph.D. thesis, Linköping University, Linköping, Sweden.

Hale, D. (2009). Structure-oriented smoothing and semblance. *CWP Report*, 635.

Lavialle, O., Pop, S., Germain, C., Donias, M., Guillon, S., Keskes, N., and Berthoumieu, Y. (2007). Seismic fault preserving diffusion. *Journal of applied geophysics*, 61(2):132–141.

Liu, Y., Fomel, S., and Liu, G. (2010). Nonlinear structure-enhancing filtering using plane-wave prediction. *Geophysical Prospecting*, 58(3):415–427.

Ottolini, R. (1983). Velocity independent seismic imaging. *SEP-37: Stanford Exploration Project*, 59:68.

Schleicher, J., Costa, J., Santos, L., Novais, A., and Tygel, M. (2009). On the estimation of local slopes. *Geophysics*, 74(4):P25–P33.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE.

Weickert, J. (1999). Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2):111–127.