TVFlow regularization applied to 4D seismic inversion

Danilo Chamorro Riascos | Alan Espinos Chavez

Background

Seismic inversion is a valuable tool in the oil and gas industry for locating hydrocarbon reserves. 4D seismic inversion has emerged as a technique to monitor reservoir changes over time, allowing for better management and production optimization. However, the high dimensionality of the data makes it difficult to accurately recover subsurface properties. The ill-posed inverse problem requires regularization, such as the use of a regularization operator that incorporates prior knowledge. Our project proposal aims to implement a new regularization operator for 4D seismic inversion that can use the target's temporal evolution in a full spatio-temporal reconstruction scheme. Lucka et al. 2018 proposes a spatio-temporal modeling framework that can encode a priori information about the configuration of the problem: it formulates an explicit PDE model for the image dynamics and then jointly estimates the image sequence and the corresponding motion field by minimizing a variational energy.

Problem formulation

The goal of post-stack seismic inversion is to solve an optimization problem that retrieves the natural logarithm of the acoustic impedance model m_t for a given time t from post-stack seismic data d_t. The problem is defined as:

$$m_t^* = \underset{m_t}{\operatorname{argmin}} \left\{ \frac{1}{2} ||Gm_t - d_t||_2^2 \right\} \quad \text{Where} \quad G = WD$$

Where the post-stack modeling operator G is defined as the chain of a time derivative operator D and a convolution operator W created for a given wavelet w. When dealing within a temporal framework we can define the model m as the entire model sequence over time frames $t = 1, 2, ..., T$, with $m \in \mathbb{R}^{nT}$. A full spatio-temporal variational defined in the literature by Burger et al. 2017 scheme reads:

$$(m^*, v^*) = \underset{m,v}{\operatorname{argmin}} \left\{ \frac{1}{2} ||Gm - d||_2^2 + \alpha \mathcal{J}(m_t) + \beta \mathcal{H}(v_t) + \gamma \mathcal{M}(m, v) \right\}$$

In this context, the vector field $v_t \in \mathbb{R}^{nT}$ represents the motion between points m_t and m_{t+1} in n-dimensional space. The terms $\mathcal{J}(m_t)$ and $\mathcal{H}(v_t)$ correspond to spatial regularization terms applied to the image and motion field, respectively, while α, β, and γ are non-negative regularization parameters. The authors of the paper propose the use of Total Variation (TV) regularization over both the model and velocity since it enhances the expected blocky aspect of the solution. The crucial factor is the term $\mathcal{M}(m, v)$, which ensures a relationship between the image sequence m and the corresponding motion field sequence v. The motion term $\mathcal{M}(m, v)$ should enforce a simple continuity equation known as the optical flow equation in the field of computer vision:

$$\partial_t m(t, \vec{x}) + (\nabla_{\vec{x}} m(t, \vec{x})) \cdot v(t, \vec{x}) = 0$$

We can achieve this by letting $\mathcal{M}$ to measure the least-squares error of a forward difference discretization of the optical flow equation:

$$\mathcal{M}(m, v) = \sum_{t}^{T-1} \frac{1}{2} ||m_{t+1} - m_t + (\nabla m_t) \cdot v_t||_2^2 \tag{1}$$

Which in total leads to the variation scheme defined as TVTVL2 by the authors:

$$(m^*, v^*) = \underset{m,v}{\operatorname{argmin}} \{\mathcal{E}(m, v)\} := \underset{m,v}{\operatorname{argmin}} \Big\{ \frac{1}{2} \sum_t^T ||Gm_t - d_t||_2^2 + \\ \alpha||\nabla^+ m_t||_1 + \beta \sum_i^d ||\nabla^+ v_{x_i,t}||_1 + \frac{\gamma}{2}||m_{t+1} - m_t + (\nabla^{\pm} m_t) \cdot v_t||_2^2 \Big\} \quad (2)$$

Where ∇^+ denotes forward differences approximation and $\nabla^{\pm}$ denotes central differences approximation. This formulations renders a large-scale, nonsmooth, biconvex optimization problem. The main difficulty in the implementation is the motion term since the product $(\nabla^{\pm} m_t) \cdot v_t$ has a biconvex behavior: if we fix one of the variables the problem is convex but it's non-convex as a function of the two variables. The **alternate convex search (ACS)** algorithm is proposed to tackle this issue since it allows to alternate between minimizing $\mathcal{M}$ for one variable while fixing the other.

The ACS iteration (j) for our problem is defined as:

$$m^{j+1} = \underset{m}{\operatorname{argmin}} \left\{ \frac{1}{2}||d - Gm||_2^2 + \alpha \sum_{i=1}^{nt} TV(m_i) + \gamma \mathcal{M}(m, \hat{v}) \right\} \\ = \underset{m}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \begin{bmatrix} d \\ 0 \end{bmatrix} - \begin{bmatrix} G \\ \gamma \mathcal{M}_m \end{bmatrix} m \right\| + \alpha \sum_{i=1}^{nt} TV(m_i) \right\} \quad (3)$$

$$v^{j+1} = \underset{v}{\operatorname{argmin}} \left\{ \frac{1}{2} \|d_v - \mathcal{M}_v v\|_2^2 + \beta \left[\sum TV(v_{z_i}) + \sum TV(v_{x_i}) \right] \right\}$$

Numerical implementation

The numerical implementation of the ACS workflow is described in the Algorithm 1. The variables and operators are defined in detail as follows:

Algorithm 1 Iterative Time-Lapse Seismic Inversion with Optical Flow

---

Require: d, G, $m^{\{start\}}$
Ensure: m, v
  Initialize $v^{\{0\}}$ with zeros
  for $i \in \{0, 1, \dots, iters\}$ **do**
    if $i = 0$ **then**
      Solve for m using primal dual with TV:
        $m^{\{0\}} = \underset{m}{\operatorname{argmin}} \quad \frac{1}{2}\|Gm - d\|_2^2 + \alpha\|\nabla^+ m\|_{2,1}$ ▷ PrimalDual is initialized with $m^{\{start\}}$
    else
      Solve for m using $\mathcal{M}_m$:
        $m^{\{i\}} = \underset{m}{\operatorname{argmin}} \quad \frac{1}{2} \left\| \begin{bmatrix} d \\ 0 \end{bmatrix} - \begin{bmatrix} G \\ \gamma \mathcal{M}_m \end{bmatrix} m \right\| + \alpha\|\nabla^+ m\|_{2,1}$
    end if
    Solve for v using optical flow equation:
      $v^{\{i\}} = \underset{v}{\operatorname{argmin}} \quad \frac{1}{2}\|M_v v - dv\|_2^2 + \beta\|\nabla^+ v\|_{2,1}$
  end for

Variables The m vector is a stack of the m_t for $t = 0, 1, 2$: $m = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \end{bmatrix}$, in the same way, the vector v is defined for the time 1 and 2 as: $v = \begin{bmatrix} v_{x1} \\ v_{z1} \\ v_{x2} \\ v_{z2} \end{bmatrix}$ where v_{dt} represents the d component of the velocity between times t and $t-1$. The G operator is the post-stack modeling operator defined for a given wavelet. The starting vector for $m^{\{start\}}$ is obtained by gaussian blurring the velocity model in order to simulate the smooth velocity estimation that is usually available (and needed) in real-life scenarios. This initial vector is employed only for the first PrimalDual solution and the forward iterations are implemented using a warm start with the results from previous iterations. The regularization parameters α, β, and γ are defined by trial and error and are discussed in the results section.

Operators Following the recommendations from the authors, the TV regularization is implemented using a forward difference operator (written as ∇^+ in this document), which is defined using the PyLops library. In the same manner, the matrix G is a block diagonal matrix containing the post-stack operator as implemented in the aforementioned library. As for the matrices $\mathcal{M}_m$ and $\mathcal{M}_v$, they are defined by discretizing the optical flow constraint while fixing one of the variables. Recalling from (1), we can solve for m using $\mathcal{M}_m$ as:

$$\mathcal{M}_m m = \left[\begin{bmatrix} -I & I & 0 \\ 0 & -I & I \end{bmatrix} + \begin{bmatrix} A_{m1} & 0 & 0 \\ 0 & A_{m2} & 0 \end{bmatrix}\right] \begin{bmatrix} m_0 \\ m_1 \\ m_2 \end{bmatrix} = 0 \quad \text{Where } A_{mt} = \left[\mathrm{diag}(v_{xt}\nabla_x^+) \ + \ \mathrm{diag}(v_{zt}\nabla_z^+)\right]$$

As for the $\mathcal{M}_v$ operator we solve for v by doing:

$$\mathcal{M}_v v = d_v \longrightarrow \begin{bmatrix} A_{v1} & 0 \\ 0 & A_{v2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} m_1 - m_0 \\ m_2 - m_1 \end{bmatrix} \quad \text{Where} \quad A_{vt} = \left[\mathrm{diag}(\nabla_x^+ m_t) \quad \mathrm{diag}(\nabla_z^+ m_t)\right]$$

All of these operators were implemented using linear operators with PyLops to make use of the sparsity in the matrices involved.

Solvers For the ease of implementation, the Primal-Dual algorithm defined by Chambolle and Pock 2011 was used as the optimization algorithm inside the given iterations. The Primal-Dual algorithm implementation in PyLops solves the following nonlinear minimization problem:

$$\underset{x \in \mathcal{X}}{\mathrm{argmin}} \quad g(Ax) + f(x)$$

Where A is a linear operator, f and g can be any convex functions that have a known proximal operator. For our interest, we define the g function as the $L_{2,1}$ norm measuring the Total Variation as defined by the operator ∇^+ and the desired vector to regularize. The function f is given by the L_2 norm of the data-model misfit.

Results

To test the proposed implementation, a set of three synthetic models resembling a geologically feasible scenario was produced where a body of contrasting impedance moves vertically between different layers (between times 0 and 1) and then shrinks along the vertical axis (between times 1 and 2). Figure 1 shows the generated data set.

To solve for the ACS Algorithm, five major iterations were defined with 50 iterations each for the internal PrimalDual solvers. The values of the regularization parameters, were defined as following: $\alpha = 5$, $\beta = 4$, $\gamma = 0.3$. These values correspond to the weight of the TV regularization applied to the model, the TV regularization applied to the velocity vectors, and the optical flow constraint when solving for m. Ideally we want blocky solutions since the earth is usually treated as a vertical stack of layers and the same can be applied to the velocity field, a large weight will produce solutions with low amplitudes and no contrast so it's needed to experiment with different values. As for the optical flow equation, we have to take in mind that the general formulation constrains the models between consecutive time frames to be overlapping and with very little changes between frames, this of course is highly violated in our approach so we have to give enough weight to this constrain so that it helps when defining the velocity field but it doesn't overpower the solution by providing ghost effects between the frames. We could see that the velocity solution is not very sensitive to this γ parameter, however, the quality of the recovered models will decrease significantly if defined too large.

Executing an inversion process on an M1 MacBook Air takes approximately 15 minutes. However, this duration can be reduced by optimizing the process with a GPU (Graphics Processing Unit) using software libraries such as Cupy. The outcomes of the inverted models are visually represented in Figure 2.

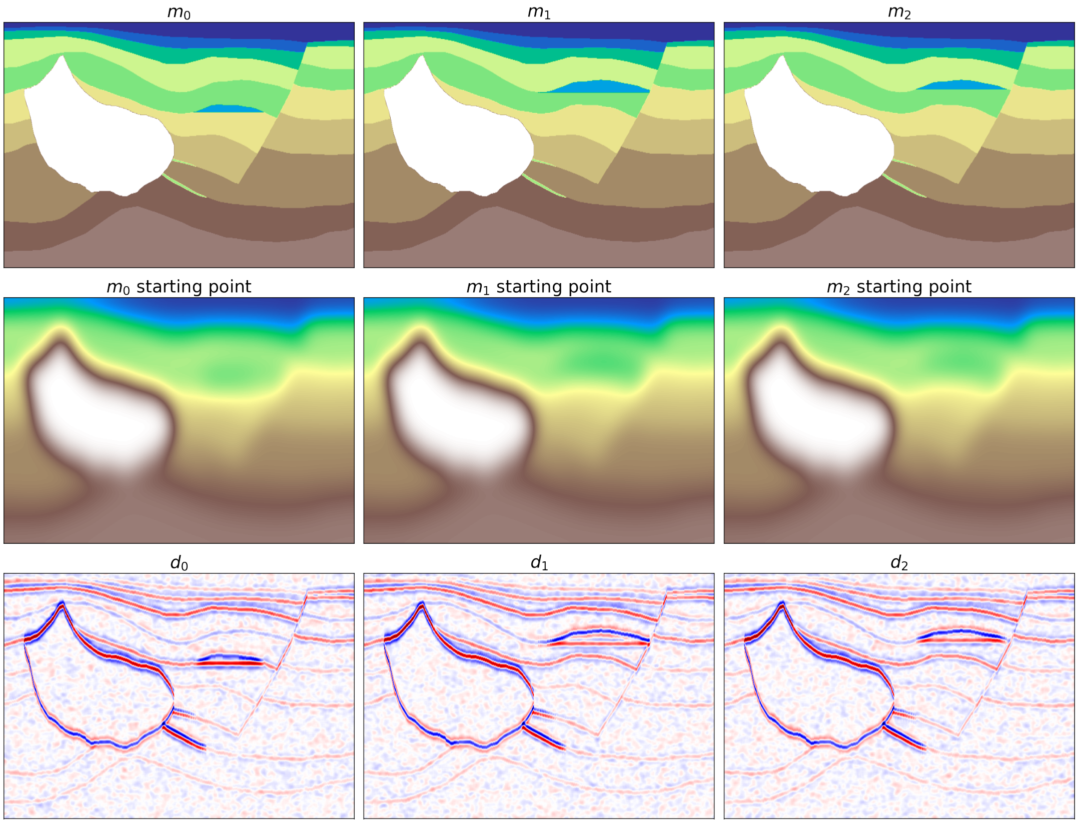

Figure 1: Synthetic dataset used to test the implementation. True models are shown in the first row, the starting point for each model defined by applying a gaussian blurring effect is shown in the second row, and the noisy data used to recover for the models is shown in the final row.

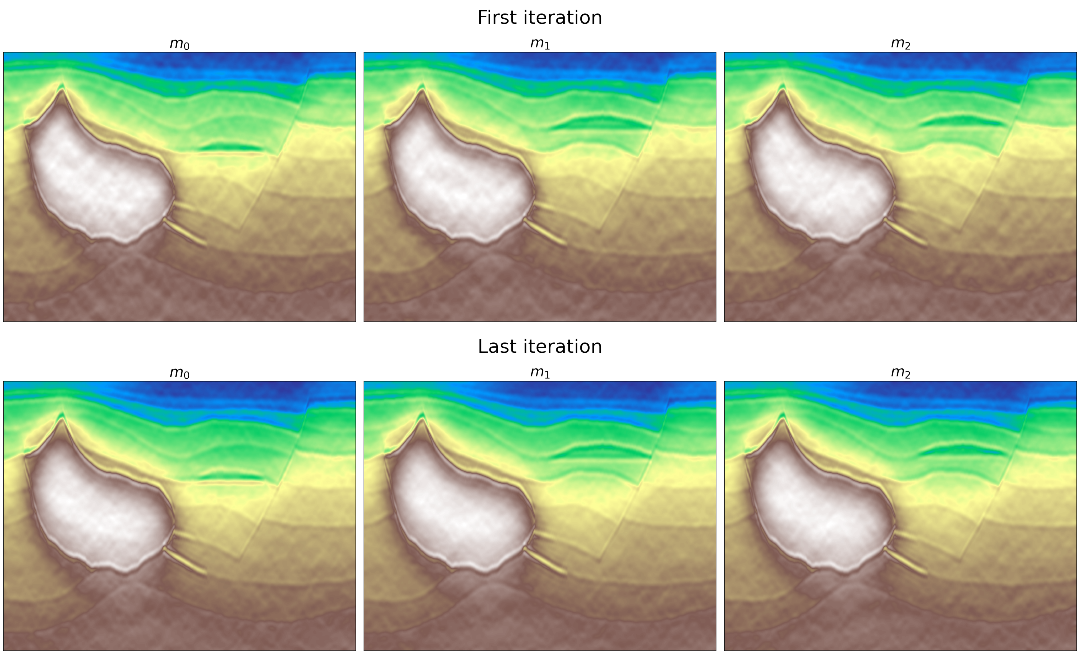

Figure 2: Inversion comparison between first and last iteration

In the initial iteration, the model is solved exclusively with the Total Variation (TV) regularization serving as a constraint. Interestingly, this constraint contributes significantly to establishing an effective starting point for the iterative process. Yet, when juxtaposing the first and the final iterations, we notice that the initial iteration is laden with noise. Moreover, the boundaries demarcating different layers aren't distinctly delineated. We also observe a minor ghosting effect between the displaced bodies, where the outline's shadow is discernible between frames. This phenomenon was previously noted and addressed by reducing the weight assigned to the optical flow constraint. Turning our attention to motion estimation, the results are depicted in Figure 3. Upon analysis, it becomes apparent that the algorithm successfully estimates the motion for the most part by gauging the vertical displacement between the first and second frames. It's intriguing to observe that the algorithm interprets the contraction in the final frame as an upward vertical displacement, thereby exhibiting a substantial vertical displacement magnitude.

However, the data noise and possibly the weight attributed to the various operators seem to cause a blurred representation of the layer boundaries. This issue is addressed by calculating the orientation angle given the velocity components, as demonstrated in the uppermost figure. This process aids in refining the clarity of the image and reducing the distortion caused by the noise and the influence of the multiple operators.

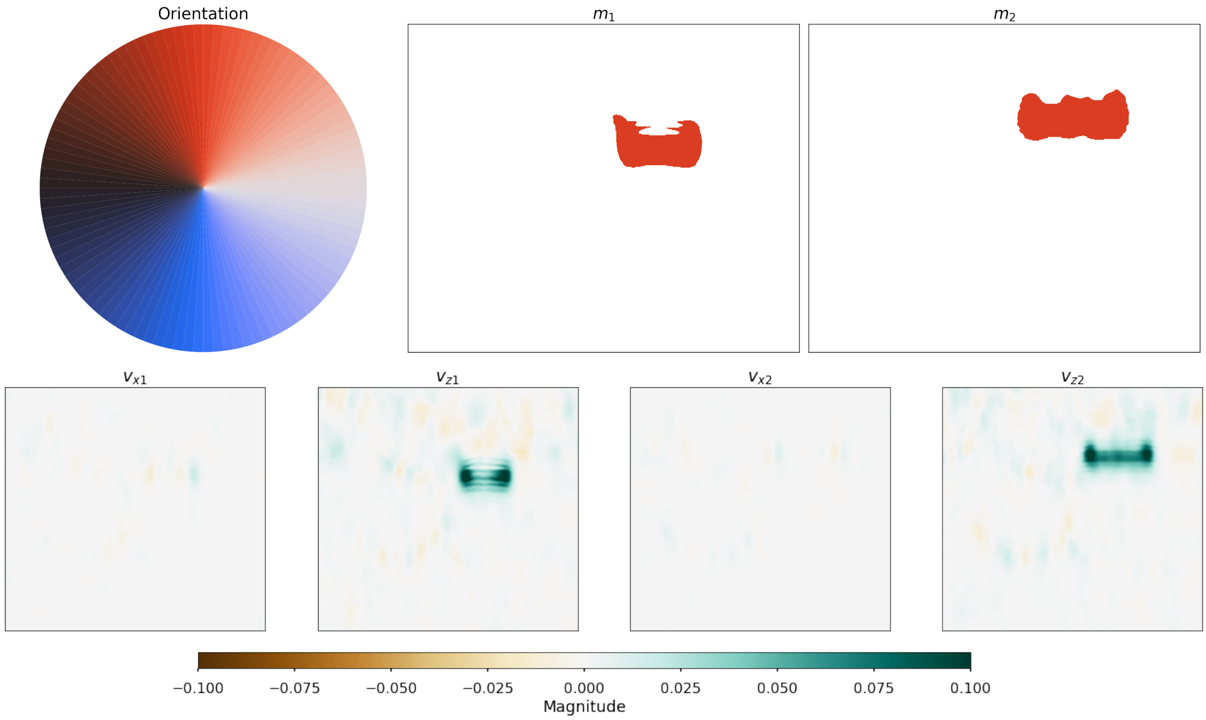

Figure 3: Motion estimation results in orientation (top) and velocity component magnitudes (bottom)

Conclusion

This project has offered a deep dive into the application and nuances of the optical flow as a constrain in a seismic inversion algorithm, particularly in the context of model and velocity field inversions. We discovered the importance of the careful selection of regularization parameters, as they significantly impact the performance of the algorithm. Balancing the Total Variation (TV) regularization applied to the model and velocity vectors is crucial, as high weights can suppress amplitudes and contrast in the solutions.

Additionally, the study emphasized the significance of the optical flow constraint, represented by the γ parameter. A balance must be struck to prevent the creation of ghosting effects between frames, while still aiding in defining the velocity field. As we move forward, continuous experimentation with varying parameters and computational strategies will be critical to further refining the process and outcomes.

Data availability

The code and dataset used to produce these results has been made available in the following repository: https://github.com/dchamorror/SeismicTVFlow.git

References

Burger, Martin et al. (Dec. 1, 2017). “A Variational Reconstruction Method for Undersampled Dynamic X-Ray Tomography Based on Physical Motion Models”. In: *Inverse Problems* 33.12, p. 124008. ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/aa99cf. URL: https://iopscience.iop.org/article/10.1088/1361-6420/aa99cf (visited on 04/30/2023).

Chambolle, Antonin and Thomas Pock (May 2011). “A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging”. In: *Journal of Mathematical Imaging and Vision* 40.1, pp. 120–145. ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-010-0251-1. URL: http://link.springer.com/10.1007/s10851-010-0251-1 (visited on 05/08/2023).

Lucka, Felix et al. (Jan. 2018). “Enhancing Compressed Sensing 4D Photoacoustic Tomography by Simultaneous Motion Estimation”. In: *SIAM Journal on Imaging Sciences* 11.4, pp. 2224–2253. ISSN: 1936-4954. DOI: 10.1137/18M1170066. URL: https://epubs.siam.org/doi/10.1137/18M1170066 (visited on 04/30/2023).