3D Deep-Tomography: 3D velocity model building with deep learning based on subsurface offset gathers

Bernardo M. O. Fraga [1] * , Ana Paula O. Muller [2 1], Jessé C. Costa [3 4] , Clecio R. Bom [1 5], Matheus Klatt [1] , Elisangela L. Faria [1] , Marcelo P. de Albuquerque [1], Marcio P. de Albuquerque [1]
1) Centro Brasileiro de Pesquisas Físicas (CBPF), 2) Petróleo Brasileiro S.A. (PETROBRAS), 3) Universidade Federal do Pará (UFPA), 4) Institute of Petroleum Geophysics (INCT-GP), 5) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ)

This paper was prepared for presentation during the 18th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 16-19 October 2023.

Abstract

Velocity model building is a critical step of seismic processing which recently has attracted great interest in Deep Learning (DL) applications. We propose in this work an extension for 3D models of a previously investigated method called Deep-tomography (DT). DT is a technique consisting of iteratively updating the velocity model by means of a deep convolutional neural network (U-Net), using migrated offset panels as input data. The network is trained to predict an appropriate update to be summed over the migration velocity model. Since the method is based on migrated data as input, which is in the same domain as the desired output, its extension for 3D is more treatable, allowing the division of output and input cubes into smaller units. In this work, we investigated just the last iteration, which is the more challenging one since it predicts the model update with the resolution of the true model. The results for this iteration show that the 3D method can also predict faults and the model interfaces with reasonable accuracy.

Introduction

An accurate and detailed velocity model is responsible for the quality and correctness of the seismic interpretation. The process of obtaining the velocity model is a complex, human-curated, and expensive task involving many steps of data processing, such as Normal Moveout Analysis and Tomography, until obtaining a good initial model for the Full-Waveform Inversion [1].

In recent years, DL algorithms have been intensively investigated in seismic problems [2], attracting particular interest in velocity model-building [3]. The complete prediction of velocity models was initially proposed by using the raw seismic shot data to train the network, which in turn predicted the velocity model that generated those input seismic shots, e.g. [4]. The results were very encouraging; however, the shot-based approaches suffer from generalization issues due to the complexity of the task defined to be solved by the neural network. Another problem with predicting the complete velocity model from seismic shots is the size of real seismic acquisitions that is much larger than what the most modern GPU devices could support. Recently, [5] proposed a shot-based approach for 3D geometry, which tried to conciliate the high memory demand with the size of the 3D models by summing different shots. This solution worked for simple models, despite not reducing the complexity of the data domain transformation involved, and still requires that the shot geometries must be regular and fixed-spread, covering all the surfaces to be investigated. This requirement is made in order to conciliate the irregularity of seismic shots and the geometric non-correspondence of this data with the seismic images or the velocity models, which may not be possible for some acquisition geometries.

One alternative to reduce the data size and the complexity of the task is to migrate the shots [6,7]. The velocity chosen for migration can be a constant velocity [6], with the true model being predicted in one step, or a rough gradient of the true model, which will be iteratively updated until reaching a final solution [7]. The second approach, called Deep-Tomography, is able to generalize to complex geological structures when compared with a one-step model prediction.

The previously mentioned techniques rely on 2D applications, which are impossible to apply for real data. The extension to 3D simulations requires careful DL design to avoid memory issues on the GPUs. This work presents a 3D approach for the Deep-Tomography technique which showed promising results for the synthetic velocity models investigated.

Method

Deep-tomography uses images migrated with RTM (Reverse Time Migration) and a cross-correlation extended imaging condition [8]. For the 3D case, this imaging condition can be written as

$$I(x,\, y,\, z,\, \lambda_x,\, \lambda_y) = \sum_{shots} \sum_{t} \, [W_s(x - \lambda_x, y - \lambda_y, z, t) \; X \; W_r(x + \lambda_x, y + \lambda_y, z, t)] \quad (1)$$

For each investigated subsurface offset (λ_x or λ_y), a different image is generated, with the property that the events migrated with the accurate velocity focalize to zero subsurface offset value. It was previously shown that focalization can be used as a measure to correctly update

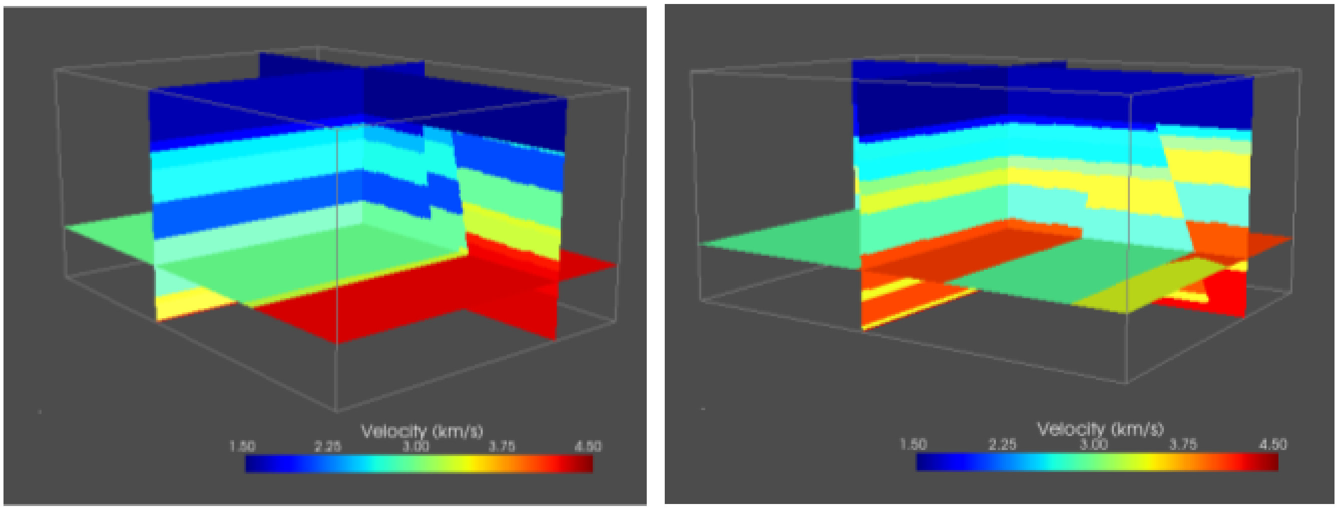

Figure 1: Two examples of the velocity models created for the supervised trainning of Deep-Tomography.

the velocity model [9]. To create the subsurface offset panels, we varied λ_x and λ_y independently, creating fifteen panels to use as input for the neural network, with the minimum value equal to zero, and the maximum value for λ_x=250m and for $\lambda_y = 312.5$m. The maximum subsurface offsets were empirically chosen in order to capture the maximum offset where it was possible to observe coherent events.

Supervised neural networks require a reasonable amount of data to be used in the training process. To have control of the labels, i.e., the expected output of the network, and since it is still a proof of concept work, we created a synthetic data set to evaluate the technique. Our dataset consists of 33 synthetic velocity models with full coverage acquisition region with 2 km of depth, 4 km of extension in the inline direction, and 3 km in the crossline direction. The spatial sampling rate is equal to 12.5 m for the three directions. We linearly extrapolated the original cubes in order to accommodate the acquisition geometry.

Our shot simulator uses a finite-difference wave propagator, isotropic, acoustic, with second-order in time, eight-order in space, and exponential attenuation on the absorbing boundaries. The acquisition geometry defines 4800 shots, with 50 meters of increment in inline and crossline directions. The simulated receivers are ten streamers 2000m long, spread with a maximum lateral distance of 500m from the source.

To make our synthetic models with high structural complexity, we simulated a 3D system of deposition and faulting. The deposition process occurs in packages where a sequence of 5 deposited layers configures a package. The algorithm randomly chooses the thickest of each layer from a range of possible values. After the complete deposition of a package, we faulted the deposited layers using a random number of normal faults with random angles and displacements. Figure 1 presents some examples of velocity models generated.

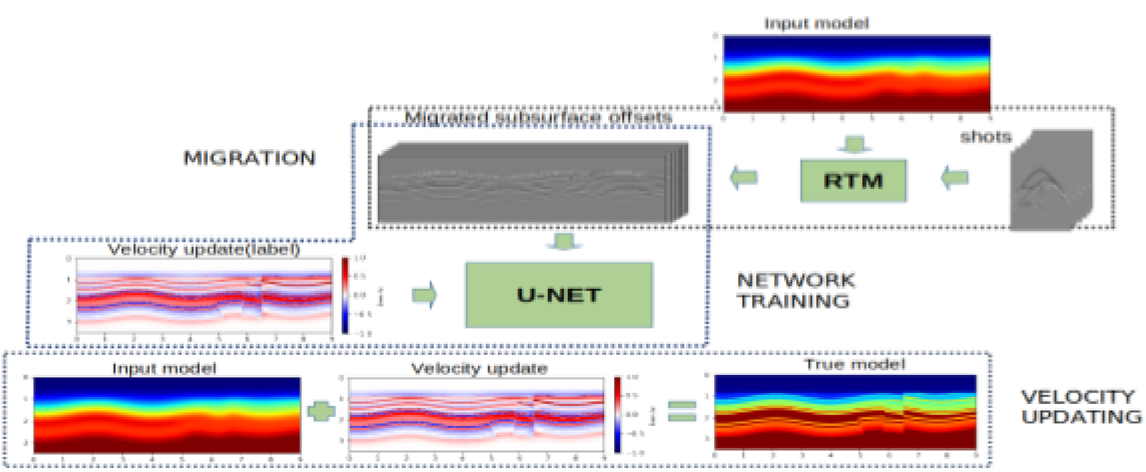

Figure 2: Workflow for the final iteration of Deep Tomography (shown here in the 2D case): the seismic shots are migrated with RTM and a smooth velocity model, and the migrated offsets are given as input to the network. The net is then trained to predict the velocity update.

As proposed by [7], Deep-tomography is an iteratively supervised DL method using the migrated subsurface offset panels as input for a U-Net. The expected output is an update for the velocity model used in migration. The technique relies on using the network to measure the velocity error from the migrated panels and, from this information generate the appropriate velocity model updating. It has the advantage that the input/output pairs present the same dimension. In its 2D implementation, it is necessary to run three iterations to obtain the true velocity from an initial quasi-horizontal gradient. At each iteration, the velocity model accuracy and resolution are improved.

Initially, for the 3D approach, we investigated just the final iteration, which uses a smooth version of the true model as the migration model. This smooth version is generated by using a Gaussian filter with a window size equal to 20. The neural network's supervised training is performed to predict the velocity update to be applied over the migration model to reach the true model which generated the shots, as exemplified by Figure 2.

We used a U-Net [10] as our network architecture of choice; it is composed of a contracting and a symmetric expanding path, defining a u-shape form. Each step in the contracting path consists of a series of convolutions followed by a pooling layer. The number of convolutional filters increases at each step and the spatial size decreases due to pooling. The expanding path does the opposite, using up-convolutions to upsample the image but reducing the number of filters at each step. During the expanding path, information from the contracting phase is concatenated to localize finer features better. In our implementation, we used three contracting steps, starting with 16 filters and doubling them at every step. At each step, we apply two successive blocks of a 3D Convolutional layer, batch normalization, and the Rectified Linear Unit (ReLU) activation.

The U-net architecture defines a flow in which the output image has the same size as the output, except for the number of channels. This feature is extremely convenient to our problem since migrated images and models share the same dimensions except for the number of subsurface offsets, treated as channels of the input images.

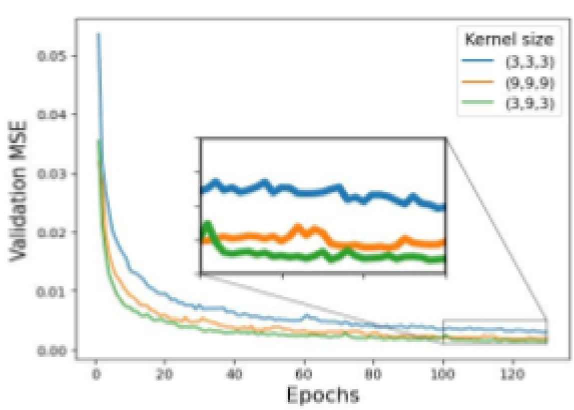

Figure 3: Mean Squared Error for the validation sample for the three different kernel shapes.

In order to train our network, we chose 27 out of the original 33 synthetic models to be our training/validation sets. We subdivided those into smaller sections of 500m in the crossline direction and 1km in the inline direction, with steps of 250 and 500m in each direction, covering the entire depth. This gives a total of 77 subcubes per model. During the training and validation phase, the subcubes were treated as independent. However, since the borders of the subcubes suffer from the lack of information from the neighborhood, we chose the division with superposition to combine the prediction results using a Hann windowing [11] to balance the results before summing the inline/crossline directions.

The depth dimension was not divided to define the subcubes because the dependency of the velocity errors in the depth direction is stronger than in the other directions. An error in the top of the velocity model propagates to the deepest portion of the model. With the proposed division, we guarantee that the network can correlate the top regions of the migrated input images to the deeper ones and deal with errors that could propagate from shallow subsurfaces to deeper ones. A sensibility study would be necessary to verify if the lateral division is reasonable to account for the lateral spreading of the velocity errors.

Results

The network was trained for 130 epochs, using the Adam optimizer [12] with a learning rate of 0.0001, and a Mean Squared Error (MSE) as the loss function. As was previously discussed in the Deep-tomography 2D implementation [7], the convolutional kernel size is an important parameter to guarantee the accuracy of the predictions, due to the long-range effects of the velocity errors over the migrated images. We tested three convolutional kernel shapes: two symmetric ones with all filter sizes equal to 3 or 9 and an asymmetric one with kernel sizes equal to (3, 9, 3), with the largest dimension corresponding to depth. The resulting MSE for the validation sample at each epoch for the three different kernel shapes is shown in Figure 3. As was previously mentioned, it is expected the depth dependency on the velocity errors to have a long-range reach, and better results using the largest kernel size only in the depth dimension are in agreement with this observation.

We chose the weights when the validation loss was lowest as the best-trained U-Net to make the predictions for each sample in our test set. We divided the 3D models in the same way as in the training phase, this time keeping track of the origin of the subsets to reassemble the velocity models. When patching together the predictions, as previously mentioned, we used a Hann window in the crossline and inline directions in the areas of overlap between subsets.

The results for two of the synthetic velocity models can be seen in Figure 4. We plotted for each result the migration model, the predicted model from DT, and the true model.

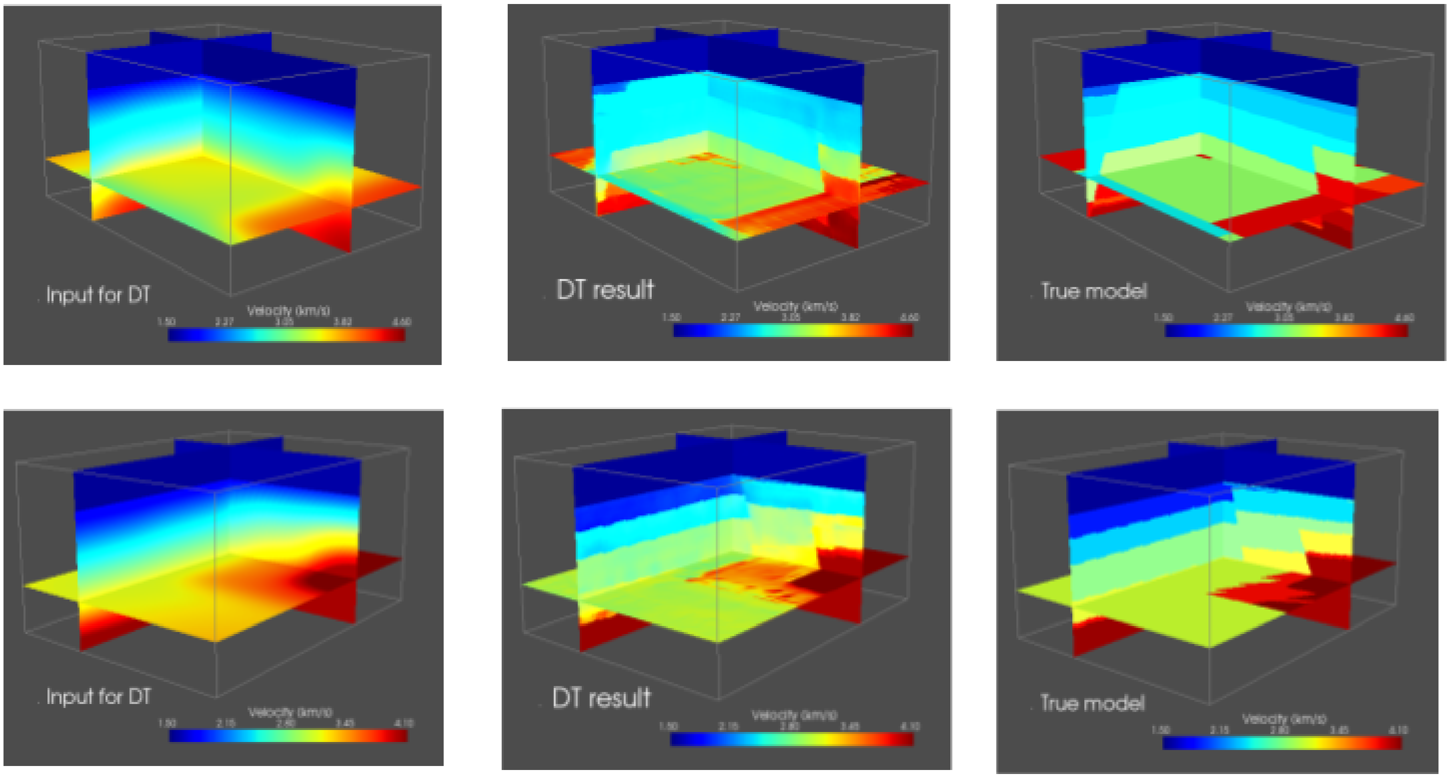

Figure 4:Results for two of the synthetic velocity models in the test sample. On the left, we plotted the velocity model used to migrate the images used as input to the neural network; on the middle, the prediction by the DT flow; on the right, the true model.

As previously mentioned, the DT result is obtained by summing the U-Net output over the migration model.

Figure 4 shows predictions with a reasonable agreement with the true models. The network predictions were able to recover the layer's contrasts present in the true model, also recovering the lateral variations defined by the faults.

Conclusions

This work presented the initial studies about the extension of Deep-Tomography to 3D data. Our work presents the advantages of a DL image-based approach over a shot-based one, by the flexibility of splitting the input data in the most suitable and memory-saving configuration, due to the close correspondence between the input and output data.

The initial results for the final interaction, which leads the model to the true model presented reasonable accuracy with the expected result. Probably better results could be achieved using a larger training data set. Further studies are required to investigate the technique as a potential approach for real data, as the application of the complete deep-tomography flow, which iteratively updates a vertical gradient model, with no structures to the true model. It is also important to investigate alternative network architectures and loss functions. Besides, it is necessary to increase the structural complexity of the generated models by, for example, including folding and erosion, and also the size of the migrated cubes.

One important advantage of DL methods for the velocity model flow is the potential to speed up the results since once the neural network is trained, the results are obtained almost instantaneously. Deep-tomography approach requires the migration of the data, but after migration, the model updating follows an automatic and fast flow.

Acknowledgments

APOM thanks Petrobras for sponsoring her postdoctoral research and for the permission to publish this work. JCC acknowledges the CNPq financial support through the INCT-GP and the grant 312078/2018-8 and Petrobras. CRB acknowledges the financial support from CNPq (316072/2021-4) and FAPERJ (grants 201.456/2022 and 210.330/2022). Finally, the authors acknowledge the LITCOMP/COTEC/CBPF multi-GPU development team for supporting the Artificial Intelligence infrastructure and Sci-Mind's High-Performance multi-GPU system, and to SENAI CIMATEC Supercomputing Center for Industrial Innovation, for the cooperation, supply, and operation of computing facilities.

References

1. Yilmaz, Ö., and Doherty, S. M. Seismic Data Analysis: Processing, Inversion and Interpretation of Seismic Data. SEG Books, 2001.
2. Mousavi, S. M., Beroza, G. C., 2022, Deep-learning seismology. Science, 377, eabm4470.
3. AlAli, A. & Anifowose, F., 2022. Seismic velocity modeling in the digital transformation era: a review of the

role of machine learning, Journal of Petroleum Exploration and Production Technology, 12, 21-34.
4. Araya-Polo, M., S. Farris, and M. Florez, 2019, Deep learning-driven velocity model building workflow. The Leading Edge, 38, 872a1–872a9.
5. Gelboim, M., Adler, A., Sun, Y., Araya-Polo, M., 2023. Encoder−Decoder Architecture for 3D Seismic Inversion. *Sensors*, *23*, 61.
6. Geng, Z., Z. Zhao, Y. Shi, X. Wu, S. Fomel, and M. Sen, 2022, Deep learning for velocity model building with common-image gather volumes. Geophysical Journal International, 228, 1054–1070.
7. Muller ,A.P.O., Bom, C.R., Costa, J.C., Klatt, M., Faria, E.L., Silva, B.S., Albuquerque,M. P., 2022. Deep-tomography: iterative velocity model building with deep learning, Geophysical Journal International, ggac374.
8. Sava, P., and I. Vasconcelos, 2011, Extended imaging conditions for wave-equation migration. Geophysical Prospecting, 59, 35–55.
9. Frigério, O., J. and Biondi, B., 2021. Tomographic waveform inversion (TWI), in IMAGE - International Meeting for Applied Geoscience & Energy, SEG|AAPG.
10. Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, 234–241.
11. Lizhe Tan, Jean Jiang, in Digital Signal Processing (Third Edition), 2019
12. Kingma, D. P. and Ba, J. Adam: a method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015.