Style transfer for generation of realistically textured subsurface models

Oleg Ovcharenko, Vladimir Kazei, Daniel Peter and Tariq Alkhalifah*

SUMMARY

Training datasets consisting of numerous pairs of subsurface models and target variables are essential for building machine learning solutions for geophysical applications. We apply an iterative style transfer approach from image processing to produce realistically textured subsurface models based on synthetic prior models. The key idea of style transfer is that content and texture representations within a convolutional neural network are, to some extent, separable. Thus, a style from one image can be transferred to match the content from another image. We demonstrate examples where realistically random models are stylized to mimic texture patterns from Marmousi II and a section from the BP 2004 benchmark velocity models.

INTRODUCTION

Deep learning models require thousands of samples to infer dependencies in the data. With the increased interest in deep learning in the field of geophysics, numerous data samples become essential for training neural network models. Especially in exploration surveys, significant amounts of seismic data have been acquired over past decades. This allowed for successful data-domain machine learning applications, such as first break picking (Yuan et al., 2018), data interpolation (Jia and Ma, 2017) and denoising (Jin et al., 2018) among others. Image-domain geophysical machine learning applications, however, require numerous sets of realistic subsurface models (Kazei et al., 2019) for training.

The task of velocity model building, e.g., from seismic data (Wang et al., 2018; Mosser et al., 2018), low-frequency data extrapolation from shot gathers (Ovcharenko et al., 2019) and salt body delineation (Shi et al., 2018), require subsurface models with specific properties. Neural networks used for these applications are usually trained in a supervised manner on hundreds of data-model pairs. Getting realistic velocity profiles from field data is a costly and non-trivial problem, and usually affordable for quality control and testing purposes only. A common workaround is the generation of velocity models which can be used to produce synthetic seismic data or any other derivatives.

There are numerous empirical approaches for the generation of custom subsurface models. Random model generators are usually tailored for a specific task and might produce generic types of random subsurface models, e.g., layered or salt induced models. However, a simple generator usually fails to deliver a model of high perceptual quality. Meaning that the resulting subsurface models do not look realistic from an expert point of view (i.e. layering, faults, trends, and so on). For this reason, we investigate here a style transfer approach, which recently has gained attention in the field of image processing.

Perceptual realism of synthetically generated images is one of the desirable outcomes in the computer vision community. Modern image processing techniques are often based on a class of neural networks called Convolutional Neural Network (CNN). A generic feed-forward CNN consists of a convolutional encoder and a target-oriented part which, e.g., is made of a number of fully-connected layers. The encoder maps an image into a descriptive latent space by hierarchically decomposing the image into a set of features learned by a bank of filters in each layer of the network. In shallow layers of the convolutional encoder, filters learn low-level features such as simple color contrasts, however filters in deeper layers retrieve more complex patterns. A beneficial feature of CNNs is that the same pre-trained encoder might be used in different applications. Once trained, the same set of filters might be used to search for matching patterns in a different dataset.

Recently, Gatys et al. (2015) demonstrated that representations of content and texture (or style) in a CNN are separable. Meaning that any image can be decomposed to some extent into its content and texture representations, which can then be mixed to produce a new output with a persistent high perceptual quality. The content representation aims to describe objects and structures which are present in the image, whereas the style representation derives and reproduces textures and color patterns in the image. In a similar way, popular public services can stylize any photographs by mimicking famous artworks. For our purposes, we amend this proposed style transfer concept from image processing to enhance the geological realism of generated subsurface models.

METHOD

According to Gatys et al. (2015), style transfer is an iterative optimization task, which aims to generate an image $\mathbf{x}$ that minimizes the discrepancy between the content representation of one image $\mathbf{a}$ and the texture representation of another image $\mathbf{b}$. The key idea of this approach is that representations of content and textures in the CNN are separable and convolutional layers of a pre-trained CNN might be used as descriptors of content and textures in the input image. Each filter in a convolutional layer of the network produces a feature map which maximizes the areas where the filter matches a pattern in the image (Yosinski et al., 2015). Thereby, a set of filters parametrizes the original image and allows similarity between images to be measured as a distance between corresponding feature maps. This difference might be formulated as a loss function, which is a sum of respective content and texture losses.

In the following, we provide a brief summary of the main ideas of Gatys et al. (2015) and Ulyanov et al. (2016), who extended the original approach by designing a neural network for fast style transfer.

Content loss
The content loss describes the spatial arrangement of objects in the image. Content loss is proportional to the root-mean-square distance between feature activation maps in selected layers of the CNN. Convolution of the i-th filter in the layer l with the input image results in a feature map $F_i^l(\mathbf{x})$. By matching activations from selected convolutional layers L_C, the optimization converges to an image $\mathbf{x}$, which mimics the content from the prior image $\mathbf{a}$. Content loss J_C is thus defined as

$$J_C(\mathbf{a},\mathbf{x}) = \sum_{l \in L_C} \sum_{i=1}^{N_l} ||F_i^l(\mathbf{a}) - F_i^l(\mathbf{x})||_2^2 \, , \qquad (1)$$

where N_l is number of filters in the l-th layer of the network. Assuming that the network is convolutional, feature maps $F_i^l(\mathbf{x})$ are matrices such as the ones shown in Figure 1, where spatial indexes in notation are omited for brevity.

Texture loss
The texture loss accounts for joint activations in feature maps in selected layers of the CNN. In other words, similarly textured images will produce a similar cumulative response for the same set of filters. The feature correlations are quantified by a Gram matrix G defined as

$$G_{ij}^l(\mathbf{x}) = \langle F_i^l(\mathbf{x}), F_j^l(\mathbf{x}) \rangle \, . \qquad (2)$$

which consists of inner products between the vectorized feature maps F_i^l and F_j^l in the layer l. The Gram matrix quantifies correlated activations of filters and has $N_l \times N_l$ members, where N_l stands for the number of filters in layer l. Texture loss J_T can be written as

$$J_T(\mathbf{b},\mathbf{x}) = \sum_{l \in L_T} ||G^l(\mathbf{b}) - G^l(\mathbf{x})||_2^2, \qquad (3)$$

which, unlike content loss J_C, is insensitive to spatial locations of textures within the image. Each inner product of two feature maps in the layer results in a scalar value which does not carry information about spatial context in the image, but is focused on the kind of features.

Total loss
The iterative search procedure for the image $\mathbf{x}$, combining content and textures from images $\mathbf{a}$ and $\mathbf{b}$ respectively, starts from a white noise distribution of the same size as the target image. The optimization algorithm attempts to jointly minimize the weighted content and texture loss terms,

$$J(\mathbf{a},\mathbf{b},\mathbf{x}) = \alpha \, J_C(\mathbf{a},\mathbf{x}) + \beta \, J_T(\mathbf{b},\mathbf{x}) \, . \qquad (4)$$

The parameter α controls the contribution of content representation into the resulting image, whereas β similarly contributes to the style representation. Only one of these two parameters is actually needed as the total magnitude of the objective function matters. However, we keep the redundant notation with both α and β as it leads to a more clear formulation of the loss function. The larger the ratio α/β becomes, the more sensitive the algorithm behaves to shapes found in feature maps from the layer l of the pre-trained CNN. A total variation regularization term (Mahendran and Vedaldi, 2015) added to the total loss makes the resulting image more consistent by imposing penalty on variation between neighboring pixels. Detailed explanation of derivatives for content and texture loss terms with respect to the target image $\mathbf{x}$ and activations in layers of the network F_i^l can be found in Gatys et al. (2015).

In the following, we will investigate this style transfer approach and effects of different α/β ratios for exploration model setups.

EXAMPLES

To get feature maps for the input image, we follow the original work of Gatys et al. (2015) and use the pre-trained VGG16 network (Simonyan and Zisserman, 2014). The VGG16 is a convolutional neural network created to challenge human performance on general object classification tasks.

The architecture of the encoder part of the network is made of 5 structural blocks, each consisting of two or three convolutional layers and one max pooling layer for dimensionality reduction. In total, the encoder architecture includes 13 convolutional and 4 max pooling layers. We extract meaningful content representations from layer `conv4_2` $\in L_C$ and compute Gram matrices for texture representations for feature maps from layers `conv{1-5}_1` $\in L_T$. Notation `conv4_2` refers to the second convolutional layer in the fourth block of the network architecture. Figure 1 shows examples of feature maps from layer `conv4_2` obtained by convolution of the Marmousi II benchmark model (Martin et al., 2006) with respective filters.

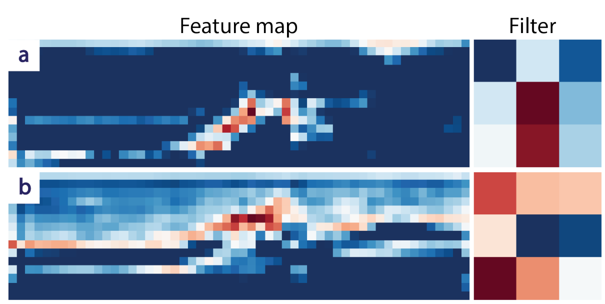

Figure 1: Feature maps for two corresponding filters (a) and (b) in the `conv4_2` layer of the VGG16 network, applied to the Marmousi II benchmark velocity model.

As an optimization algorithm, we employ the L-BFGS scheme to minimize the total loss function J. Additionally, we use a mild total-variation regularization to enforce smoothness in the resulting images. For all examples provided in this section, we run 50 iterations of the L-BFGS algorithm.

Workflow explained
We demonstrate the iterative style transfer workflow by texturizing a content template image $\mathbf{a}$ to exhibit features extracted from a texture sample image $\mathbf{b}$, which is a schematic representation of geology in the Meander belt outcrop Deschamps et al. (2012) . The content image, Figure 2(a), has dimensions

of 175×600 pixels and is assembled of two concatenated linear profiles with sparse stepping, inverted one with respect to another. The sample of geological texture, Figure 2(b), mimics only a realistic spatial distribution of fine layers. It has been resized to have the same size as the content image.

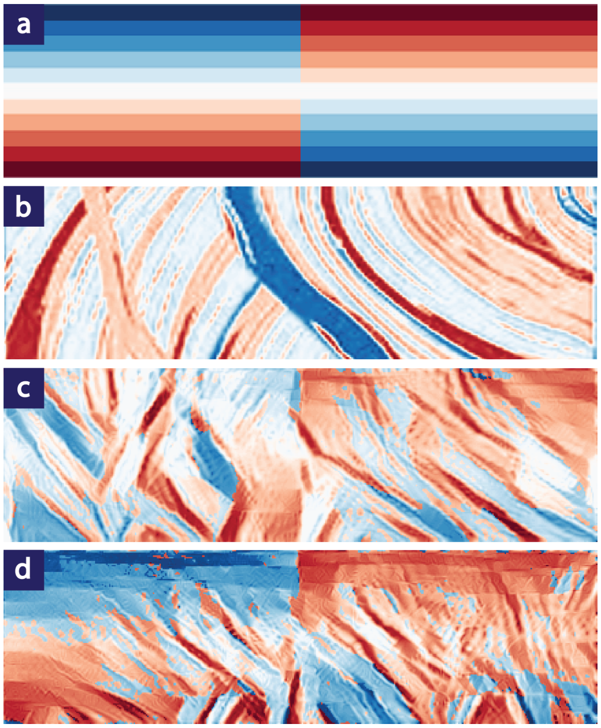

Figure 2: Style transfer example for content model (a) and texture sample (b) from the Meander belt outcrop. Results are shown for content to texture ratio, α/β, of (c) 0.25 and (d) 2.0.

First, the content image **a**, the texture image **b** and the white noise image **x** independently proceed through the layers of the pre-trained VGG16 network, where they are decomposed into a number of feature representations. Then, the content loss is computed between **a** and **x** by substituting 512 feature maps of size 15×54 from layer `conv4_2` into eq. 1. Feature maps from a different set of convolutional layers, L_T, are then used to build Gram matrices and to compute the texture loss between **b** and **x** according to eq. 3. The last step is to compute the total loss J according to eq. 4, and its gradient with respect to all pixels of the image **x**.

The resulting textured models for two different α/β ratios are shown in Figure 2(c-d). Texture contributions in Figure 2(c) dominate over the content, which leads to a perceptually uniform texturing of the image where horizontal layers from the original image are barely distinguishable. The layered structure is showing up when the contribution of the content loss increases, Figure 2(d). Despite completely artificial structure of the content image, the optimization algorithm attempts to create an output which mimics geological features.

Random subsurface models

In this section, we show examples of style transfer on a number of synthetic subsurface models and use the Marmousi II and a section from the BP 2004 benchmark model as donors of geological features. The Marmousi model is dominated by layered structures, whereas the section from the BP 2004 model has a smooth background and a contrasting salt body in the middle.

We create a set of prior content images by using custom random model generators, which were applied to create training datasets for shot-to-shot low-frequency data extrapolation by a deep CNN (Ovcharenko et al., 2019). The left column in Figure 3 lists a number of models which were created using: (1) a random Gaussian field, (2) an assembly of vertical 1D velocity profiles, (3) manipulations with wavelets (Kazei et al., 2019), (4-5) a linear gradient and with an embedded reflective body, respectively, (6) the BP 2004 model, (7) white noise and (8) a homogeneous background.

In general, we observe a few dominant textures in each of style images, which dominate the generated images. For the Marmousi, these are parallel layers, and for the section from the BP 2004 model, these are smooth gradients interrupted by sharp contrast inclusions. Being initiated from the white noise, the algorithm attempts to merge the content prior with the texture sample preserving, however, perceptual consistency in the resulting image.

White noise as content prior leads to a scattered image, challenging the optimization algorithm. The BP 2004 model used both as content and style priors, as expected, converges to itself. The gradient and homogeneous content priors produced the geologically most realistic subsurface models.

DISCUSSION

The described iterative optimization approach for style transfer takes tenths of seconds on a modern GPU to complete the texturing for a single image. Moreover, computational costs linearly grow with the size of the input image. Ulyanov et al. (2016) and Johnson et al. (2016) proposed neural network architectures which might be trained to complete a specific texture transfer for an image of arbitrary size in a fraction of a second. Such a fast formulation enables on-the-fly generation of random subsurface models with shared texture patterns. However, seismic applications rarely require real-time generation of training datasets. Thus, in this work we employed a simple iterative implementation, with weights adjustable by the user.

Another note is about dimensionality of the input and target data. Widely used descriptor CNNs (Canziani et al., 2016) are usually trained for real-world image classification. Meaning that the input data to the CNN is an image with three color channels. This might be a built-in benefit for transferring texture patterns in elastic media parametrized by V_p, V_s and ρ. However, for acoustic media, parametrized by the V_p only, the generated three color channel image has to be properly mapped into a monochromatic representation.

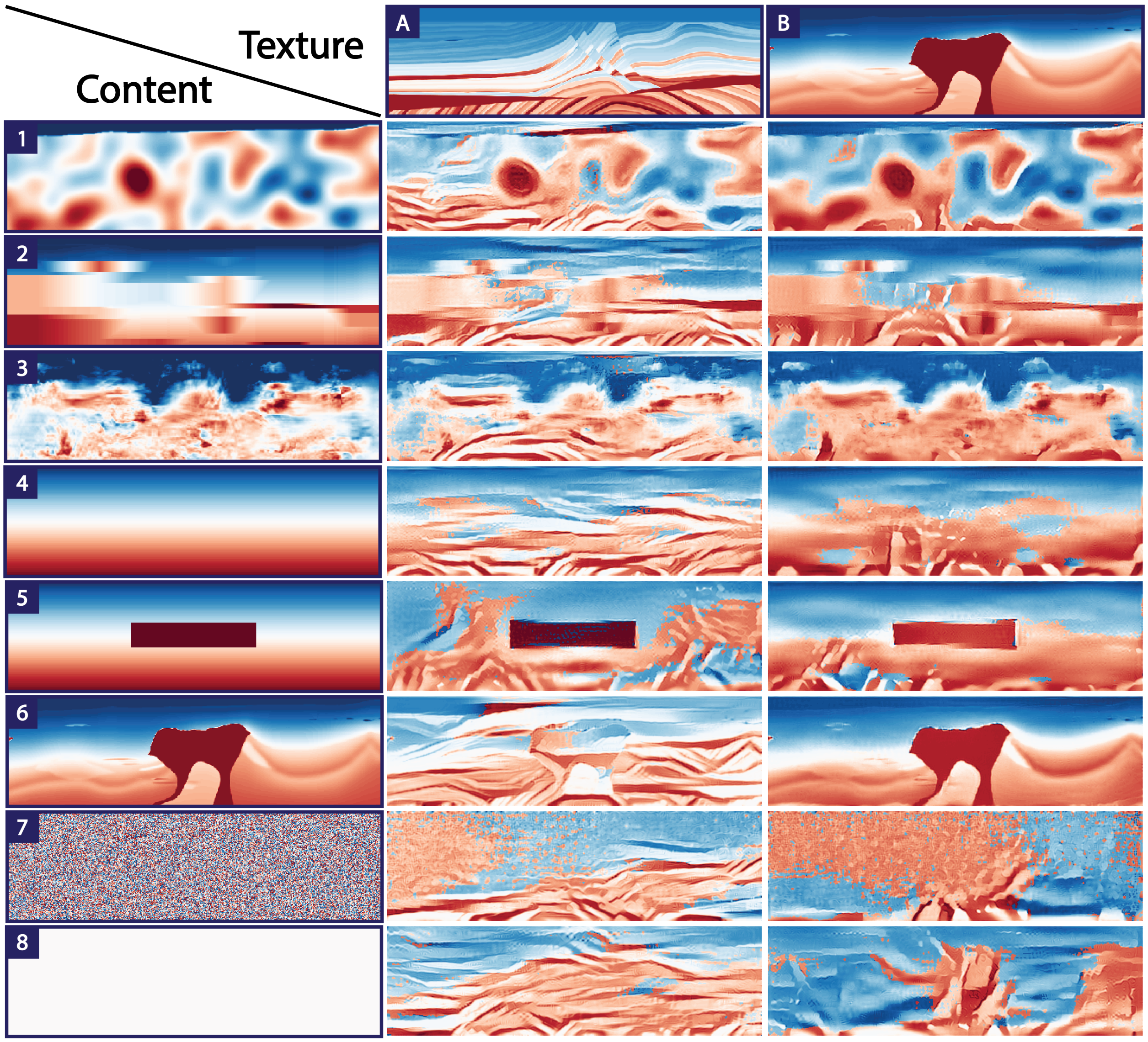

Figure 3: Synthetic velocity models textured to mimic features from (A) Marmousi II and (B) a central section from the BP 2004 benchmark models for even contribution of content and texture loss terms, $\alpha/\beta = 1$. Synthetic content models (1-8) were created using (1) a random Gaussian field, (2) random vertical profiles, (3) wavelet permutations, (4-5) linear gradient and contrast body, (6) BP 2004 model, (7) white noise and (8) a homogeneous model.

CONCLUSIONS

We demonstrated a style transfer approach from image processing to enhance the perceptual realism of generated subsurface velocity models. In our application, the content prior of a subsurface model can be stylized to exhibit complex textures from a sample geology prior. The produced mixed models do not alter significantly from their content priors, however, they are mimicking features from the respective texture priors. We plan to incorporate this style transfer approach into generating the training datasets of realistic subsurface models to improve machine learning solutions for seismic inverse problems.

ACKNOWLEDGMENTS

We thank Kevin Zakka for his implementation of the Gatys et al. (2015) algorithm (https://github.com/kevinzakka/style-transfer). The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia.

REFERENCES

Canziani, A., A. Paszke, and E. Culurciello, 2016, An analysis of deep neural network models for practical applications: arXiv preprint arXiv:1605.07678.

Deschamps, R., N. Guy, C. Preux, and O. Lerat, 2012, Analysis of heavy oil recovery by thermal EOR in a meander belt: from geological to reservoir modeling: Oil & Gas Science and Technology–Revue dIFP Energies nouvelles, **67**, 999–1018.

Gatys, L. A., A. S. Ecker, and M. Bethge, 2015, A neural algorithm of artistic style: arXiv preprint arXiv:1508.06576.

Jia, Y., and J. Ma, 2017, What can machine learning do for seismic data processing? an interpolation application: Geophysics, **82**, V163–V177.

Jin, Y., X. Wu, J. Chen, Z. Han, and W. Hu, 2018, Seismic data denoising by deep-residual networks, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 4593–4597.

Johnson, J., A. Alahi, and L. Fei-Fei, 2016, Perceptual losses for real-time style transfer and super-resolution: European conference on computer vision, Springer, 694–711.

Kazei, V., O. Ovcharenko, T. Alkhalifah, and F. J. Simons, 2019, Realistically textured random velocity models for deep learning applications: Presented at the 81th EAGE Conference and Exhibition 2019.

Mahendran, A., and A. Vedaldi, 2015, Understanding deep image representations by inverting them: Proceedings of the IEEE conference on computer vision and pattern recognition, 5188–5196.

Martin, G. S., R. Wiley, and K. J. Marfurt, 2006, Marmousi2: An elastic upgrade for Marmousi: The Leading Edge, **25**, 156–166.

Mosser, L., W. Kimman, J. Dramsch, S. Purves, A. De la Fuente Briceño, and G. Ganssle, 2018, Rapid seismic domain transfer: Seismic velocity inversion and modeling using deep generative neural networks: Presented at the 80th EAGE Conference and Exhibition 2018.

Ovcharenko, O., V. Kazei, M. Kalita, D. Peter, and T. Alkhalifah, 2019, Deep learning for low-frequency extrapolation from multi-offset seismic data : Geophysics, *submitted*.

Shi, Y., X. Wu, and S. Fomel, 2018, Automatic salt-body classification using a deep convolutional neural network, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 1971–1975.

Simonyan, K., and A. Zisserman, 2014, Very deep convolutional networks for large-scale image recognition: arXiv preprint arXiv:1409.1556.

Ulyanov, D., V. Lebedev, A. Vedaldi, and V. S. Lempitsky, 2016, Texture Networks: Feed-forward Synthesis of Textures and Stylized Images.: ICML, 4.

Wang, W., F. Yang, and J. Ma, 2018, Velocity model building with a modified fully convolutional network, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 2086–2090.

Yosinski, J., J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, 2015, Understanding neural networks through deep visualization: arXiv preprint arXiv:1506.06579.

Yuan, S., J. Liu, S. Wang, T. Wang, and P. Shi, 2018, Seismic waveform classification and first-break picking using convolution neural networks: IEEE Geoscience and Remote Sensing Letters, **15**, 272–276.