Using input-adaptive dictionaries trained by the method of optimal directions to estimate the permeability model of a reservoir

Mohammad Hosseini a,b, Mohammad Ali Riahi a,*

a Institute of Geophysics, University of Tehran, I.R., Iran
b Geophysics branch of the Geology Department, NISOC, I.R., Iran

A B S T R A C T

The thirst for perfect characterization of petroleum reservoirs is yet to be quenched. Sparse transform is an image compression method the full potential of which is still not employed by the geoscientists in the reservoir characterization workflow. Sparse transform is the state of representing an image model based on a weighted linear combination of a sparse set of columns from a matrix, called dictionary. The required dictionary in the sparse transform scheme is obtained by learning over a training set of model images using dictionary learning algorithms. Meanwhile, the sparse coding algorithms are used to select the sparse proper subset of columns from the trained dictionary and specify appropriate weight to each column in the linear combination. To enhance the results, the property of internal structure and generality for the implicit dictionaries are integrated with the property of adaptability for the explicit dictionaries to regularize and enhance the effectiveness of the input-adaptive dictionary in reconstructing the approximate model.

This phenomenon can be employed in the geosciences to estimate the reservoir model if the dictionary is trained over a set of geologic models of a delighted property. The geologic models should be diverse enough to represent the possible variety of the desired property and the limited number of columns from the dictionary should be properly selected such that it adequately captures the variety inherent within the geologic models. In this study, the variety of the geologic models are provided by manipulating two training images by FilterSim multiple-point geostatistical method and generating a large set of realizations. The training images are extracted from a spectral decomposition profile to represent a stationary fracture system and a nonstationary delta system model in an Iranian petroleum reservoir. The dictionary learning algorithm is an alternation of dictionary updating and sparse coding steps. In this paper, the method of optimal directions is used as dictionary learning algorithm and the least angle regression with shrinkage algorithm is used as the sparse coding method.

The resultant approximate model achieved by the sparse transform compression scheme is comparable to the reference models both in terms of physical structures and fluid flow characteristics. For the current study, the approximate fracture system model is shown to be superior to 86.16% of the model images in the corresponding population space in terms of fluid flow characteristics. In the case of the delta system, the approximate model is superior to 99.17% of the model images in the population space. Further experiments indicate similar deductions using other methods for dictionary learning, sparse coding, and multiple-point geostatistical modeling.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Reservoir property inversion and modeling from seismic data is in fact a non-unique process. When there is no constraint on a delighted property at specific locations, data can vary corresponding to any possible trend. Under the maximum likelihood constraint, the inverted property model from the seismic data will be the most probable model and it is unique (Constable, 1988). Geostatistical inversion methods are designed to provide a limited set of possible inverted models of the desired property from the seismic data. Furthermore, multiple-point geostatistical (MPS) methods are developed to provide the mathematical tools for manipulating a single reservoir property model, which is called ‘training image’, and generate equally probable reservoir models with similar statistical properties.

Relying on any method to build a reservoir model, it is beneficial to generate a set of realizations by MPS methods; the realizations are actually the manipulations of the training image. This is specifically advantageous when the task is to select the most suitable reservoir property model by constraining the reservoir models with fluid flow data. The
training image is in fact a location-independent model which is considered as the source for information and statistical changes in MPS methods (Haldorsen and Damsleth, 1990; Mariethoz and Caers, 2015).

1.1. MPS simulation

Pattern-based multiple-point geostatistical methods are designed based on scanning a location-independent training image using a predefined template and extracting the patterns inherent within the training image. The extracted patterns are classified and to each class a prototype is specified. For each data event the closest class is found and a pattern is randomly selected from that class and pasted on the data event. The realizations generated from the MPS methods are stochastic. FilterSim (Zhang, 2006) is one of pattern-based MPS methods that works based on pre-designed filters. The filters are designed to extract different characteristics of the data event and the patterns. The extracted characteristics are in turn used to enhance classification and determine the closest pattern to the data event.

1.2. Sparse approximation setting

A set of realizations generated by the FilterSim method can be used as a training set for sparse approximation image compression scheme to obtain a compressed approximate to the reference model. In this paper, this task is performed by training a dictionary (that is a matrix) over the training set and selecting a proper number of columns (which are called atoms) from the trained dictionary along with specifying proper weight to each atom. The weighted linear combination of the selected atoms approximates the reference model. The task of selecting a parsimonious set of columns from the dictionary and specifying proper weights to each column is called sparse coding.

1.3. Explicit and implicit dictionaries

The dictionary which is used for sparse approximation is of explicit type. The elements of explicit dictionaries are stated explicit numbers which are obtained by training the dictionary over a set of specific images, i.e. the training set. Accordingly, the explicit dictionaries are content-specific and they perform better for approximating the images that they are adapted for; therefore, such dictionaries lose generality and they do not show specific internal structure. In contrary, the implicit dictionaries are out of the shelves dictionaries which do not need training and their elements are implicit analytical statements. The implicit dictionaries are better fitted for general purposes of image approximation and compression but they are internally structured.

1.4. Input-adaptive dictionaries

The property of adaptability from the explicit dictionary can be combined with the property of generality from the implicit dictionary in order to yield an input-adaptive dictionary. Such input-adaptive dictionary preserves the property of generality while acting effectively, in terms of compression and approximation, on images of the type that they are trained for.

1.5. Dictionary training

The process of dictionary training, as an important step in the sparse approximation scheme, is a sequential alternation between dictionary updating and sparse coding. The sequence starts with an initial dictionary and at each sequence, the dictionary is updated by getting trained over a subset of images which are randomly selected from the training set. The process of dictionary training is alternating in the sense that at one step the dictionary is kept fixed and the weights are calculated by sparse coding and at the next step the weights are fixed and the dictionary is updated accordingly.

The sparse approximation image compression scheme uses the input-adaptive dictionary to approximately represent the reference model as a weighted linear combination of a subset of dictionary columns. The approximate model can be compared to the reference model in terms of physical structure and fluid flow characteristics (i.e. history-matching).

In this paper, the method of optimal directions (MOD) is used to train a dictionary over a training set of FilterSim generated realizations. The dictionary is updated in alternation with least angle regression with shrinkage (LARS) method as the sparse coding method. The final trained dictionary is used to approximate the main permeability model of a reservoir located southwest of Iran under sparsity constraint. The reservoir is producing from the Asmari Formation and it is under gas-injection for pressure maintenance.

The input training image to the FilterSim algorithm is selected from a spectral decomposition profile at the top of the Asmari Formation. Two permeability models of fracture system and delta system are extracted from the spectral decomposition profile and large sets of realizations are generated using the FilterSim algorithm based on the two extracted training images. The MOD is used to update and train the dictionary over each set of generated realizations in alternation with the LARS method for the sparse coding step. The permeability models for the fracture and delta system are approximated under sparsity constraint and their physical structure are compared with the reference model. The fluid flow characteristics of the approximated models are compared with those of the reference model and the RMS values for the difference in saturation and pressure profiles are calculated. For each of the realizations in the training sets the two corresponding RMS values are obtained and the quality of the approximated models are reported as the percentage of superiority among the population space.

2. Methodology

2.1. Filter-based MPS method

FilterSim (Zhang, 2006; Wu, 2007) is a pattern-based multiple-point geostatistical method which works based on six pre-designed filters, for 2D case, in order to enhance feature extraction of the patterns and leads to effective pattern classification. In pattern-based MPS methods a template is designed to scan the training image and extract the patterns.

The template is placed on every pixel in the training image randomly and the extracted pattern corresponding to the location of each pixel is saved in a pattern database. Each visited pixel, during one run of simulation, is called a node, which is the small square located at the center of the template placed on the training image. For example, the node for the illustration in Fig. 1 is the small square colored in red. It is worth noting that the template size is always chosen to be an odd number (e.g. 5, as in Fig. 1), hence it always has a central square.

The filters apply on the patterns extracted by the template after randomly visiting the node locations on the training image. The result of applying each filter on any pattern is a score which is a scalar value and it is specified to the node on which the template is located (and from which the pattern is extracted). The default filters are designed to highlight directional pattern centers, gradients, and curvatures. For a 2D image, six filters (refer to Fig. 2) are used and are defined as follows (Zhang, 2006):

1) N-S average: \( f_1(x, y) = 1 - \frac{x}{m} \in [0, 1]; y = -m, \ldots, +m; \) for this filter, all weights are positive and large weights are applied to values in the middle of the template; thus, allowing to highlight the center of the pattern.
2) E-W average: \( f_2(x, y) = 1 - \frac{x}{m} \in [0, 1]; \) same as \( f_1(x, y) \) but rotated 90 degrees.
3) N-S gradient: \( f_3(x, y) = \frac{y}{m} \in [-1, +1] \); the weights linearly decrease from top (+1) to (-1); thus providing NS edge detection.

4) E-W gradient: \( f_4(x, y) = \frac{x}{m} \in [-1, +1] \); same as \( f_3(x, y) \) but rotated 90 degrees.

5) N-S curvature: \( f_5(x, y) = \frac{2y}{m^2} - 1 \in [-1, +1] \); the weights linearly decrease from the top (+1) to the middle (-1) and then increase again to (+1) at the bottom. This filter detects NS pattern curvature.

6) E-W curvature: \( f_6(x, y) = \frac{2x}{m^2} - 1 \in [-1, +1] \); same as \( f_5(x, y) \) but rotated 90 degrees.

Applying six filters to each pattern will result in 6 scores. These six scores will form the six features or attributes according to the node location for which the pattern is extracted. The patterns are classified based on these six scores. For a 2D training image six filters are used and for a 3D training image, by adding a component of ‘Z’, nine filters are designed and used. In the current study, because it is a 2D case, six filters are used (as described above). These filters are general filters and they are designed to apply indiscriminately on any kind of training images. A prototype is specified to each class of patterns which is in fact the pixel-based average of the patterns present in that class. Each node on the grid is randomly visited and the extracted data event is compared with the prototype of the clusters. The most similar prototype to the data event is identified and a pattern from the corresponding cluster is randomly selected and pasted on the node location. The random visit to the node locations and the random selection of the patterns from the class with most similar prototype are the means to introduce stochasticity into the realizations. The simulation stops when all nodes are visited. The FilterSim uses a multi-grid approach in order to capture the large scale features of the training image in the coarse grid and inject the large-scale properties to the finer grids. At each level, the training image is resampled to the proper resolution of the simulation grid and

**Fig. 1.** (a) 5 × 5 template; (b) binary training image (15 × 15); (c) the captured local training pattern (Zhang, 2006).

**Fig. 2.** Six default 2D filters (Zhang, 2006).
the process of pattern classification and simulation is performed separately for each grid. The FilterSim also has the ability to integrate hard and soft data. Hard data refer to the obtained information or measured properties at well locations, i.e., the well data. The hard data are certain although sparse and are strictly imposed on proper node locations in the simulation grid; hence, the information corresponding to hard data locations do not change during simulation. FilterSim specifies the hard data to the nearest node in the simulation grid. In finding the most similar prototype to the data event using Manhattan distance function, the FilterSim specifies a weight of 0.5 to the hard data, a weight of 0.3 to the frozen data which are previously simulated, and a weight of 0.2 to the nodes informed during previous simulations which are not yet frozen. On the other hand, the soft data refer to low resolution secondary information that are incorporated in simulation process as global trends, providing vital information to refer to low resolution secondary information that are incorporated in simulations which are not yet frozen. On the other hand, the soft data can be written as

\[ \lambda \left( \begin{array}{c} d \end{array} \right) = \text{argmin}_{G} \|m\|_0 \text{ subject to } \|Gm-d\|_2 \leq \epsilon \] (1)

\( \lambda \) is the sparest solution to Eq. (1) and the process of finding such a solution is called sparse approximation. Notice that the matrix \( G \) in Eq. (1) is an overcomplete dictionary.

For a set of \( N \) signals as columns of the matrix \( D \in \mathbb{R}^{n \times N} \), having the matrix \( G \in \mathbb{R}^{m \times n} \) with normalized columns, and considering the matrix \( M \in \mathbb{R}^{n \times N} (n \cdot T \ll N) \) with a number of \( s_0 \) nonzero elements in each column to be sparse enough, then the solution for \( GM \approx D \) exists, it is unique, and it is the sparsest solution that can be achieved by pursuit methods or model selection algorithms (Aharon et al. 2006a and 2006b; Horev et al., 2012; Rubinstein et al., 2010).

### 2.3. Sparse coding method

Dealing with the \( P \) norm is the main problem in sparse representation. The basis pursuit method (Chen et al., 1998) solves the sparse representation by convexicing the optimization problem in Eq. (1) through replacing the \( P \) norm by the \( l^1 \) norm. This means that the basis pursuit method solves the sparse representation problem by minimizing the number of nonzero elements in the solution. The least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996) solves the optimization problem in Eq. (1) by nullifying the elements in the solution until a definite number of nonzero elements in the solution, and \( s_0 \), is reached.

The Lasso method is similar to the forward stepwise regression method. The coefficients in forward stepwise regression are obtained by minimizing the residual sum of squares (RSS) with applying a constraint on the magnitude of the coefficients:

\[ m_{\text{stepwise}} = \arg\min_m \left\{ \sum_{j=1}^{n} \left( d_i - m_0 - \sum_{j=1}^{T} g_j m_j \right)^2 + \lambda \sum_{j=1}^{T} m_j \right\} \] (2)

where \( g_j \) is the elements of the matrix \( G \). In this equation, the scalar \( \lambda \) controls the shrinkage complexity and its higher magnitudes correspond to more shrinkage. The matrix form representation of the forward stepwise regression is:

\[ \text{RSS}(\lambda) = (d-Gm)^T(d-Gm) + \lambda m^Tm \] (3)
The input data to the forward stepwise regression must be centered and standardized. The solution to the forward stepwise regression optimization problem will be:

$$m^{\text{stepwise}} = \left( G^T G + \lambda I \right)^{-1} G^T d$$  \hspace{1cm} (4)

Comparing Eq. (4) with the least squares solution, i.e. $m^h = (G^T G)^{-1} G^T d$, indicates that the forward stepwise regression optimization problem leads to the least squares solution shrunk by a value of $(1 + \lambda)$ (Efron et al., 2004; Hastie et al., 2009).

In Lasso, the $L_2$ norm constraint applied on the magnitude of the forward stepwise regression coefficients, $\sum |m|^2 \leq t$, is replaced by an $L_1$ norm constraint, i.e. $\sum |m| \leq t$. Accordingly, the Lasso optimization problem can be written as:

$$m^{\text{lasso}} = \underset{m}{\text{argmin}} \left\{ \sum_{j=1}^{n} \left( d - \sum_{i=1}^{t} g_{ij} m_j \right)^2 + \lambda \sum_{j=1}^{t} |m_j| \right\}$$  \hspace{1cm} (5)

As in the forward stepwise regression, the inputs can be centered and reparameterized in order to eliminate $m_0$ from the formula. The solution to the Lasso optimization problem is actually a soft thresholding applied on the least squares solution such that it truncates the least squares solution at zero and translates the rest of coefficients by a factor of $\lambda$, i.e. $\text{sign}(m)|m| - \lambda$ (Efron et al., 2004; Hastie et al., 2009).

The least angle regression with shrinkage (LARS) method is presented by Efron et al. (2004) which is similar to the forward stepwise regression method but less greedy. In forward stepwise regression algorithm, the regression model is sequentially updated and at each stage, the best variable is added to the set of active variables in the regression model. The updating of the regression model is continued until all variables are used in the final regression model. The LARS algorithm is different from the stepwise regression model in the sense that it only includes a proper set of variables in the final regression model.

In the LARS algorithm, the inputs should be centered to the mean value of zero and standardized to the unit value. The algorithm starts with setting all coefficients to zero, i.e. $m_1, m_2, ..., m_T = 0$, and setting the residual to $r = d - \hat{d}$ such that $\hat{d} = \sum_{i=1}^{n} d_i / N$. The algorithm begins with a variable which has the highest correlation with the response, e.g. $m_1$. The value of $m_1$ (which is zero) is continuously dragged towards its corresponding value from the least squares solution, i.e. $\hat{m}_1$, and the residual is continuously dropping. This dragging is continued until a new variable, e.g. $m_j$, has as much correlation with the current residual as does the previous variable $\hat{m}_1$. The continuous dragging of the values for the two coefficients $m_1$ and $\hat{m}_1$ towards their corresponding least squares values (i.e. $\hat{m}_1$ and $\hat{m}_1$) is continued until a third variable, e.g. $m_2$, has as much correlation with the current residual as the joint variables $m_1$ and $\hat{m}_1$ do. After $f$ steps, the values for a number of $f$ variables are non-zero and the rest are zero. This process is repeated until a proper number of variables, e.g. $q$, are included in the regression model and the complete answer to the least squares solution is achieved after $\min(q, n-1)$ steps.

The forward stepwise regression and Lasso methods are in fact variants of the LARS method. LARS is the fastest method and the Lasso stands in the middle.

2.4. The method of optimal directions

Returning to Eq. (1), the task is to find an approximate solution to the set of equations and the answer can be obtained by the LARS method. In Eq. (1), matrix $G$ represents the explicit dictionary which is trained over a set of FilterSim generated realizations. The method of optimal directions (MOD) is one of the dictionary learning algorithms which can be used to obtain the matrix $G$ as described.

The Method of Optimal Direction (MOD) for dictionary training was proposed by Engan et al. (1999a and 1999b) and is one of the first methods to implement sparse approximation processes. The MOD is inspired by the Generalized Lloyd Algorithm (GLA). In the GLA, two steps are iterated until an optimal dictionary for vector quantization is achieved. In the first step, the GLA finds the most optimal classification for a training set by complete search using a given dictionary. In the second step, the GLA updates the dictionary using the current approximation coefficients. In the GLA, each obtained dictionary is superior to the previous one in term of the MSE and the algorithm will at least reach a local optimum. New coefficients are found using the new dictionary and iteration continues until the criterion is reached. In the MOD, however, there is no guarantee that the new dictionary is superior to the previous one. Therefore, MOD is not same as the GLA but it is inspired by the GLA and designed to learn a dictionary over a
Fig. 7. (a) Simulated saturation profile for the fracture permeability model; (b) same for the delta permeability model. (c) Thresholded (for $S_w > 0.7 \times \max (S_w)$) picture corresponding to 'a' for better visualization; (d) thresholded picture corresponding to 'b'.

Fig. 8. (a) Porosity distribution as input to reservoir simulator. (b) Permeability distribution corresponding to the stationary fracture network training image for the simulation model; (c) permeability distribution corresponding to the nonstationary delta training image for the simulation model.
training set of model images (Engan, 2000; Engan et al. 1999a and 1999b).

Generally speaking, finding an optimal transform is not a trivial task. The signals are nonstationary and the optimal dictionary suitable to span the signal space depends on the statistics of the signal space. In a Z-dimensional vector space, a set of \( T(|T > Z) \) linearly dependent vectors that span the vector space, are overcomplete and form a dictionary. Any vector in such a space can be represented as linear combination of the column vectors in the dictionary but the expansion will not be unique. Having a dictionary \( G = \{g_i\}, i = 1, \ldots, T \), a signal \( m_j \in \mathbb{R}^N \) can be represented as \( d_j = \sum_i m_j(i)g_i \), where \( m_j(i) \) is the coefficients corresponding to the vector \( g_i \). For a good approximation, many of the \( m_j(i) \) coefficients will be zero. However, if the set of vectors spanning the signal space are linearly independent, then they form a basis to span the signal space. If any two vectors from the basis are orthogonal to each other, then the basis is an orthogonal basis. Comparing to an orthogonal basis, a dictionary requires less number of vectors to represent a vector from the signal space and there is a better chance of finding such set of vectors because unlike the orthogonal transform, the selected vectors are no longer

Fig. 9. Few randomly selected FilterSim MPS generated realizations for the stationary fracture network training image.

Fig. 10. Few randomly selected FilterSim MPS generated realizations for the nonstationary deltaic system training image.
Fig. 11. (a) Few atoms from the MOD-LARS trained dictionary for the fracture network training images; (b) same for the deltaic system training images.

Fig. 12. (a) The sparsity-based compressed permeability model for fracture network using MOD dictionary training with the LARS; (b) same for deltaic system. (c) Difference between the compressed and the reference models for the fracture network; (d) same for the deltaic system.
required to be linearly independent and orthogonal to each other (Engan, 2000; Engan et al. 1999a and 1999b).

An input signal can be represented as a linear combination of the dictionary columns (atoms). The minimization objective function for this sparse representation can be written as:

$$\min_{G,m} \|d - Gm\|_2 \text{ subject to } \|m\|_0 \leq s_0$$

The MOD algorithm is a nested alternating minimization in which the inner minimization takes place over $m$ while keeping $G$ fixed and an outer minimization takes place over $G$. In the $k$th iteration, the dictionary $G(k-1)$ is used to update for the $m(k)$ by solving for

$$\min_{m} \|d - G(k-1)m\|_2 \text{ subject to } \|m\|_0 \leq s_0$$

using a greedy algorithm like Lasso or LARS, $N$ times for each signal $d_i$, $1 \leq i \leq N$. Engan et al. (1999a and 1999b) have used the MOD with OMP and FOCUSS for sparse approximation, respectively. Having the $m(k)$'s, the outer iteration is applied to update the dictionary, $G(k)$, using the least squares minimization while keeping the $m(k)$ coefficients, obtained from the previous step, fixed. This process is called the Block-Coordinate Relaxations algorithm. The inner minimization is the sparse approximation step and the outer minimization is the dictionary update.

Fig. 13. (a) Saturation profile at the observation well for the sparsity-based compressed fracture network model and its comparison with the reference model; (b) same for the deltaic system. (c) Pressure profile at the observation well for the sparsity-based compressed fracture network model and its comparison with the reference model; (d) same for the deltaic system.
For a set of data $D = [d_1, d_2, ..., d_N]$, after each iteration the dictionary columns are normalized to avoid scaling issues and the error $\|D - GM\|_F^2$ is calculated, where $M = [m_1, m_2, ..., m_N]$. Having $R = [r_1, r_2, ..., r_N]$, and defining $R_i = D_i - GM_i$, at each iteration the dictionary is updated as $G_i = G_{i-1} + (R_i M_i^T) (M_i M_i^T)^{-1}$. If the error $\|D - GM\|_F^2$ becomes smaller than $\epsilon$, the algorithm is converged and it will stop. The dictionary $G$ and the sparse coefficients $M$ are then reported as the products. The MOD is an effective method which needs a few iterations to converge but suffers from relatively high complexity of the matrix inversion (Engan, 2000; Engan et al. 1999a and 1999b; Skretting and Engan, 2011).

Having a set of data $(D_i)_{i=1}^N$, form an initialization dictionary for which the entries are random numbers or randomly selected from the observations; normalize the dictionary columns. Set $itr = 0$.

While $\|D - G^{itr} M^{itr}\|_F^2$ is not less than $\epsilon$

Sparse coding stage:
- For each signal data, $D_i (1 \leq i \leq N)$, approximate $M_i^{itr}$ using any model selection method like Lasso or LARS ($M_i$ is the $i$th sparse column of $M$).
- $M_i^{itr} = \arg\min_{M_i} \|D - G^{itr-1} M_i^{itr-1}\|_F^2$ subject to $\|M_i\|_0 \leq s^0$

MOD dictionary update stage:
- Update the dictionary $G^{itr}$ using the dictionary from previous iteration:
- $G^{itr} = \arg\min_{G} \|D - G^{itr-1} M_i^{itr-1} - (D M_i^{itr-1}) (M_i M_i^T)^{-1} M_i^{itr-1}\|_F^2$
- Normalize the dictionary $G$.
- Set $itr = itr + 1$;
- Output: $M$ and $G$.

The MOD algorithm; modified after (Elad, 2010).

2.5. Sparse compression with MOD and LARS

The dictionary, $G$, trained over a training set of FilterSim generated realizations can now be used in a sparse approximation scheme, $GM \approx D$, to approximate the permeability distribution of a petroleum reservoir. The compression scheme involves two alternating algorithms, an offline learning algorithm using the MOD method and an online sparse coding scheme using the LARS method.

To apply such scheme on the training set, the model images are first partitioned into smaller patches and for each patch, the mean value is calculated and deducted from the cell values. The nonzero values in each patch are then quantized using a uniform quantizer function. The Entropy, Arithmetic, and Huffman coding are also applied on patches. The coded patches are then used in the MOD dictionary learning algorithm to train the dictionary. As previously stated, the trained explicit dictionary $G$ has the advantage of adaptability and effectiveness on the type of images that it is trained for but lacks the properties of generality, fastness, and internal structure of the implicit dictionaries. In order to exploit the advantages of both types of dictionaries, an input-adaptive dictionary is defined as $C = \phi G$ where $\phi$ and $G$ represent implicit and explicit dictionaries, respectively. In this setting, $\phi$ applies on the $G$ matrix to represent the atoms in $G$ as a sparse representation of the columns of the implicit dictionary, $\phi$. The reason for this is that although the explicit dictionary is known to be unstructured as compared to the implicit dictionaries (which are strictly structured and regular), but there is still a visible structure inherent within the columns of the explicit dictionaries. This inherent structure within the explicit dictionaries can be extracted by representing its atoms as sparse approximation of bases (columns) of some implicit dictionary.

Therefore, the matrix $C$ is a new input-adaptive dictionary with parametric structure which take advantage of the internal structure of the implicit dictionary $\phi$ as a regularizer in the process of dictionary learning and avoids overfitting and instability in the presence of noise. The atoms of matrix $C$ are in fact a sparse combination of the atoms in the matrix $\phi$. In other words, matrix $C$ is an extension of matrix $\phi$ with an added layer of adaptability which comes from matrix $G$ (Horev et al., 2012). Accordingly, matrix $C$ benefits both from the properties of an implicit dictionary $\phi$, which is highly structured and has fast numerical implementation, and an explicit dictionary $G$, which is a much finer-tuned dictionary comparing to $\phi$ and it is more effective in reconstructing the type of images similar to the training set. Consequently, generating a new matrix such as $C = \phi G$ will bridge the gap between the two types of implicit and explicit dictionaries and exploits the advantages of both dictionaries.

In this study, the discrete cosine transform (DCT) is used to act as a regularizer, i.e., matrix $\phi$, in the process of dictionary learning. Selecting the DCT as the implicit dictionary in this paper is only a matter of choice. For a specific set of data, the most suitable implicit dictionary can be determined by performing sensitivity analysis and studying the effect of each implicit dictionary on the final reconstructed model and its fluid flow properties. Performing a sensitivity study is beyond the scope of this paper.

Coefficients of the resultant input-adaptive matrix $C$ is then quantized and reshaped and used in the LARS algorithm to obtain the LARS coefficients as it was described before. Image compression scheme through sparse approximation schemes has the advantage of finding and capturing the data redundancy.

3. Results

In this paper, the image compression scheme via sparse transform is applied on a real petroleum reservoir located southwest of Iran. The reservoir has been under gas injection through one well for pressure maintenance and the gas-oil-ratio and solution-gas-oil-ratio has increased in the nearby wells since then. An early breakthrough of the injected gas is suspected by the reservoir engineers according to the production and pressure data and they explain the high-porosity and high-permeability interbeds of sandstone as conductive paths to have caused the early breakthrough of the injected gas in the observation wells. This study was primarily invoked to characterize the reservoir and the main focus was on the rock properties. Seismic, well log, and reservoir data were then integrated and interpreted to detect the main reason for the early breakthrough of the injected gas.

This study represents three levels of results; results which delineate the rock properties, results which characterize the fluid distribution of the reservoir, and the results concerning the application of sparse transform compression scheme.
3.1. Results for the rock properties

The spectral decomposition cube extracted from the seismic data at the top of the Asmari formation is depicted in Fig. 3. The reservoir is oriented northwest to southeast. Several features are observable in this picture. A depositional environment of the deltaic system has been detected west of the reservoir, indicated by red lines. The continuous green line shows the crest of the anticline; according to the spectral decomposition profile, the crest of the reservoir is not fractured but on the two sides of the crest line, a fracture set is observable. The fracture sets are parallel to the crest line and they follow the bends of the crest line. This feature is most indicative to a set of conjugate fractures. The fractures themselves, within the fracture set, are indicated as parallel lines and noting the seismic resolution, these fractures are mostly longitudinal.

The acoustic impedance cube inverted from the seismic data at the top of the Asmari reservoir is illustrated in Fig. 4. Diverging fingers of the delta system are observed on this picture and they are shown by black lines. The continuous dark purple line represents the crest of the reservoir which indicates low acoustic impedance. A spot of high acoustic impedance at the crest and its surroundings, shown by the red circle, indicates the position of the gas injection well. The area in red color, shown in parallel lines above and below the crest line, represent high acoustic impedance. A fault from northwest of the reservoir is joining the northern set of fractures. This fault is indicated by a red long rectangle.

The density vertical profile indicates interbeds of layers with different density values. The extracted seismic porosity cube indicates two high porosity and two low porosity layers interbedded such that the lower porous layer is caked between the two nonporous layers. Furthermore, a horizontal slice from the extracted seismic porosity indicates an increase in porosity from the west of the reservoir to the center and east.

3.2. Results for the fluid distribution

The water saturation cube is extracted from the seismic data. A demonstration of this cube with water saturation <50% ($S_w < 50\%$) is illustrated in Fig. 5. In this figure the two oil-bearing intervals as illustrative. The upper reservoir layer, depicted with light blue, is already known and producing; however, the lower reservoir layer, shown as dark blue, is discovered in this study and it is untouched. Furthermore, a vertical profile of the water saturation cube indicates that the reservoir under study is a double layered reservoir which is pushed from edges by two water aquifers. The water saturation cube for values <20% is depicted in Fig. 6. This depiction corresponds to the gas bearing zone evolved from well No. 8 and distributed to the sides. The gas is injected into the reservoir from well No. 8. The cube of water saturation <10% also indicates similar results but with narrower gas-swept areas. A fluid flow simulation for the fracture system and delta system models are performed and the simulated saturation maps for both models are depicted in Fig. 7a and b, respectively. Fig. 7c and d are the thresholded versions of Fig. 7a, as compared with that of the delta model (Fig. 7d), is more similar to the observed gas saturated area (Fig. 6) extracted from the seismic data. The above statements provide enough evidence for reliability of the interpretation made from integrated reservoir data studies.

In order to investigate the effectiveness of the sparse transform compression scheme for reconstructing the reference model based on training a dictionary over a training set of MPS generated realizations, the outmost stochasticity is introduced into the generated realization by not conditioning them with hard or soft data. In addition to this, no scaling and rotation conditioning was applied for the delta system training image which is assumed to be nonstationary. Figs. 9 and 10 indicate that the generated realizations, while preserving stochasticity, fairly represent the fracture and delta system models. A training set of 3000 realizations for the fracture system are used to learn a dictionary based on the MOD method in conjunction with the LARS algorithm as the sparse coding methodology. Few atoms of the trained dictionaries are depicted in Fig. 11a and b.

The trained explicit dictionaries are used as matrix $C$ in the combination matrix of the form $C = \phi G$ with the DCT implicit dictionary to approximate the permeability model under sparsity constraint. The approximated models are shown in Fig. 12a and b, for fracture and delta systems respectively. Two criteria are used to check the quality of the approximated models; the physical similarity check and the fluid flow characteristic check. The difference between the main and approximated models are depicted in Fig. 12c and d for fracture and delta systems, respectively. The water saturation and pressure profiles for the approximated models are also illustrated in Figs. 13a to 13d and they are sketched against saturation and pressure curves of the reference models for fracture and delta systems, respectively. The MATLAB code by Lie, 2015, is used as the reservoir simulator.

4. Discussion

The input training images to the FilterSim MPS method must be based on reliable models of the reservoir. Maximum amount of information must be extracted from integrated studies to generate the most reliable models. In the acoustic impedance profile (Fig. 4) the red area indicates the part of the reservoir where the fracture intensity is high and the reunion of the northwestern fault with the fracture system on northern flank is an additional indication for presence of the fracture set in that area. Furthermore, the porosity change from the porosity cube is compatible with the change in depositional system demonstrated on both acoustic impedance and spectral decomposition profiles in Figs. 3 and 4. This lateral change in porosity might explain the inconsistent rate of pressure drop for the west of the reservoir as compared with the center and east of the reservoir due to production. The two oil-bearing layers detected on the saturation profile (Fig. 5) are compatible with the two porous zones extracted from the porosity cube. Additionally, the geometry of the gas front extracted from the seismic data in Fig. 6 copies the feature of conjugate fracture systems from Figs. 3 and 4. Finally, the comparison of the two simulated saturation profiles for the fracture and delta systems (Fig. 7a and b) illustrates that the figuring effect observed for the fracture model (Fig. 7c), as compared with that of the delta model (Fig. 7d), is more similar to the observed gas saturated area (Fig. 6) extracted from the seismic data. The above statements provide enough evidence for reliability of the interpretation made from integrated reservoir data studies.

In order to investigate the effectiveness of the sparse transform compression scheme for reconstructing the reference model based on training a dictionary over a training set of MPS generated realizations, the outmost stochasticity is introduced into the generated realization by not conditioning them with hard or soft data. In addition to this, no scaling and rotation conditioning was applied for the delta system training image which is assumed to be nonstationary. Figs. 9 and 10 indicate that the generated realizations, while preserving stochasticity, fairly represent the fracture and delta system models. A training set of 3000 realizations for the fracture system are used to learn a dictionary based on the MOD method in conjunction with the LARS algorithm as the sparse coding methodology. Few atoms of the trained dictionaries are depicted in Fig. 11a and b which do not indicate an internal structure in contrary to the implicit dictionaries, e.g. DCT. The application of the input-adaptive dictionary $C = \phi G$, as defined before, with the LARS sparse coding method results in structurally appealing approximate models. The background color in the difference maps of the main and approximated models, depicted in Fig. 12c and d,
represents a difference of zero for the errors corresponding to structural similarity of the main and reconstructed models. The maximum and minimum difference values are around 20% of the highest pixel value of the reconstructed model images, which is appealing. Additionally, the structural discrepancies are not correlated and they are scattered throughout the difference map (Fig. 12c and d) which is another indication for the good performance of the compression scheme in reconstructing the reference model.

As far as the fluid flow characteristics are concerned, the water saturation curves indicate close correspondence between the approximated and reference models in both fracture and delta systems cases (Fig. 13a and b). This is the case in most experiments performed by the authors. However, for the pressure curves the discrepancies are more drastic (Fig. 13c and d). For the fracture system case this discrepancy is larger; note that the difference between main and approximated models is larger for the case of fracture system model (Fig. 12c). One reason for such an observation might be that, referring to realization models for both fracture system and delta system models which are selected in random (Figs. 9 and 10), the fracture system realizations are more similar to each other than the delta system realizations. Accordingly, the atoms in the trained dictionary captures and represents more diversity in the realization models for the delta system case as compared with that in the fracture system case. Hence, the dictionary trained by the delta system is able to reconstruct the approximated model for the delta system with more accuracy in terms of physical structure and fluid flow characteristics. Further inferences and justifications are mean to a comparison study where the MPS realization models are once generated under hard/soft data conditioning versus realization models with no conditioning.

Ultimately, the superiority of the fluid flow characteristics for the approximated models as opposed to the population space is investigated by comparing the RMS error of the saturation and pressure profiles for the approximated models and each of the MPS generated realization models against the saturation and pressure observations from the reference model as the benchmark. Such study indicates that the approximated fracture model outperforms 86.16% of the population space in terms of fluid flow characteristics. This means that out of a total number of 3000 MPS realizations, 355 realizations result in a better history-match than the approximated model. The same comparison for the nonstationary image of the deltaic system indicates a superiority over 99.17% of the MPS realizations in the corresponding population space, which corresponds to a number 24 MPS realizations in the population space outperforming the delta system approximate model. Based on the results of a set of 48 experiments performed within a one-at-a-time experimental design scheme, presented in Fig. 14, the probability of obtaining an approximate model that outperforms 90% of the population space is equal to 89.58%. These experiments are performed with different dictionary learning algorithms, sparse coding, and MPS methods based on the training images described in this paper (Fig. 8b and c).

5. Conclusions

In this paper, the sparse transform compression scheme was studied and applied on a petroleum reservoir located southwest of Iran. The sparse transform is an alternation between two steps of dictionary updating and sparse coding. The dictionary is updated by learning over a set of realizations generated by the FilterSim MPS method using the method of optimal directions.

To generate a set of FilterSim generated realizations a reliable training image from the reservoir is required. An interpretation of the reservoir is presented by integrating the seismic, well log, and reservoir data. The rock properties and fluid content characteristics of the reservoir were delineated by the integrated study. A new oil-bearing reservoir layer was discovered as the result of this study and the discovery was approved by drilling and perforating a new oil well into the suggested interval. The fluid content for the discovered interval was discovered to be heavy hydrocarbon, further validating the results of the integrated study as the cap rock for the lower oil-bearing interval is a nonporous interval of the reservoir rock which has led to migration of the light hydrocarbon components to the upper oil-bearing interval.

Based on this integrated study, two semi-industrial permeability models were extracted from a seismic spectral decomposition profile, representing a fracture system and a delta system model, and used as training images in the FilterSim MPS method. Based on each training image, a large set of realizations were generated which are used as training set to the dictionary updating in the sparse transform compression scheme. Upon each training set, a dictionary was trained using the method of optimal directions, as the dictionary learning algorithm, in alternation with the least angle regression with shrinkage, as the sparse coding method. An input-adaptive parametric dictionary was constructed based on the MOD trained dictionaries for each training set, in conjunction with the DCT implicit dictionary as the dictionary regularizer.

The resultant input-adaptive dictionary was then used to approximate the main permeability models for both fracture and delta systems using the LARS algorithm for sparse coding. The reconstructed approximate models for both fracture and delta systems are in good accordance with the reference models. However, the reconstructed model for the case of delta system indicates better quality both in terms of physical structure and fluid flow characteristics as compared with the reconstructed fracture model. The quality of the approximated models is also checked within the population space for each case. For the case of fracture system, the approximate model outperforms 86.16% of the 3000 realization models in the population space in terms of fluid flow characteristics. For the case of delta system, the approximate model is superior to 99.17% of MPS realization models in the population space.

Based on the results of this study, the sparse transform compression scheme can reconstruct approximate models which are close to the reference models both in terms of physical structure and fluid flow characteristics. Further enhancements in the results require parameter optimizations for the three blocks of dictionary learning, sparse coding, and multiple-point geostatistical modeling.

Acknowledgement

This work was performed at the Institute of Geophysics (University of Tehran). The authors have enjoyed discussions with individuals from the Geology, Geophysics, and Reservoir Engineering departments at the National Iranian South Oil Company. The authors express gratitude towards the individuals from the named institutes. We also gratefully appreciate the scientific works of the scholars who are referenced in this article.

References


