



Sparsity-based compressive reservoir characterization and modeling by applying ILS-DLA sparse approximation with LARS on DisPat-generated MPS models using seismic, well log, and reservoir data

5 Mohammad Hosseini^{1,2}, Mohammad Ali Riahi¹

¹Institute of Geophysics, University of Tehran, Tehran, 14155-6466, Iran

²Geophysics Department, National Iranian South Oilfields Company, Ahwaz, 61735-1333, Iran

Correspondence to: Mohammad Hosseini (hosseini.mhmd@yahoo.ca)

Abstract. In the earth sciences, there is only one single true reality for a property of any dimension whereas many realization models of the reality might exist. In other words, a set of interpreted multiplicities of an unknown property can be found but only one unique fact exists and the task is to return from the multiplicities to the uniqueness of the reality. Such an objective is mathematically provided by sparse approximation methods. The term 'approximation' indicate the sufficiency of an interpretation that is close enough to the true mode, i.e. reality. In geosciences, the multiplicities are provided by multiple-point statistical methods. Realistic modeling of the earth interior demands for more sophisticated geostatistical methods based on true available images, i.e. the training images. Among available MPS methods, the DisPat algorithm is a distance-based MPS method which generate appealing realizations for stationary and nonstationary training images by classifying the patterns based on distance functions using kernel methods. Advances in nonstationary image modeling is an advantage of the DisPat method. Realizations generated by the MPS methods form the training set for the sparse approximation. Sparse approximation is consisted of two steps, i.e. sparse coding and dictionary update, which are alternately used to optimize the trained dictionary. Model selection algorithms like LARS are used for sparse coding. LARS optimizes the regression model sequentially by choosing a proper number of variables and adding the best variable to the active set in each iteration. Out of numerous training dictionary methods given in the literature, the ILS-DLA is a variant of the MOD algorithm where the latter is inspired by the GLA and the whole trained dictionary is sequentially updated by alternating between sparse coding and dictionary training steps. The ILS-DLA is different from the MOD for addressing the internal structure of the dictionary by considering overlapping or non-overlapping blocks and modifying the MOD algorithm according to the internal structure of the trained dictionary. The ILS-DLA is faster than the MOD in the sense that it inverts for smaller blocks constructing the trained dictionary rather than inverting for the entire block. The subject of this paper is an integration study between sparse approximations from image processing and compressed sensing, multiple-point statistics from the field of geostatistics, and the geophysical methods and reservoir engineering from the branch of petroleum science. This paper specifically emphasizes the utilization of image processing in solving reservoir complexities and enhancing reservoir models.

35



1 Introduction

In geoscience, there is only one single true model for a property of the earth but numerous models of the real earth property exists. This concept is referred to as *uncertainty* in the geoscience and reservoir modeling. In geoscience and reservoir modeling it is mostly acknowledged to manipulate the models and find the best one among a finite set of possible models by performing further experiments and observing the results. It is neither guaranteed that the best model of a finite set of models is actually the true model nor that it is the best practically achievable model. Trivially, the more available models as the population samples, the more it is guaranteed that the selected best model is a closer translation of the reality, i.e. the true model.

1.1 Model multiplicities by the MPS methods

In reservoir characterization and modeling, providing a large set of population samples is practically impossible unless introducing stochasticity in the models using the multiple-points statistics (MPS) methods. It is known that simple two-point geostatistics methods (Deutsch and Journel, 1998; Strebelle, 2000) fail to properly generate stochastic models especially for complex reservoirs, e.g. a reservoir which is deltaic in one part and fractured carbonate ramps in other parts. Delta properties are nonstationary and fracture network properties can also be nonstationary. Two-point geostatistics methods use variogram to model property variations between two points in the reservoir based on spatial distance. It will fail to properly model for farther distances if the changes in property is abrupt or nonstationary. To alleviate, different variogram models can be built for a neighborhood to capture property variations, but even this method does not completely capture the variations.

As a more advanced variant, a location-independent model can be used as a source of information and statistical changes for multiple-point statistics (MPS) algorithms to train and produce reliable reservoir static models. This location-independent model is called training image and its location-independence property will lead to producing various stochastic reservoir models (Haldorsen and Damsleth, 1990; Mariethoz and Caers, 2015). More realistic training images and more reliable stochastic models will be generated if wide and accurate prior information about the reservoir is available. Pattern-based methods like SimPat (Arpat, 2005), FilterSim (Zhang, 2006), Direct Sampling (Mariethoz and Renard, 2010), DisPat (Honarkhah, 2011), and WaveSim (Chatterjee et al., 2012) are generally best methods for MPS reservoir modeling. The SimPat algorithm is time-consuming and in the FilterSim, defining the filters adds complexity to the algorithm. The WaveSim algorithm is similar to the FilterSim and its complexity is in selecting an optimum scale for wavelet decomposition in the task of pattern classification. DisPat algorithm is known as the best of this kind for its ability to integrate data based on visual system of the human being and representing new algorithms for modeling images with nonstationary properties.

1.2 Sparse approximation as mathematical tools to sparsity-based compressed model

The MPS methods are used to generate the set of multiplicities based on one single training image which itself is not known as the true model. Therefore, the MPS generated realizations are all translation of an unknown true model. The task is to achieve one single model image as the representation of the true model from a large set MPS realizations which are considered as the manipulations of the true model. The mathematical tools to perform such a task is known as the sparse approximation from the field of image processing. The term *approximation* reminds the impossibility or unnecessary of obtaining the exact true model.



In a wider branch of image processing, the large set of stochastic reservoir MPS model images (generated by any of the MPS methods) could be used as a set of training images for dictionary training in the context of image compression (Aharon et al., 2006a and 2006b; Bryt and Elad, 2008; Cheng, 2015; Elad, 2010; Elad and Aharon, 2006; Horev et al, 2012; Khaninezhad and Jafarpour, 2013; Mairal et al., 2009 and 2008; Rubinstein et al., 2010a, 2013, 2010b and 2008; Skretting and Engan, 2011a, 2011b and 2010; Skretting and Husøy, 2003; Skretting et al., 1999; Starck et al., 2010). There are generally two types of dictionaries that can be employed to compress a set of model images: explicit and implicit dictionaries. The explicit dictionaries are out of the shelf dictionaries with fixed elements usually explicitly stated by analytical equations. The implicit dictionaries are different from the explicit ones in the sense that they are trained and adapted by the specific set of training images. Therefore, the explicit dictionaries are general but less effective and the implicit dictionaries are trained on specific training images but more effective on that set or similar sets of training images.

The process of sparse approximation basically deals with implicit dictionaries which are trained and adapted on specific sets of training samples (images or signals). The sparse approximation is itself a sequential alternation between two steps: sparse coding and dictionary update. The sparse coding step deals with selecting the best vector (or model) from a large set of vectors (or models) which minimizes an optimization problem under some sparsity constraint. In other words, in each iteration, the sparse coding methods like BP (Chen et al., 1999), MP (Mallat and Zhang, 1993), OMP (Rubinstein et al., 2010b; Tropp, 2004; Tropp and Gilbert, 2006), and ORMP (Gharavi-Alkhansari and Huang, 1998) and model selection algorithms like LARS (Efron et al., 2004; Tibshirani, 1996) and Lasso (Hastie et al., 2009) are employed using a fixed dictionary to obtain the sparse coefficients. The obtained sparse coefficients are alternately used in the dictionary update step.

Different methods of dictionary updating are presented in the literature, e.g. MOD (Cheng, 2015; Elad and Aharon, 2006), K-SVD (Aharon et al., 2006b; Bryt and Elad, 2008; Rubinstein et al., 2010b; Skretting and Engan, 2011a and 2011b), ODL (Mairal et al., 2009), ILS-DLA (Engan et al., 2007; Skretting and Engan, 2011b), and RLS-DLA (Skretting and Engan, 2011a and 2010). The MOD algorithm is inspired by the GLA to sequentially update the trained dictionary by alternating between the sparse coding and dictionary update steps inverting for the whole set of training data. The K-SVD algorithm is the generalization of the k -means clustering algorithm. In the K-SVD, the dictionary is updated sequentially by applying the singular value decomposition on each column of the trained dictionary in each iteration. The Online Dictionary Learning algorithm also follows the two step procedure and it is specifically designed to handle large scale data sets under the scope of online optimization algorithms. The ODL is fast for being based on stochastic approximation and updating the dictionary recursively by using the previous one as the warm restart. The ILS-DLA is a variant of the MOD algorithm and it is different from the MOD for addressing the internal structure of the dictionary by considering overlapping or non-overlapping blocks and modifying the MOD algorithm according to the internal structure of the trained dictionary. The ILS-DLA is faster than the MOD in the sense that it inverts for the smaller blocks which are constructing the trained dictionary rather than inverting for the whole block. The RLS-DLA exploits the same methodology as in the MOD algorithm where the RLS-DLA starts with an initial dictionary and at each step a new vector of data is introduced into the training set and the dictionary is updated accordingly. A forgetting factor is introduced into the algorithm to forget the early stages dictionaries and focus on the late stages dictionaries.



1.3 The case study

In this paper, the sparsity-based compressive reservoir characterization and modeling workflow using the ILS-DLA training dictionary with LARS sparse coding methodologies is explained and applied on a gas injection case of an Iranian oilfield located southwest of Iran. The gas is injecting into the Asmari reservoir of this oilfield to maintain pressure and preserve the recovery factor. The Asmari reservoir in this oilfield is described as a very complicated reservoir because of the presence of interbeds of sandstones and carbonates, the possibility of a fracture network connecting the reservoir along the crest, and the presence of a deltaic depositional system in the west of the reservoir. Modeling such a complicated reservoir demands for new stochastic MPS methods rather than simpler two-point geostatistical methods.

5

10

15

20

25

The subject of sparsity-based compressive reservoir characterization and modeling is an integration study which involves the geostatistics (for multiple-point statistics modeling), image processing (for dictionary training, sparse approximation, and image compression), seismic data interpretation and inversion (for providing AI along with saturation and porosity inverted 3D cubes, seismic spectral decomposition 3D cube, interpreted horizons),

30

petrophysics (for providing well log data and interpretation), and reservoir simulation (for providing criteria to quantify the goodness of model image).

1.4 Paper structure

In this paper, first the major blocks of the sparsity-based compressive workflow in reservoir characterization and modeling is presented. These main blocks are consisted of the DisPat multiple-point statistics method and the

35

sparse approximation algorithm. The sparse approximation algorithm is consisted of sparse coding and dictionary training steps. For the sparse coding step, the LARS algorithm and its variants are introduced and for the dictionary update step, the ILS-DLA algorithm is represented. The application of the sparsity-based compressive workflow



on a case study is discussed afterwards and the results are presented. The paper is finalized by some concluding points.

2 DisPat MPS Algorithm

5 DisPat is a distance-based pattern-based multiple-point geostatistical method which was first introduced by Honarkhah (2011). In this algorithm, alike the other pattern-based MPS methods, the training image is scanned by the designed template and the patterns are extracted from the training image. The patterns are then classified based on distance functions using the kernel methods. A distance function measures the distance between each pair of patterns in the metric space. Any two close points in the metric space refer to two similar patterns from the pattern database. The kernel k -means clustering algorithm is used to classify the patterns in the pattern database.

10 2.1 Pattern Simulation

Having the patterns classified, the sequential simulation is performed on the realization grid. The closest cluster to the data event is determined and one pattern from that cluster will be randomly chosen to be replaced on the node location. The distance function which is used to find the closest cluster to the data event has to account for the informed nodes, frozen nodes, and hard data nodes by specifying different weights to each type of the data. The SimPat method uses Euclidean distance function to find the most similar pattern to the data event. Filtersim method uses the Manhattan distance function whereas DisPat algorithm uses two distance functions for two different purposes; the proximity distance transform is used to build the distance matrix and the Manhattan distance function is used to find the most similar pattern to the data event.

2.2 Multigrid and Multiresolution

20 To improve the long-range spatial characteristics of the realizations, DisPat utilizes the widely used multigrid approach and the newly invented multiresolution approach. In the multigrid approach, the grid size does not change but the template size changes from one level to another level. Sequential simulation starts from larger scale and the final realization at each level will be transferred to the lower level considered as informed nodes. As a result, large-scale properties are transferred from one grid level to the next finer grid level. In the multiresolution method, 25 on the other hand, the template size is fixed and the grid size is changing from one level to the next. The simulation starts from the coarsest resolution and continues to the last finer resolution grid. The final realization at each level is informed to the next resolution level where a one-to-one correlation between two consequent grids is not held. Correspondingly, the realization and the training image are rescaled and interpolated to the next resolution level.

2.3 Hard Data Integration

30 The DisPat algorithm considers two hard data integration scopes for the two multigrid and multiresolution approaches. For the multigrid approach, the DisPat will strictly specify each hard data to the closest possible node and if the node has been introduced before, it will be specified to the next closest enclosing node. The hard data which are left out of this procedure will be rejected. In the multiresolution approach, the grids are different in size and the location of nodes changes from one level to the next level and a one-to-one correlation does not hold. The 35 cubic interpolation is used to transfer the hard data from one level to the next level. For interpolation, either kriging



or inverse-distance weighting method is used. Hard data integration based on multiresolution approach will result in more appealing MPS realizations. The reason for such an improvement is that the multiresolution approach captures the statistical information of the training image better than the multigrid approach. Furthermore, the inverse-distance weighting will better represent the data structure in lower resolution grids.

5 2.4 Soft Data Integration

Hard data refers to well data and soft data refers to seismic data. In SimPat, along with the training image, a soft training image is considered and the soft pattern data base is sought and formed as it is done for the training image itself. A weight is given to the soft data to diminish or increase its role in building the final pattern based on the reliability weight given to the soft data. In Filtersim, Zhang (2006) and Wu (2007) proposed a new method for continuous images. They do the conventional unconditional simulation and find the most similar pattern for the node location. Simultaneously, the corresponding soft data event is found and pasted on the uninformed part of the data event and hence, the soft data is injected into the final simulation. For categorical images, the method in SimPat is integrated with the τ model from the SNESim and a new data event is produced. The algorithm searches for the most similar prototype to the data event and the τ model is used to combine this data event with the soft data event. Another search is done to find the closest prototype to this newest data event produced by the τ model. In DisPat, the soft data integration is performed through distance calculations. The pattern databases are formed for the two sets of training image and the soft training image, patterns are clustered and the prototypes are specified. For each data event on a node location, the distance to each of the prototypes in either sets of the data (training image and the soft data) are calculated. These two distances are combined to form a new distance function in which the soft distance (i.e. the distance of the data event to the prototypes of the soft data) is weighted. This specified weight is itself a factor of the soft data reliability multiplied by the fraction defined as the ratio of the number of the informed nodes in a data event to the total number of nodes in that data event. It can be proven that this distance-based soft data integration is a special case of the τ model.

2.5 Nonstationarity

Training images could be stationary or nonstationary. In a stationary image the patterns are repeated and the statistical properties are steady, whereas in a nonstationary image, no prior information is expected for any location. In SimPat, a reservoir is subdivided into different regions upon the user's decision and based on the morphological characteristics of the reservoir. Each subregion is scaled or rotated such that the transition to the adjacent subregions are smooth and gradual. In Filtersim, additional morphological information is provided for grid nodes through the two other distinct grids conveying the scaling and rotation information. The drawback to these methods is the discontinuities which appear at the borders of these subregions where their corresponding training images are not coherent. Honarkhah (2011) has proposed three algorithms to generate nonstationary realizations needing only the training image as the prerequisite.

In the Spatial Similarity Method (SSM), the DisPat algorithm saves the patterns in the database along with the location of each pattern in a separate location database. This method is based on the fact that the patterns closer to the data event are more similar to it than the farther patterns. The SSM searches all over the grid for the most similar pattern. The most similar pattern to the data event is the one which minimizes a distance function and is



defined as combination of the weighted pattern distance with the weighted location distance. These weights are complementary and add up to 1. The neighborhood-radius method (NRM) is principally similar to the SSM method except that the algorithm searches within a definite circle of neighborhood and during the simulation, similar patterns inside this neighborhood will be searched for. The distance function will be the distance between the data event and the patterns; the location criteria is applied on the boundary within which the patterns are selected. This method is most suitable for semi-stationary training images where the circle of neighborhood controls the distance for which the statistical properties can be assumed stationary. The automatic segmentation method (ASM) is used for highly nonstationary images in which stationary regions cannot be defined for any border locations. As a result, such images must be automatically segmented into supposedly stationary regions. The approach to such a task is to extract several features of the property in the image and cluster the training image based on these extracted features by the k -mean clustering method. Gabor filter banks are used to extract image properties in few orientation angles at constant frequencies. The energy of each filtered images is then exposed by applying the sigmoid function and convolving with the Gaussian function as the smoothing filter. Additional Spatial component images are added to the set of features as input to the k -mean clustering algorithm. The k -means clustering algorithm will automatically segment the training image into k subregions based on the set of input features. The number of subregions, k , is defined by the user.

3 Sparse Approximation via Dictionary Learning

The exact way to represent the signal $d \in \mathbb{R}^n$ as a linear combination of the columns of matrix $\mathbf{G} \in \mathbb{R}^{n \times k}$ is given by $\mathbf{G}m = d$, where $m \in \mathbb{R}^k$. The sparsest solution to this problem is expressed as $\min_m \|m\|_0$ for which the $\|\cdot\|_0$ counts the number of non-zero entries in the solution. The most desired solution to this system of equations is the one with the fewest number of nonzero coefficients.

In an image processing problem, matrix \mathbf{G} is called dictionary and its columns are called the atoms. These atoms are the building blocks of the dictionary and are normalized. Vector d can be represented as a linear combination of the dictionary atoms, $\{G_j\}_{j=1}^k$. Curvelets, contourlets, wedgelets, bandlets, steerable wavelets, and short-time Fourier transforms are examples of the dictionary matrices. This representation is either exact or approximate and the exact representation of the vector d is expressed as

$$\min_m \|m\|_0 \quad \text{subject to} \quad \mathbf{G}m = d \quad (1)$$

The solution to the exact problem is $Q(\mathbf{G}^{-1}d)$ and this is only achievable if \mathbf{G} is invertible. In real problems, the exact solution is usually unachievable or it is not favorable and the approximate solution suffices. Obtaining the approximate solution to $\mathbf{G}m \approx d$ which bears the criterion of being the sparsest, is called the sparse approximation. The formulation for sparse approximation is represented as

$$\min_m \|m\|_0 \quad \text{subject to} \quad \|\mathbf{G}m - d\|_2 \leq \epsilon \quad (2)$$

In this case, matrix \mathbf{G} is an overcomplete dictionary. The whole process is called compression technique. In the well-known cases of transform coding, the DCT or wavelet dictionaries are used to compress the data. The compression process finds and captures the redundancies in the data. For a set of N signals, if the columns of $\mathbf{G} \in \mathbb{R}^{n \times k}$ are normalized and $m \in \mathbb{R}^{k \times N}$ ($n < k \ll N$) is sparse enough with s_0 non-zeros in each column, the solution



to the factorization $\mathbf{G}\mathbf{m} = \mathbf{d}$ is guaranteed, it is unique, and it is the sparsest solution and can be achieved by the pursuit methods. Here, $\mathbf{d} \in \mathbb{R}^{n \times N}$ ($n \ll N$) is the set of signals containing N signals as its columns and \mathbf{m} is the compressed solution and it certainly has lower entropy than the signal \mathbf{d} (Aharon et al., 2006a and 2006b; Horev et al, 2012; Rubinstein et al., 2010b).

- 5 Sparse transform has wide applications in compression, feature extraction, regularization in inverse problems, denoising, dynamic range compression in images, separation of texture and cartoon content in images, inpainting, facial imagery and more.

Optimal dictionaries are considered as flexible, simple, efficient for which the objective functions are well-defined. It is appropriate to have a structured dictionary rather than a free one. The proper dictionary is either drawn from
10 a prespecified set of linear transforms or it is adapted to a set of training signals. The prespecified dictionaries are simple and fast and can be easily pseudoinversed if tight frames are used. These predetermined dictionaries are also called analytic dictionaries because the algorithm to derive the coefficients is known and analytically stated. For the same reason, they are called implicit dictionaries because the coefficients are implicit analytical statements instead of explicitly stated numbers. These dictionaries are highly structured, practically fast and efficient but they
15 lack adaptability and they are only used for general-purpose image compression tasks.

Alternatively, the trained content-specific dictionaries outperform the prespecified ones. One drawback to the content-specific dictionaries is its computational time. Whereas this drawback is downplayed nowadays by the ever-growing computational capabilities, but still their drawback of loss of generality remains as these content-specific dictionaries are optimized for specific classes of images. These learning-based explicit dictionaries are
20 adaptable but costly and inefficient. The coefficients of the learning-based dictionaries are explicitly computed by using machine learning algorithms which train the dictionary on a set of examples. The explicit dictionaries are tuned much finer than the implicit dictionaries and they perform significantly better. The size of the trained dictionary and the processed signal are limited by the complexity constraints.

As a remedy, an input-adaptive approach can be designed to restore generality while preserving adaptivity. A novel
25 dictionary, \mathbf{G} , with a parametric structure is defined as the product of a fixed non-adaptive base dictionary, $\boldsymbol{\phi}$, and a sparse atom representation matrix, \mathbf{S} , i.e. $\mathbf{G} = \boldsymbol{\phi}\mathbf{S}$. The choice of $\boldsymbol{\phi}$ imposes a structure on the process of dictionary training which acts as regularizer and reduces the overfitting and instability in the presence of noise. The universal base dictionary consists of a fixed set of fundamental signals from which all observable dictionary atoms are formed. The generic matrix $\boldsymbol{\phi}$ can have any number of atoms but they are assumed to span the signal
30 space. The matrix \mathbf{S} provides efficient forward and adjoint operators which bridges the gap between implicit and explicit dictionaries (Horev et al, 2012).

Adaptive methods prefer explicit dictionary representation over the structured ones. In fact, the atoms in \mathbf{G} is a sparse combination of the atoms in $\boldsymbol{\phi}$. In a sense, the dictionary \mathbf{G} can be viewed as an extension to existing dictionaries (matrix $\boldsymbol{\phi}$) adding them a new layer of adaptivity. The adaptive structure (\mathbf{G}) is significantly more
35 efficient than the generic dictionary ($\boldsymbol{\phi}$) depending on the $\boldsymbol{\phi}$ -choice. It is evidently more compact to store and transmit. Parametric dictionaries bridges the gap between complexity and adaptivity for gaining a higher degree of freedom in training but sacrificing regularity and efficiency of the result. As a result, the input-adaptive approach compressions are applicable to a wide range of images (Horev et al, 2012) and it provides a simple, flexible,



adaptive and efficient dictionary representation for its low complexity, compact representation, stability under noise, and reduced overfitting.

The main difficulty of the sparse representation is the problem of dealing with l^0 norm. The basis pursuit (Chen et al., 1999) turns this problem into an l^1 norm which is an optimization problem, i.e. Eq. (1) and Eq. (2) are convexified by replacing l^0 semi-norm with an l^1 norm. Therefore, the goal of the basis pursuit is to find a solution to the sparse representation problem for which the number of non-zero coefficients is minimized. On the other hand, Lasso (Tibshirani, 1996) does not minimize the number of non-zero coefficients but it nulls the coefficients by changing the non-zero coefficients until the number of non-zero coefficients reaches a definite threshold. Lasso is in fact a sparsity-constrained sparse coding problem. Lasso is a modification of the Least Angle Regression and Shrinkage (LARS) algorithm and both are described under the scope of model selection algorithms.

3.1 Model Selection Algorithms

Subset Selection, Forward Selection, Forward Stagewise Linear Regression, Backward Elimination, Lasso, and LARS are different types of model selection algorithms used to select a parsimonious set of covariates among a larger set of them. Subset Selection is a discrete process which produce interpretable models by retaining a subset of predictors and discarding the rest. Shrinkage methods like ridge regression, Lasso, and LARS are more continuous than the subset selection method and are less susceptible to high variance. The followings are selected from (Efron et al., 2004; Hastie et al., 2009).

3.1.1 Ridge Regression

The ridge coefficients are obtained by minimizing the residual sum of squares which are also penalized by imposing a constraint on their size

$$\hat{m}^{ridge} = \underset{m}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (d_i - m_0 - \sum_{j=1}^k g_{ij} m_j)^2 + \lambda \sum_{j=1}^k m_j^2 \right\} \quad (3)$$

where λ is a shrinkage controlling complexity factor and its larger values leads to larger shrinkage. The following is an equivalent way to write the ridge problem and explicitly state the size constraint

$$\hat{m}^{ridge} = \underset{m}{\operatorname{argmin}} \sum_{i=1}^n (d_i - m_0 - \sum_{j=1}^k g_{ij} m_j)^2 \quad \text{subject to} \quad \sum_{j=1}^k m_j^2 \leq t \quad (4)$$

The inputs to the ridge regression should be centered and standardized before solving for the coefficients. The ridge problem can be rewritten in matrix form as

$$RSS(\lambda) = (d - \mathbf{G}m)^T (d - \mathbf{G}m) + \lambda m^T m \quad (5)$$

And the solution will be

$$\hat{m}^{ridge} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T d \quad (6)$$

This formula indicates that the ridge estimates are just the ordinary least squares solution scaled by a factor of $(1 + \lambda)$. Writing the SVD decomposition of the $n \times k$ matrix \mathbf{G} as $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the least squares fitted vector can be written as

$$\mathbf{G}\hat{m}^{ls} = \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T d = \mathbf{G}\mathbf{G}^T d \quad (7)$$

and the ridge solution is equivalently



$$\mathbf{G}\hat{\mathbf{m}}^{ridge} = \mathbf{G}(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T\mathbf{d} = \mathbf{U}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{d} = \sum_{j=1}^k \mathbf{u}_j \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \mathbf{u}_j^T \mathbf{d} \quad (8)$$

where \mathbf{u}_j is the column of matrix \mathbf{U} . Knowing that $\lambda \geq 0$, the $\sigma_j^2/(\sigma_j^2 + \lambda)$ coefficient is obviously less than 1. By comparing the ridge regression to the linear regression we conclude that the coordinates are shrunk by a factor of $\sigma_j^2/(\sigma_j^2 + \lambda)$, which suggests that greater shrinkages are corresponding to smaller σ_j^2 's. Small singular values σ_j correspond to the directions in the column space of \mathbf{G} having small variance.

3.1.2 Lasso

The Lasso is in fact a constrained version of ordinary least squares. In image processing, the Lasso is known as basis pursuit. Lasso is similar to the ridge regression and it is given as

$$\hat{\mathbf{m}}^{lasso} = \underset{\mathbf{m}}{\operatorname{argmin}} \sum_{i=1}^n (d_i - m_0 - \sum_{j=1}^k g_{ij} m_j)^2 \quad \text{subject to} \quad \sum_{j=1}^k |m_j| \leq t \quad (9)$$

Similar to the ridge regression, the inputs can be centered and reparametrized to eliminate the role of m_0 in the formula. The L_2 size constraint $\sum_1^k m_j^2 \leq t$ in ridge regression is replaced by the L_1 size constraint $\sum_1^k |m_j| \leq t$. The Lasso Lagrangian form is given as

$$\hat{\mathbf{m}}^{lasso} = \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (d_i - m_0 - \sum_{j=1}^k g_{ij} m_j)^2 + \lambda \sum_{j=1}^k |m_j| \right\} \quad (10)$$

The parameter t should also be adaptively set alike the λ factor. Each of the subset selection, ridge regression, and Lasso methods apply a simple transformation to the least squares solution. The subset selection estimate retains the M largest coefficients of the least squares solution, $\bar{m}_i \cdot I[\operatorname{rank}(|\bar{m}_i| \leq M)]$, and is known as a sort of hard thresholding. The ridge regression applies a proportional weight of $1/(1 + \lambda)$ as the shrinkage factor to the least squares solution, i.e. $\bar{m}_i/(1 + \lambda)$. The Lasso solution truncates the least squares solution at zero and translates the rest by a factor of λ , i.e. $\operatorname{sign}(\bar{m}_i)(|\bar{m}_i| - \lambda)_+$, which is known to be soft thresholding.

The constraint region for the ridge regression is the disk $m_1^2 + m_2^2 \leq t$, and for the Lasso is the diamond $|m_1| + |m_2| \leq t$. The solution is the first point where the elliptical response contours of the minimization function touches these constraint regions. In the case of Lasso, since the diamond has corners, the corners are specific solutions to the problem where one of the coefficients is zero. In higher dimensions, the constraint region will be a rhomboid and the number of specific (sparse) solutions will be more and more coefficients will equal to zero at those locations.

A generalization to the Lasso and ridge regression solutions is given by

$$\hat{\mathbf{m}} = \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (d_i - m_0 - \sum_{j=1}^k g_{ij} m_j)^2 + \lambda \sum_{j=1}^k |m_j|^q \right\} \quad (11)$$

for $q \geq 0$. Setting $q = 0$, the equation will solve for variable subset selection and $q = 1$ is corresponding to the Lasso. For $q = 2$, the solution estimates the ridge regression coefficients. For $1 < q < 2$, the equation is a compromise between Lasso and ridge regression. It is worthy to note that in the case of $q = 1$, the solution is not differentiable at the corners whereas for $1 < q < 2$, it is differentiable. To take advantage of this property in Lasso (sparse solutions at the corners) elastic net equation is introduced as $\lambda \sum_{j=1}^k (\alpha m_j^2 + (1 - \alpha)|m_j|)$ to represent the



compromise between Lasso and ridge regression while keeping the property of having sharp solutions at the corners.

3.1.3 LARS

5 The least angle regression and shrinkage (LARS) method is introduced by Efron et al. (2004) and it is similar to the Forward Stepwise Regression but less greedy. The Forward Stepwise Regression builds the regression model sequentially and at each step, the best variable is added to the active set until all the active variables are included. LARS is similar to the Forward Stepwise but it does not work with the whole set of active variables. It only works with a proper number of variables.

10 The inputs must first be centered and standardized to zero mean and unit norm. The algorithm starts with empty coefficients $\hat{m}_1, \hat{m}_2, \dots, \hat{m}_k = 0$ and the residual is set to $r = d - \bar{d}$ where $\bar{d} = \sum_{i=1}^n d_i / N$. The algorithm starts with finding the variable most correlated with the response, say \hat{m}_i . The value of \hat{m}_i (which is zero) is continuously dragged towards its least squares coefficient \bar{m}_i ; the amount of residual is continuously decreasing. This continuous dragging is stopped as soon as the correlation of a new variable like \hat{m}_j with the evolving residual catches up with the correlation of the current variable \hat{m}_i . This continuous dragging of values for the two
15 coefficients \hat{m}_i and \hat{m}_j towards their least squares coefficient is continued until the correlation of a third variable like \hat{m}_l with the residual equals the correlation of the joint variables \hat{m}_i and \hat{m}_j with the current residual. After f steps, f variables are non-zero and the rest are still zero. This process is continued until all necessary variables, say k , are included in the model and the full least squares solution is reached at $\min(k, n - 1)$.

20 The geometrical interpretation of the LARS is depicted in figure 1 based on which the name “least angle” is specified. For each variable, the LARS estimate \hat{m} gradually approaches the least squares estimate \bar{m} but does not reach it. This geometry applies to the Lasso too.

In fact, Lasso, and Forward Stagewise Linear Regression are variants of the LARS algorithm. LARS is always faster than the least squares solution. The computational time for LARS is the same as the least squares solution solving for the same number of variables. The LARS and Lasso are similar in term of efficiency but the Lasso
25 takes a bit longer in some cases. LARS, Lasso, and Stagewise Will finally reach the least squares solution but Lasso and Forward Stagewise take more steps to finish such that Lasso stays between the other two.

3.2 ILS-DLA Algorithm

30 PCA (Zou et al., 2006), Generalized PCA (Vidal et al., 2005), MOD (Cheng, 2015; Elad, 2010), K-SVD (Aharon et al., 2006b; Bryt and Elad, 2008; Rubinstein et al., 2013 and 2008; Skretting and Engan, 2011a), ODL (Mairal et al., 2009), ILS-DLA (Engan et al., 2007; Skretting and Engan, 2011b), and RLS-DLA (Skretting and Engan, 2011a and 2010) are among the training algorithms that can be employed to adopt the matrix coefficients to the images in the training set. The idea of iterative least squares dictionary learning algorithm (ILS-DLA) is proposed by Engan et al. (2007). In fact, the ILS-DLA method is similar to the MOD and RLS-DLA methods. The concept, formulation, and the algorithm of the ILS-DLA is the same as the MOD and the difference is only in the dictionary
35 structure and the constraints that are applied on the problem. The ILS-DLA solves for the sparse approximation problem according to the internal structure of the dictionary. The dictionary matrix is blocked into a definite



number of submatrices which are either blocky or overlapping. The difference between the ILS-DLA and the MOD is that in the MOD, the whole dictionary is updated to be used along with the entire corresponding coefficients and the entire training set in a single inversion step; whereas in the ILS-DLA, the dictionary is structured and for each block inside the dictionary, the corresponding data and coefficients are involved in the inversion step to update for the block dictionary. This block dictionary is consistent over the whole dictionary. Therefore, in each inversion step, the process of dictionary update is reduced to updating a very smaller dictionary corresponding to a block. On the other hand, the RLS-DLA is quite different in the sense that it starts with an initial dictionary and it updates the dictionary at each step by introducing a new training vector to the set. In the RLS-DLA the dictionary is updated recursively and no matrix inversion is involved. Therefore, in the RLS-DLA no internal structuring of the dictionary is taking place. In a sense, the ILS-DLA stands in the middle of the MOD and RLS-DLA matrix.

Engan et al. (2007) discuss the ILS-DLA family of methods under three categories: unconstrained block based dictionaries, unconstrained overlapping dictionaries, and constrained.

3.2.1 Unconstrained block based dictionaries

For each subset of data, i.e. signals \mathbf{d}_l , the corresponding weights are given by \mathbf{m}_l and they are related by a dictionary matrix $\mathbf{G} \in \mathbb{R}^{n \times k}$ ($k > n$) defined as $\mathbf{d}_l = \mathbf{G}\mathbf{m}_l$, for $l = 1, 2, \dots, L$. Defining matrix \mathbf{d}_{aug} as the augmentation of the training vectors \mathbf{d}_l and the coefficient matrix \mathbf{m}_{aug} to be the augmentation matrix for of the coefficient vectors \mathbf{m}_l corresponding to \mathbf{d}_l , for the whole data set we can write $\mathbf{d}_{aug} = \mathbf{G}_{aug}\mathbf{m}_{aug}$, as the following

$$\begin{bmatrix} \vdots \\ \mathbf{d}_l \\ \mathbf{d}_{l+1} \\ \mathbf{d}_{l+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & & & & \\ & \mathbf{G} & & & \\ & & \mathbf{G} & & \\ & & & \mathbf{G} & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{m}_l \\ \mathbf{m}_{l+1} \\ \mathbf{m}_{l+2} \\ \vdots \end{bmatrix} \quad (12)$$

Matrix \mathbf{G}_{aug} is the augmentation of matrices \mathbf{G} . The optimization problem for finding the coefficients \mathbf{m}_{aug} , i.e. the sparse coding step, is given as

$$\operatorname{argmin}_{\mathbf{G}, \mathbf{m}_{aug}} \|\mathbf{d}_{aug} - \mathbf{G}_{aug}\mathbf{m}_{aug}\|_F^2 \quad \text{or} \quad \operatorname{argmin}_{\mathbf{G}, \mathbf{m}_{aug}} \|\mathbf{d}_{aug} - \mathbf{G}\mathbf{m}_{aug}\|_F^2 \quad \text{subject to } \|\mathbf{m}_l\|_0 = s_0 \quad \forall l \quad (13)$$

where $\|\cdot\|_F$ indicate the Frobenius norm, $\|\cdot\|_0$ represent the cardinality, and $\|\cdot\|$ express the Euclidean norm. Solving for the coefficients \mathbf{m}_{aug} is an iterative solution and sparse coding techniques like ORMP, LARS, and their variants can be used to obtain the coefficients. The ILS-DLA is developed to solve the optimization problem stated in Eq. (13) which is an error minimization with sparsity constraints or equivalent to Eq. (13) but with additional constraints.

The ILS-DLA algorithm for unrestricted block based dictionaries is in fact reduced to the method of optimal direction (MOD) for which the algorithm can be written as below. Note that the matrix inversion in Eq. (16) is usually affordable (pose no problem) if the dictionary columns are normalized and a sufficiently large training set is used to train the dictionary.



$$\operatorname{argmin}_{\mathbf{G}, \mathbf{m}} \|\mathbf{d} - \mathbf{G}_{aug} \mathbf{m}\|_F^2 \text{ or } \operatorname{argmin}_{\mathbf{G}, \mathbf{m}} \|\mathbf{d} - \mathbf{G} \mathbf{m}\|_F^2 \quad \text{subject to } \|\mathbf{m}_l\|_0 = s_0 \quad (19)$$

and \mathbf{G}_{aug} is structured as in the Eq. (17); and $\mathbf{G} = [\mathbf{G}_1 \dots \mathbf{G}_P]^T$ where $\mathbf{G} \in \mathbb{R}^{NP \times K}$ and $\mathbf{G}_p \in \mathbb{R}^{n \times k}$ for $p = 1, \dots, P$.

For every block and its corresponding weight and signal data, the following setting holds

$$\tilde{\mathbf{d}}_l = \sum_{p=0}^{P-1} \mathbf{G}_{p+1} \mathbf{m}_{l-p} \Rightarrow \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_l \\ \vdots \\ \mathbf{d}_L \end{bmatrix}^T = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \vdots \\ \mathbf{G}_P \end{bmatrix}^T \begin{bmatrix} \mathbf{m}_1 & \dots & \mathbf{m}_l & \dots & \mathbf{m}_L \\ \mathbf{m}_0 & \dots & \mathbf{m}_{l-1} & \dots & \mathbf{m}_{L-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{m}_{1-(P-1)} & \dots & \mathbf{m}_{l-(P-1)} & \dots & \mathbf{m}_{L-(P-1)} \end{bmatrix} \quad (20)$$

- 5 which is a generalization of the system of Eq. (18). In fact, the coefficients with negative indices principally do not exist, i.e. equal to zero. The corresponding system of equations is written for the whole blocks of the submatrices and their corresponding coefficients and signal data. Note that Eq. (18) and Eq. (20) are equivalent (refer to Appendix A to review an example).

According to Eq. (20), we can write $\mathbf{d}^T = \mathbf{G}^T \tilde{\mathbf{m}}$ and for convenience we change notation by writing $\tilde{\mathbf{d}} = \tilde{\mathbf{G}} \tilde{\mathbf{m}}$
 10 where matrix $\tilde{\mathbf{m}}$ is block Toeplitz matrix. Based on the above, the ILS-DLA algorithm for the unconstrained block based matrices can be rewritten as

$$\operatorname{argmin}_{\mathbf{m}_{aug}} \|\tilde{\mathbf{d}} - \tilde{\mathbf{G}}^i \tilde{\mathbf{m}}\|_F^2 \quad \text{subject to } \|\mathbf{m}_l\|_0 = s_0 \quad \forall l \quad (21)$$

for sparse coding step which uses a suboptimal vector selection algorithm or matching pursuit method to obtain \mathbf{m}_l (and thus \mathbf{m}_{aug}^i); and the dictionary update step can be written as

$$15 \operatorname{argmin}_{\mathbf{G}} \|\tilde{\mathbf{d}}_{aug} - \tilde{\mathbf{G}} \tilde{\mathbf{m}}_{aug}^i\|_F^2 \quad (22)$$

to obtain $\tilde{\mathbf{G}}^{(i+1)}$ (and thus $\mathbf{G}_{aug}^{(i+1)}$) using the following inversion step

$$\tilde{\mathbf{G}}^{(i+1)} = \tilde{\mathbf{d}}_{aug} (\tilde{\mathbf{m}}_{aug}^{(i)})^T \left[\tilde{\mathbf{m}}_{aug}^{(i)} (\tilde{\mathbf{m}}_{aug}^{(i)})^T \right]^{-1} \quad (23)$$

Again, the dictionary columns need to be normalized at each step.

As it can be noticed, for each iteration in the ILS-DLA, the process of dictionary update is taking place block by
 20 block by using an inversion step; this is unlike the MOD method in which the entire data set, dictionaries, and coefficients is inverted for in a single inversion step.

3.2.3 Constrained dictionaries

Constrains are usually applied on 2D and 3D signal data to restrict the problem from getting very expensive and
 25 time-consuming. For dictionaries with linear constraints, $\mathbf{A} \mathbf{G} = \mathbf{b}$, the optimization problem of the sparse coding step is written as

$$\operatorname{argmin}_{\mathbf{G}, \mathbf{m}} \|\mathbf{d} - \mathbf{G}_{aug} \mathbf{m}\|_F^2 \quad \text{subject to } \|\mathbf{m}_l\|_0 = T_0 \quad \text{and } \mathbf{A} \mathbf{G} = \mathbf{b} \quad (24)$$



which is similar to Eq. (13) with additional constraint. The general dictionary \mathbf{G}_{aug} is structured as in Eq. (17). Matrix \mathbf{G}_{aug} is an overlapping matrix as in figure 1 and \mathbf{G} is defined $\mathbf{G} = [\mathbf{G}_1 \dots \mathbf{G}_P]^T$. Having these settings, the dictionary updating optimization problem for step 2 becomes

$$\underset{\mathbf{G}}{\operatorname{argmin}} \|\mathbf{d}_{aug} - \mathbf{G}\mathbf{m}_{aug}^{(i)}\|_F^2 \quad \text{subject to } \mathbf{A}\mathbf{G} = \mathbf{b} \quad (25)$$

5 which follows the same setting as in the case of unconstrained overlapping dictionaries. Looking into the design of Eq. (20) where $\mathbf{d}^T = \mathbf{G}^T \tilde{\mathbf{m}}$, one could equivalently write $(\mathbf{d}^T)^T = (\mathbf{G}^T \tilde{\mathbf{m}})^T$ and have $\mathbf{d} = \tilde{\mathbf{m}}^T \mathbf{G}$; therefore, another setting for Eq. (25) is to write

$$\underset{\mathbf{G}}{\operatorname{argmin}} \|\mathbf{d} - \mathbf{m}_{aug}^{(i)} \mathbf{G}\|_F^2 \quad \text{subject to } \mathbf{A}\mathbf{G} = \mathbf{b} \quad (26)$$

The above setting is generated by the augmentation over the coefficients, instead of the dictionaries and writing
 10 \mathbf{m}_{aug} as a block diagonal Toeplitz matrix having

$$\mathbf{m}_{aug} = \begin{bmatrix} \tilde{\mathbf{m}}_{aug}^T & & & \\ & \tilde{\mathbf{m}}_{aug}^T & & \\ & & \ddots & \\ & & & \tilde{\mathbf{m}}_{aug}^T \end{bmatrix} \quad (27)$$

where $\mathbf{m}_{aug}^{(i)} \in \mathbb{R}^{nL \times nkP}$.

Engan et al. (2007) solve the Eq. (26) by using the Lagrange polynomial; they give the final answer to be

$$\mathbf{G} = \left(\mathbf{m}_{aug}^{(i)T} \mathbf{m}_{aug}^{(i)} \right)^{-1} \mathbf{m}_{aug}^{(i)T} \mathbf{d} - \frac{1}{2} \left(\mathbf{m}_{aug}^{(i)T} \mathbf{m}_{aug}^{(i)} \right)^{-1} \mathbf{A}^T \lambda^T \quad (28)$$

$$15 \quad \frac{1}{2} \lambda^T = \left[\mathbf{A} \left(\mathbf{m}_{aug}^{(i)T} \mathbf{m}_{aug}^{(i)} \right)^{-1} \mathbf{A}^T \right]^{-1} \mathbf{A} \left(\mathbf{m}_{aug}^{(i)T} \mathbf{m}_{aug}^{(i)} \right)^{-1} \mathbf{m}_{aug}^{(i)T} \mathbf{d} - \mathbf{b} \quad (29)$$

where λ is the Lagrange multiplier.

Another case discussed by Engan et al. (2007) is when a large overlapping dictionary, \mathbf{G}_{aug} with $\mathbf{G} \in \mathbb{R}^{n^P \times k}$ is defined as the multiplication of a smaller overlapping dictionary, \mathbf{X}_{aug} with $\mathbf{X} \in \mathbb{R}^{n^P \times n}$ (figure 1a, Eq. (17)), and a block based dictionary, \mathbf{Y}_{aug} with $\mathbf{Y} \in \mathbb{R}^{n \times k}$ (figure 1b, Eq. (12)), given by $\mathbf{G}_{aug} = \mathbf{X}_{aug} \mathbf{Y}_{aug}$. The
 20 approximated signals $\tilde{\mathbf{d}}$ are represented as $\tilde{\mathbf{d}} = \mathbf{G}_{aug} \mathbf{m} = \mathbf{X}_{aug} \mathbf{Y}_{aug} \mathbf{m}$ ($\mathbf{G}_{aug} \in \mathbb{R}^{nL \times kL}$, $\mathbf{X}_{aug} \in \mathbb{R}^{nL \times nL}$, and $\mathbf{Y}_{aug} \in \mathbb{R}^{nL \times kL}$).

Consequently, the optimization problem of Eq. (19), i.e. the sparse coding step, changes into

$$\underset{\mathbf{G}, \mathbf{m}}{\operatorname{argmin}} \|\mathbf{d} - \mathbf{G}_{aug} \mathbf{m}\|_F^2 = \underset{\mathbf{X}, \mathbf{Y}, \mathbf{m}}{\operatorname{argmin}} \|\mathbf{d} - \mathbf{X}_{aug} \mathbf{Y}_{aug} \mathbf{m}\|_F^2 \quad \text{subject to } \mathbf{X}_{aug}, \mathbf{Y}_{aug}, \text{ and } \|\mathbf{m}_l\|_0 = s_0 \quad (30)$$

By defining a suitable \mathbf{X}_{aug} and fixing it through the iteration process, a suitable \mathbf{Y}_{aug} can be found to minimize
 25 the optimization problem. Having $\mathbf{m}^{(i)}$ coefficients obtained from the sparse coding step, the optimization problem for dictionary updating can be written as

$$\underset{\mathbf{Y}_{aug}}{\operatorname{argmin}} \|\mathbf{d} - \mathbf{X}_{aug} \mathbf{Y}_{aug} \mathbf{m}^{(i)}\|_F^2 \quad \text{subject to } \mathbf{Y}_{aug} \quad (31)$$



This optimization problem is solved as for the unconstrained block based dictionaries. Note that since \mathbf{X}_{aug} is known, we can weigh the signal vectors, \mathbf{d} , by \mathbf{X}_{aug}^{-1} and rewrite Eq. (31) as

$$\underset{Y_{aug}}{\operatorname{argmin}} \|\mathbf{d} - \mathbf{X}_{aug} \mathbf{Y}_{aug} \mathbf{m}^{(i)}\| = \underset{Y_{aug}}{\operatorname{argmin}} \|\mathbf{X}_{aug}^{-1} \mathbf{d} - \mathbf{Y}_{aug} \mathbf{m}^{(i)}\| \quad \text{subject to } Y_{aug} \quad (32)$$

Therefore, instead of \mathbf{m} , the weighted vector/matrix $\mathbf{X}_{aug}^{-1} \mathbf{d}$ will be the input to the minimization problem.

- 5 As it can be seen, in functionality, the idea of defining the dictionary matrix as a multiplication of an overlapping and a block based matrix resembles the idea of the input-adaptive approach where we defined the dictionary as a multiplication of a fixed non-adaptive base dictionary, $\boldsymbol{\phi}$, with an adaptive learning dictionary, \mathbf{S} , defining $\mathbf{G} = \boldsymbol{\phi} \mathbf{S}$.

- The ILS-DLA algorithm can be used in an image compression scheme to produce sparsely represented images.
 10 The process of image compression/decompression consists of two stages of an offline ILS-DLA training algorithm and an online image encoding/decoding process. In this scheme, the image is first partitioned into smaller patches and the mean DC value is subtracted from each patch. The non-zero elements of the patches are then quantized by a uniform quantizer and entropy (and also AC and Huffman) coded. The resultant patches are used to train the dictionary by the ILS-DLA algorithm. The discrete cosine transform (DCT) is used as the fixed dictionary matrix.
 15 Other alternatives include LOT and ELT methods. The final trained dictionary would be a matrix like \mathbf{S} which is a dictionary specifically trained for the images in the training set by the offline ILS-DLA algorithm. The coefficients of matrix \mathbf{S} are then quantized and reshaped. Multiplication of the fixed implicit dictionary (i.e. DCT, $\boldsymbol{\phi}$) and the sparse explicit dictionary (\mathbf{S}) will result into an input-adaptive matrix \mathbf{G} ($\mathbf{G} = \boldsymbol{\phi} \mathbf{S}$). The order recursive matching pursuit (ORMP) is used for sparse coding.

20 4 Results

- Results of this study are multifold. The 3D cube seismic data is deterministically inverted and AI, porosity, and saturation cubes are obtained. The spectral decomposition cube was generated and studied. Two reality-based permeability models were selected from the spectral decomposition cube based on which, many MPS realizations were generated using different MPS methodologies. Different dictionary learning algorithms were used to train
 25 content-specific dictionaries based on the training set of generated MPS realizations. Different sparse coding methods were utilized to reconstruct the final sparsity-based compressed image model. Statistics are given to indicate the goodness of the sparsity-based compressed reconstructed image model comparing to the training set of MPS generated realizations. Physical structure resemblance to the true models and fluid flow characteristics are considered as criteria to evaluate the sparsity-based compressed reconstructed image model.
 30 The AI profile (figure 3b) contains very important fluid and rock properties information about the reservoir. On the AI profile, the position for the set of fractures are indicated by the red colors which possess low acoustic impedance and suggest that the fracture sets are carrying the injected gas along the structure. Between these fracture sets, which are indicated by red colors, there is a band of blue color along the structure. This band of high acoustic impedance can be correlated with the non-fractured zone at the crest of the anticline. On the left of the AI profile,
 35 the deltaic depositional system is detected to diverge through the center of the field.



The porosity profile corresponding to the AI profile indicates that the difference in pressure behavior of the western part from the center and eastern parts are due to porosity change. Looking back into the AI profile approves that the porosity change from west to the center of the reservoir is due to the change in the depositional system. This change of depositional systems causes a drop of porosity by a value of almost 5% from a mean value of 11-12% typical for carbonates to a mean value of 16-17% typical for consolidated fine sandstone. Further study of the porosity cube approves a double-layered reservoir which is compatible with reservoir engineering data. Based on the porosity cube, the petrophysical setting of the reservoir is indicated to be a layering of two porous intervals interbedded within three upper, middle, and lower intact layers.

Studying the saturation profile, further approves that the reservoir is double layered and water-edged. This finding is also compatible with reservoir pressure and production data. The inversion results have been recently approved by drilling a well into the lower layer and producing petroleum. Another very important feature detected in the inverted saturation cube is the feature corresponding to the gas fingering effect (figure 4) which can also be highly correlated with the features observed on the acoustic impedance section. The two tongues of the injected gas can be correlated with the positions of the conjugate fracture sets distributed along the structure. Furthermore, the non-gaseous zone in the middle of the two tongues corresponds to the non-fractured part of the crest which is also observed on the AI profile (figure 3b).

The spectral decomposition profile (figure 3a) represents a descent depiction of the delta system on the western part of the reservoir. It also illustrates the conjugate fracture systems which are spread in parallel along the anticline on both sides of the anticline crest. As it is evident in the figure, the area at the crest is non-fractured. The interpreted features on the spectral decomposition section is compatible with those on the AI section (figure 3b) and the injected gas fingering (figure 4).

The above interpretations are provided to emphasize that the selected models utilized as basis for training images must represent the reality of the field and must be highly validated by any types of available data. Accordingly, a porosity section (figure 5a) is selected from the porosity profile extracted from the seismic cube. Furthermore, two permeability models are selected based on the features observed in the spectral decomposition profile (figures 5b and 5d).

Deterministic reservoir characterization methods will result in a single most likely model. The model obtained by deterministic processes might not be the best possible model. Stochastic processes, e.g. stochastic inversion, are used to generate a limited set of equally probable models. The stochasticity and diversity of models obtained by methods like stochastic inversions are not usually enough to manipulate and generate a large set of models which can adequately represent almost every possible models of the desired property. The best remedy to this problem is to use multiple-point statistics methods and generate a large set of realizations based on realistic training images. In this paper the attempt is to use MPS methods to generate large sets of realizations and to show these training sets of realizations can be used to obtain a single sparsity-based compressed image model which is similar to the true image model in physical structure and outperforms the majority of the population sample regarding the fluid flow characteristics criteria. Mentioning the condition of physical similarity to the true image is very important because the physical structure of the MPS realizations are diverse and they might not be even similar to the true image model.



Accordingly, the MPS methods are used to generate a large number of stochastic models. A few number of DisPat-generated realizations are randomly selected from the set of 3000 population sample and are depicted in figures 6 and 7 for based on fracture system (figure 5b) and delta system training images (figure 5d), respectively. The generated stochastic MPS realizations are equally probable. In order to introduce most stochasticity within the models, there is no data conditioning applied on the process of generating the MPS realizations.

These sets of DisPat-generated realization models are used as training sets to learn the dictionary by ILS-DLA method with LARS. Each image will contribute to the sparse approximation process represented in Eq. (1) as a column vector in m . The columns of the trained dictionary are called atoms and will form the basis to sparsely represent signals and images. Figures 8a and 8b illustrate a number of 56 atoms of the dictionaries trained over the set of stationary fracture system and non-stationary delta system MPS realizations, respectively.

The sparsity-based compressed image model reconstructed by training the dictionary using ILS-DLA method (Engan et al., 2007; Skretting and Engan, 2011b) with LARS (Efron et al., 2004; Hastie et al., 2009) over the training set of stationary fracture system realizations generated by the DisPat MPS algorithm (Honarkhah, 2011) is illustrated in figure 9a. Figure 9c depicts the difference between the reconstructed and the original image. As it can be seen in the difference picture (figure 9c), the difference is random and it does not show any special structure which is indicative for good quality of the reconstructed image. Furthermore, the maximum and minimum difference values are not high and general difference value remains around zero (note the background color). Same interpretations can be given for figures 9b and 9d corresponding to the nonstationary delta system.

Rather than the overall physical structure of the reconstructed images, the fluid flow properties of the reconstructed images are also considered as further criteria to check for the goodness of reconstructed image. The MRST MATLAB code (Lie, 2015) is used to calculate the production and pressure change at the observation well by time and sketch their profile. The production and pressure profiles for the sparsity-based compressed fractured system image model are sketched against the production and pressure profiles of the original model. Production of the injected fluid is increasing by time at the observation well and its pressure is declining. The production profiles for the reconstructed stationary fracture system model (figure 10a) and the reconstructed nonstationary delta system model (figure 10b) show almost perfect match to the production profile of the original images. The pressure profiles of the reconstructed images are also sketched against the original images (figures 10c and 10d). For both production and pressure profiles, the errors are calculated and used as criteria for goodness of fit, i.e. history-matching.

The workflow explained in this paper is performed 48 times for different settings of experiments. For each experiment three different parts are considered, i.e. the generation of MPS realizations, dictionary training, and the sparse coding. In terms of methodology, the experiments follow a one-at-a-time design which means that each experiment is different from other experiments by the utilized methodology for one of the three parts. For two types of stationary and nonstationary training images (figures 5b and 5d), a total number of 48 experiments were set up based on a one-at-a-time design in the methodologies.

In specific, six training sets were used for the experiments among which four training sets are generated by the DisPat, FilterSim, and SNESim MPS methods based on the fracture system training image (figure 5b). The two other sets are generated by DisPat and FilterSim MPS methods based on the delta system training image (figure



5d). Each training set contains 3000 generated realizations and for every single realization, the production and pressure errors are calculated against the original models and these errors have been used as criteria for goodness of the model.

5 Rather than using different training images (i.e. the stationary fracture system and the nonstationary delta system) and utilizing different MPS methods to generate the realizations, a number of eight different sparse approximation methods were used to generate the final sparsity-based compressive reconstructed image model. These eight different methods are considered by combining different dictionary learning algorithms with different sparse coding methods.

10 The results of the total number of 48 experiments are plotted in figure 11. Each point corresponds to one experiment. The horizontal axis indicates the production superiority of the resultant sparsity-based compressed model reconstructed based on the specific setting of the problem in that experiment. Likewise, the vertical axis indicates the pressure superiority. The superiority index for each axis is calculated based on the error of fit for the reconstructed model against the true model (i.e. the original images, figures 5b and 5d) in a population sample of 3000 realizations used as training set in that specific experiment. For example, if the production profile of the reconstructed sparsity-based compressed image model in one experiment shows 98% of superiority, it means that
15 the error of the production profile of that reconstructed image is less than the error of production profile of 98% of the realizations in that training set. Note that the errors are calculated against the production profile of the true image which is used as basis to generate the population sample (i.e. the 3000 realizations used as training set in that specific experiment).

20 Considering the total 48 experiments, the results are shown to be very encouraging. It is very interesting to note that out of 48 performed experiments, 54.17% of the total sparsity-based compressed models fall in the area of 95%-95% superiority to the population samples (note that the first 95% corresponds to the probability in production superiority and the second one corresponds to the probability in pressure superiority or vice versa), 89.58% of the sparsity-based compressed models fall in the area of 90%-90%, and 95.83% fall in the area of 85%-85%.

25 Specifically considering the setting of experiment discussed in this paper, the results indicate superiority of the sparsity-based compressed model image over 97.08% of the total MPS realizations generated based on the stationary fracture image. The same comparison for the nonstationary image of the deltaic system indicate a superiority over 94.41% of the total number of MPS realizations.

5 Discussions

30 We have practiced the sparsity-based compressive reservoir characterization and modeling algorithm which is able to manipulate the model images and reveal a close translation of the true model. The resultant sparsity-based compressed model is neither the true model nor it is guaranteed that it is the best possible model among a large data set. What it can be claimed is that our setting of experiments, applied on our set of MPS realizations generated based on training images (shown in figures 5a and 5d) and using different sparse approximation methods, illustrate
35 that the resultant sparsity-based compressed model image outperform 90% of the MPS generated realizations by 89.58% probability and 85% of the MPS realizations by 95.83% probability.



The sparsity-based compressive reservoir characterization and modeling workflow integrates the multiple-point statistics modeling from the geostatistics with the dictionary learning, sparse approximation, and image compression from the image processing and works on the input data to result in sparsity-based compressed model images. The quality controlling is performed by passing the whole MPS realizations along with the sparsity-based compressed resultant models through the reservoir simulation and the pressure drop and production profiles are used as criteria for goodness of models. The input data to this workflow is the seismic inverted 3D cubes (Acoustic Impedance, porosity, and water saturation), seismic spectral decomposition 3D cube, seismic interpretation data, production data, and the well logs. This workflow is applied on semi-industrial model images in the gas injection area of an Iranian Asmari reservoir due to the importance of the subject and to illustrate the effectiveness of the sparsity-based compressive workflow in a complex reservoir.

5.1 The case study reservoir, complexities, and uncertainties

The case study for this research is an Iranian oilfield located southwest of Iran. The Asmari reservoir is the main producing layer in this oilfield and two types of matrix and fracture porosities are considered in the simulation models for this reservoir. It is believed that the reservoir is communicating along the crest of the reservoir through the open fracture networks.

At the time of this study, 20 wells were drilled across the reservoir. Suitable distribution of the wells has provided an appropriate bank of information gathered from the entire reservoir. The gas is injecting into the reservoir from well No. 8 since 2005 and from well No. 3 since 2010 for pressure maintenance. The acquisition of the single component 3D seismic data was conducted during the years of 2009 and 2010. The results from deterministic inversion of seismic data indicate early breakthrough of the injected gas observed in two wells on both sides of the injection well No. 8 (figure 4) which is also proven by production data (i.e. GOR and pressure). Two scenarios are considered for this evidence: existence of high porosity high permeability sandstone layers, and longitudinal fracture systems at the crest that provide pressure and fluid flow communication across the reservoir. Results from spectral decomposition (figure 3a) and acoustic impedance inversion (figure 3b) are in favor of this latter assumption and in order to verify it at an industrial scale, reliable fracture and lithological models are required.

The pressure behavior of the western part of the reservoir is different from the central and eastern parts. The spectral decomposition of 3D seismic data establishes a deltaic depositional system for the western part of the reservoir and it also evidences that the genesis of the reservoir rock for the western part is different from the rest of the reservoir. Accordingly, the discussed Asmari reservoir is a complicated reservoir which is deltaic in west and it is fractured carbonates with interbeds of sandstone elsewhere (figure 3a and 3b). The most probable scenario explaining the early breakthrough of gas in the observation wells is to consider the extension of the conjugate fractures at the crest from center to the east.

In figure 3b, the accumulation of the red color (low AI) is where the fracture networks are located. As it is illustrated, two parallel fracture networks are identified and in the middle, there is a non-fractured zone. A fault is joining into the northern fracture set from the northwest of the reservoir. Looking closer to the low AI feature, it is clearly noticed that low AI features are linear and parallel but not continuous; these features are mostly recognizable as fractures. Whereas in the west, branches of the deltaic system diverging from west to east (left quarter of the picture) are quite recognizable.



5.2 Uncertainty and Stochasticity

As the cumulative production of the Iranian oil reservoirs increase, switching to secondary and tertiary oil recovery methods proves a necessity. Few of the southern oilfields are already under secondary recovery methods among which gas injection is the most popular. The success of pressure maintenance scenarios for a reservoir, e.g. gas injection, relies on our knowledge about the reservoir properties. These properties are either dynamic or static. Seismic data interpretation qualitatively characterize the reservoir, for example the reservoir structure and the existence and form of the faults. The 3D cube seismic data for the reservoir under study was interpreted using a commercial software and the well log and vertical seismic profiling data were utilized as the inputs to perform such a task. No effective fault was interpreted hitting the WOC boundary of the reservoir. Thorough interpretation of the faults is important especially when the case of gas injection is considered. The injected gas can find its way out of the reservoir structure through the faults that reach the surface which is the case of an adjacent reservoir. Along with qualitative properties, quantitative properties also exist, e.g. porosity and permeability. Having seismic attributes as variables, the porosity distribution in the reservoir can be obtained by linear or nonlinear transforms and constraining to porosity data at well locations. The 3D cube porosity distribution was obtained using a multi-attribute transform.

Finding the permeability distribution in the reservoir is always more difficult and more uncertain because unlike porosity, no conventional well log measures the permeability in a well interval. Consequently, the permeability is an estimation of the porosity which is itself uncertain. On the other hand, permeability does not always have a direct relation with the porosity; for example, in carbonate reservoirs, permeability of a layer is directly related to the number and direction of open fractures and it does not necessarily follow the porosity trend. Furthermore, matrix porosity is a syndepositional property related to the rock genesis and usually changes gradually across the reservoir, whereas tectonic-related fractures are usually post-depositional and the changes can be abrupt and unpredictable. A thin bed of shale can completely separate a layer from the reservoir. Unlike porosity, there is another constraint on quantifying the permeability, i.e. the well test data. The well test data record the amount of production for different reservoir intervals due to an implied pressure drop and a permeability value is specified for the tested interval according to the Darcy equation. In the commercial simulation model, both matrix and fracture porosities are included, however, in our semi-industrial simulation model the seismic driven porosity distribution was considered (figure 5a) which can be interpreted as the consequent of both types of porosities.

In order to simulate fluid flow in the reservoir rock, hydraulic units are considered to transform the fluid through the reservoir rock. To determine the number of hydraulic units and specify their properties, microscopic core studies and some well log data are needed. These rock properties are considered to be constant under a specific rock type and hence, the hydraulic units are different from each other because their properties differ. Therefore, rock type definition and their lateral and vertical distribution in the reservoir is another crucial step in reservoir simulation. In the presence of a high porosity high permeability sand layer, the injected gas will break early through this specific hydraulic unit and the gas injection objectives are not well provided. Same scenario will take place in case of an open fracture network providing a convenient pressure communication and fluid flow across the reservoir. 3C seismic data are the least necessity for recognizing lithology and fracture network in the reservoir extent using seismic data. The 3D seismic data of the studied well is single component. Core studies and image logs interpretation supply the main data input to building a reliable fracture network. Very few image logs and



core sections were available from the reservoir under study and therefore the information to build a proper fracture network was poor.

The uncertainty is intrinsically related to measurements and reservoir modeling is no exception. To study uncertainty in reservoir modeling, determining the effective parameters and providing the statistical information about these parameters are the prerequisites (Caers, 2011). In order to include, quantify, and report the uncertainty, suitable modeling methods should be selected and conducted. As the main source of information in the extent of the reservoir, seismic data should be inverted stochastically and based on reliable geostatistical information from the reservoir (Castro, 2007; Wu, 2007). For a reservoir, different models with similar or dissimilar uncertainties can give the same response and hence, more constraining information are required. 4D seismic data provide spatial constraints mostly on the fluid content and transport in the reservoir, which is missing for the studied reservoir. This type of data is especially practical for studying the motion of the gas injection front and its adherence to the high structure, the fracture network directions, and the high porosity high permeability layers.

5.3 MPS realizations

A fracture and lithological model based only on the well data is quite uncertain as the wells are sparsely distributed in the reservoir. Geostatistics is the technique to spread the well properties from well locations to the extent of the reservoir. Two-point geostatistics are based on linear changes between two points in the space (variogram). Two-point geostatistical methods do not generate satisfactory models for complex reservoirs. Multiple-point statistics methods are able to generate more realistic models because it accounts for the changes from a point to different directions simultaneously. Among the current MPS methods, the pattern-based ones are known to have the best performance. SimPat (Arpat, 2005), FilterSim (Zhang, 2006), Direct Sampling (Mariethoz and Renard, 2010), DisPat (Honarkhah, 2011), and WaveSim (Chatterjee et al., 2012) are the well-known pattern-based MPS methods. These algorithms stochastically reproduce the extracted patterns from the training images which are provided based on prior information about the reservoir. The location-independency of the training images along with the stochasticity of the pattern-based MPS algorithm will result in generating different models which are equally probable. The advances made in the DisPat algorithm has facilitated modeling a stationary fracture network (figure 5b) or nonstationary deltaic system (figure 5d) based on training images and soft and hard data integration.

There are further expectations from the MPS in contribution with the field of reservoir engineering and the reservoir simulation in specific. As a specific case, new achievements in the nonstationary image modeling through segmenting the training images into stationary subregions could be correlated with sectoring the reservoir extent based on pressure and production data and eventually providing a better reservoir history-matching. In our example, the reservoir depositional system is a combination of a deltaic and a carbonate ramp system. It is expected that the new MPS methods like DisPat can model such a complex depositional system reliably and the pressure and production data meet this differentiation of the depositional systems.

For our set of experiments, two training images were selected from the 3D spectral decomposition seismic cube representing the fracture network (figure 5b) and the delta system (figure 5d). The fracture network was considered stationary whereas the delta system was assumed nonstationary. The two images were used as input to the DisPat MPS algorithm and a large set of stochastic realizations were generated for each case. The MPS simulation was not conditioned by any hard or soft data to introduce more stochasticity into the realization models. Therefore, the



MPS realizations which are utilized as training sets for dictionary training in the sparse approximation procedures, are generated based on the training images under pure stochasticity with no conditioning.

5.4 Applying sparse approximation on the MPS realizations

5 An image is an assembly of information and data redundancy. Information and data are not the same concept; in fact, information is conveyed by the data. Different amounts of data might transfer the same amount of information. Some part of data might contain no related information or repeat the same scope of information. This concept is referred to as data redundancy. Coding, interpixel, and psychovisual are the three sources of data redundancy. Image compression methods are designed to eliminate the data redundancy and they are either lossy or lossless (Gonzalez and Woods, 2002). Reducing the unnecessary details of the model images generated by the MPS
10 methods as inputs to the reservoir simulation preserves the quality of the results while reducing the computational time. Compressing a model without losing the fundamental information but reducing the dispensable details in an almost lossless manner (Gonzalez and Woods, 2002) will accelerate reservoir simulation especially for complex systems. Reservoir model upscaling can also be substituted by lossy compression (Gonzalez and Woods, 2002) if the amount of data loss is observed and controlled.

15 As it has been noted, signals can be sparsely represented or approximated by the dictionaries. These dictionaries can be either fixed and predetermined, e.g. DCT and wavelet transforms, or be trained by a training set using dictionary training methods like MOD (Cheng, 2015; Elad, 2010), K-SVD (Aharon et al., 2006b; Bryt and Elad, 2008; Rubinstein et al., 2013 and 2008; Skretting and Engan, 2011a), ODL (Mairal et al., 2009), ILS-DLA (Engan et al., 2007; Skretting and Engan, 2011b), and RLS-DLA (Skretting and Engan, 2011a and 2010). The trained
20 dictionaries can learn from the MPS realizations as the input training set (figures 6 and 7) to the sparsity-based compression scheme to generate a single sparsity-based compressed model image which is in turn an input to the reservoir simulation (figure 9a and 9b).

Reviewing the workflow applied on the case study, the spectral decomposition cube (figure 3a) was used as the source to extract a basis for the fracture network stationary training image and a basis for the deltaic system
25 nonstationary training image (figures 5b and 5d). The DisPat algorithm was utilized to create a large set of MPS realizations for each training image (figures 6 and 7). The dictionaries were trained on the generated MPS realizations using the ILS-DLA algorithm and the trained dictionaries (figures 8a and 8b) were used with the LARS method in a sparsity-based image compression scheme to generate a single sparsity-based compressed model image. Each realization along with the two resultant sparsity-based compressed model images (figures 9a and 9b)
30 were used as input models to the reservoir simulation and the production and pressure history of each model were compared with the main true model images for comparison (figure 10) (Lie, 2015).

5.5 The sparsity-based compressed model

The sparsity-based compressive reservoir characterization and modeling workflow is applied on several sets of stationary and nonstationary generated DisPat, FilterSim, and SNESim MPS realizations. The sparsity-based
35 compression scheme was applied on these training sets using different available dictionary training methods. The statistical information from simulation models for each compressed model image is illustrated in figure 11. For each model image, two criteria are considered for comparison, the production and pressure. The horizontal axis is



the percentage of superiority of the sparsity-based compressed model image in terms of production and compared to the model images in the population sample. The vertical axis indicate the same but in terms of pressure. It is encouraging to note that out of 48 performed experiments, 54.17% of the total sparsity-based compressed models fall in the area of 95%-95% (upper 5%) superiority to the population samples, 89.58% in the area of 90%-90% (upper 10%), and 95.83% in the area of 85%-85% (upper 15%). It is even more encouraging to note that each single point in figure 11 is referred to different methods of sparsity-based image compression. In setting up each experiment, three different methodologies are involved, i.e. MPS methods, dictionary learning algorithms, and sparse coding methods. In each part, there are a few parameters that can be set and optimize to achieve the best resultant sparsity-based compressed image model. The parameter optimization is not performed for any of the 48 number of experiments and therefore, results improvements are quite expecting in these regards too. It is highly expected that more encouraging results would be achieved if same sparse approximation method is repeatedly applied on MPS realizations generated from various training images. Another important point to note is that, the generated MPS realizations are quite stochastic, that is, no hard or soft data constraint is applied on the images and accordingly, the authors expect even more encouraging results if any constraint is applied on the MPS algorithms to generate the realizations.

When the criteria of production and pressure history-match is considered, the results of the current experiment, i.e. applying the ILS-DLA dictionary training algorithm with LARS sparse coding method on DisPat-generated MPS realizations, for the stationary fracture image indicate the superiority of the sparsity-based compressed model image over 97.08% of the total MPS realizations. This means that for a total number of 3000 MPS realizations, only 88 realizations result in a better history-match. The same comparison for the nonstationary image of the deltaic system indicate a superiority over 94.41% of the total number of MPS realizations, i.e. 167 MPS realizations out of 3000 are superior in history-match.

It would be interesting to check for the superiority of the resultant sparsity-based compressed model if a set of bad model images are used as training set to the sparse approximation algorithm. Basically, it is interesting to check if there exists at least one model in the population sample which outperforms the sparsity-based compressed model image in case of using a set of bad model images. In other words, the task would be to check if it is possible that the sparsity-based compressed model image outperforms the whole population sample. If not so, how can we invent new algorithms in order to achieve such model? This issue is even more interesting when noting that the process of vector selection in the sparse approximation scheme is a random process which means that the chance of selecting a good model/vector for sparse approximation is limited. Therefore, suitable vectors or models, i.e. the MPS realizations for which the pressure and production criteria are better than the sparsity-based compressed model and form a low percentage of the population sample, have limited chance to be selected by the sparse coding step to form a base for reconstructing the sparsity-based compressed model. In a sense, we can loosely conclude that the goodness of a sparsity-based compressed model is mostly relying on the inferior MPS realization models, which form the majority, rather than relying on the superior MPS realization models, that form a low fraction minority. Further experiments are needed to verify such claim which are subjects of future studies.



6 Conclusions

The sparsity-based compressed reservoir characterization and modeling workflow is introduced, presented, and applied on semi-industrial scale models from a gas injection case Iranian Asmari reservoir. This workflow is set to achieve a close translation of an unknown true model from a set of models which are assumed as alterations of the true model. The main blocks of this workflow is the multiple-point statistics modeling and the sparse approximation algorithms. The former algorithm generates the set of alterations of true model which are used by the latter algorithm to produce the final sparsity-based compressed model images.

Extracting the properties of porosity and saturation from the deterministically inverted seismic data of the oilfield under study, along with discovering a new potential hydrocarbon layer interval (which was later confirmed by a drilled well) and defining its extents, obviated the structural complexity of this reservoir. This study approved the presence of sandstone interbeds in the Asmari carbonate reservoir and found that the overall fall in porosity is responsible for different behavior of the pressure drop in the western part of the reservoir. The saturation distribution based on deterministic inversion of the seismic data was able to detect the front of the injected gas. Findings are also consistent with pressure data and the oil components in the observation wells.

The geologic complexities in the Asmari reservoir of this oilfield, noting as the presence of sandstone interbeds in the carbonate reservoir, the existence of a connected and communicating fracture network at and along the crest, and the deltaic system of the western part of the reservoir, makes the fluid flow simulation history-matching of this reservoir a difficult task. Some of these problems refer to the incomplete studies and lack of information but some others are related to the intrinsic incapability of the current algorithms in building realistic static reservoir models. Accordingly, the DisPat algorithm was used to produce a set of realizations for the two training images of the fracture network and the deltaic system because of its capabilities in hard and soft data integration at different resolution levels and its new nonstationary image modeling techniques.

The DisPat MPS algorithm was used to generate a set of stochastic realizations based on the fracture network stationary training image. A similar set of realizations was generated for the deltaic system nonstationary training image. A 2D injection model was tested on both images and the simulation results were set as the basis for comparison. Same scenario was run for each individual MPS realizations of both sets of stationary and nonstationary images and the population sample for the history-matching simulation results were formed. The ILS-DLA trained dictionaries were used in a sparsity-based image compression scheme and the resultant compressed images for both stationary and nonstationary sets were run into the reservoir simulation process for history-matching. Based on the stationary fracture network MPS realizations population samples, the resultant sparsity-based compressed model image is superior to 97.08% of the MPS realizations; for the nonstationary deltaic system MPS realizations population samples this number is 94.41%.

Reservoir engineers are proposing the existence of high porosity high permeability sandstone interbeds for witnessing the fingering feature of the injected gas. Whereas, studying the acoustic impedance and spectral decomposition 3D seismic cubes along with noting the fact of having a very few number of image logs recorded in the field, the overall conclusion would be that a communicating open fracture network along the crest is more probable to be accounted for the observed gas fingering feature.



- Bryt O. and Elad M., 2008. Compression of facial images using the K-SVD algorithm. *Journal of Visual communication and Image Representation*, 19(4):270–283.
- Caers J., 2011. *Modeling Uncertainty in the Earth Sciences*, John Wiley & Sons, Ltd, Chichester, UK.
- Castro S., 2007. A probabilistic approach to jointly integrate 3D/4D seismic, production data and geological information for building reservoir models. Ph.D. thesis, Stanford University, Stanford, CA.
- Chatterjee S., Dimitrakopoulos R., Mustapha H., 2012. Dimensional reduction of pattern-based simulation using wavelet analysis. *Math Geosci* (2012) 44:343–374.
- Cheng H., 2015. *Sparse Representation, Modeling and Learning in Visual Recognition*. Springer, London
- Chen S. S., Donoho D. L., and Saunders M. A., 1999. Atomic decomposition by basis pursuit. *SIAM J.Sci. Comp.*
- 10 Deutsch C. V., Journel A. G., 1998. *GSLIB: Geostatistical Software Library and Users Guide*, second edition Edition. Oxford, New York.
- Efron B., Hastie T., Johnstone I. and Tibshirani R., 2004. Least angle regression (with discussion), *Annals of Statistics* 32(2): 407–499.
- Elad M., 2010. *Sparse and Redundant Representations, from Theory to Applications in Signal and Image Processing*. Springer, New York, USA.
- 15 Elad M. and Aharon M., 2006. Image denoising via learned dictionaries and sparse representation, *International Conference on Computer Vision and pattern Recognition*, New-York, June 17-22.
- Engan K., Skretting K. and Husy J. H., 2007. Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation. *Digital Signal Process.*, vol. 17, no. 1, pp. 32-49
- 20 Gharavi-Alkhansari M., Huang TS., 1998. A fast orthogonal matching pursuit algorithm. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98)*, Seattle, Wash, USA 3: 1389-1392.
- Gonzalez R. C., Woods R., 2002. *Digital Image Processing*. Prentice Hall, 2nd edition.
- Gorodnitsky I. F. and Rao B. D., 1997. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted norm minimization algorithm. *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600-616.
- 25 Haldorsen H. H., Damsleth E., 1990. Stochastic modelling. *Journal of Petroleum Technology* 42 (4), 404-412.
- Hastie T., Tibshirani R., Friedman J., 2009. *The elements of statistical learning*. 2nd Edition, Springer Series in Statistics Springer New York Inc., New York, NY, USA
- Honarkhah M., 2011. Stochastic simulation of patterns using distance-based pattern modeling. Ph.D. thesis, Stanford University.
- 30 Horev I., Bryt O., and Rubinstein R., 2012. Adaptive image compression using sparse dictionaries. In *International Conference on Systems, Signals and Image Processing*, pages 592-595. IEEE.



- Khaninezhad M. M., Jafarpour B., Li L., 2013. Sparse geologic dictionaries for subsurface flow model calibration: Part I. Inversion formulation. *Advances in Water Resources*.
- Lesage S., Gribonval R., Bimbot F., and Benaroya L., 2005. Learning unions of orthonormal bases with thresholded singular value decomposition. *IEEE Int. Conf. Acoust., Speech, Signal Process.*
- 5 Lie K.-A., 2015. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST)*. SINTEF ICT.
- Mairal J., Bach F., Ponce J., Sapiro G., 2009. Online dictionary learning for sparse coding, in *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, pp. 689-696, ACM.
- 10 Mairal J., Elad M., and Sapiro G., 2008. Sparse representation for color image restoration, *IEEE Trans. on Image Processing*, 17(1):53–69.
- Mallat S., Zhang Z., 1993. Matching pursuits with time–frequency dictionaries. *IEEE Trans. Signal Process.*, 41 (12), pp. 3397–3415
- Mariethoz, G., Renard, P., Apr. 2010. Reconstruction of incomplete data sets or images using direct sampling. *Mathematical Geosciences* 42 (3), 245-268.
- 15 Mariethoz G., Caers J., 2015. *Multiple-Point Geostatistics: Stochastic Modeling with Training Images*. John Wiley & Sons, Ltd, Chichester, UK.
- Rubinstein R., Bruckstein A.M., and Elad M., 2010a. Dictionaries for sparse representation modeling. *Proc. IEEE*, vol. 98, no. 6, pp. 1045-1057.
- 20 Rubinstein R., Peleg T., and Elad M., 2013. Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model. *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661-677.
- Rubinstein R., Zibulevsky M., and Elad M., 2010b. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553-1564
- Rubinstein R., Zibulevsky M., and Elad M., 2008. Efficient Implementation of the K-SVD Algorithm using Batch
- 25 Orthogonal Matching Pursuit. Technical report.
- Skretting K., Engan K., 2011a. Image compression using learned dictionaries by RLS-DLA and compared with K-SVD. in *Proceedings ICASSP 2011*, Pargue, Czech Republic, June 2011, vol. 1, pp. 1517-1520.
- Skretting K., Engan K., 2011b. Learned dictionaries for sparse image representation: properties and results. *Proceedings of the SPIE*, Volume 8138, id. 81381N.
- 30 Skretting K., Engan K., 2010. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, vol. 58, pp. 2121-2130
- Skretting K., Husøy J.H., 2003. Partial search vector selection for sparse signal representation. *NORSIG-03*, Bergen, Norway.



- Skretting K., Husøy J.H., Aase, S.O., 1999. Improved Huffman coding using recursive splitting. NORSIG-99 Conference on Signal Processing and Signal Compression.
- Starck J.L., Murtagh F., Fadili J., 2010. Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity. Cambridge University Press, Cambridge.
- 5 Strebelle S., 2000. Sequential simulation drawing structures from training images. Ph.D. thesis, Stanford University.
- Tibshirani R., 1996. Regression Shrinkage and Selection via the LASSO. J. Royal Statistical Soc. B, vol. 58, no. 1, pp. 267-288.
- Tropp J.A., 2004. Greed is good: Algorithmic results for sparse approximation. IEEE Trans. Inform. Theory, 50
10 (10), pp. 2231–2242
- Tropp J.A., Gilbert A.C., 2007. Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inform. Theory, 53 (12), pp. 4655–4666
- Tropp J.A., Gilbert A.C., 2006. Signal recovery from partial information via orthogonal matching pursuit. IEEE Trans. Inform. Theory, 53 (12), pp. 4655–4666
- 15 Vidal R., Ma Y., and Sastry S., 2005. Generalized principal component analysis (GPCA),” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 12, pp. 1945–1959.
- Wu J., 2007. 4D seismic and multiple-point pattern data integration using geostatistics. Ph.D. thesis, Stanford University.
- Zhang T., 2006. Filter-based training pattern classification for spatial pattern simulation. Ph.D. thesis, Stanford
20 University.
- Zou H., Hastie T., and Tibshirani R., 2006. Sparse principal component analysis. Journal of Computational and Graphical Statistics, 15(2):265-286.

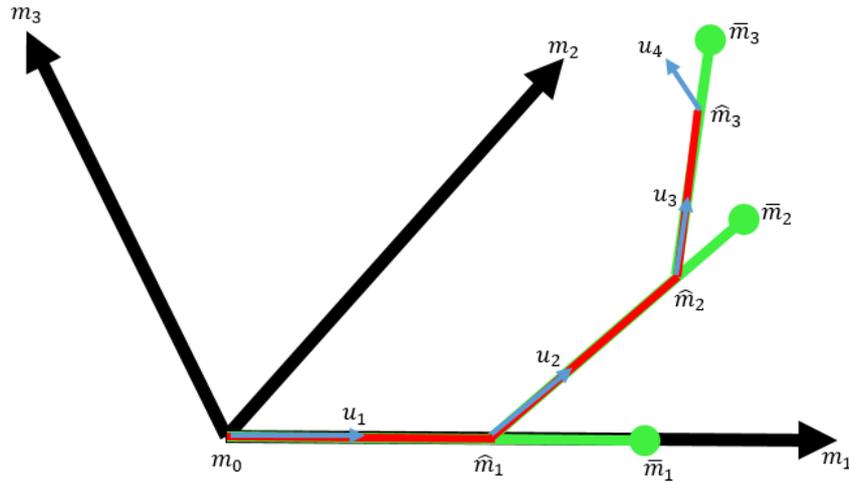


Figure 1: Geometrical interpretation of the LARS algorithm; m_1 , m_2 , and m_3 are the variables; u_1 , u_2 , and u_3 are the norm for each variable; \bar{m}_1 , \bar{m}_2 , and \bar{m}_3 are the least squares solution corresponding to each variable. \hat{m}_1 , \hat{m}_2 , and \hat{m}_3 are corresponding LARS solutions. After Efron et al. (2004).

5

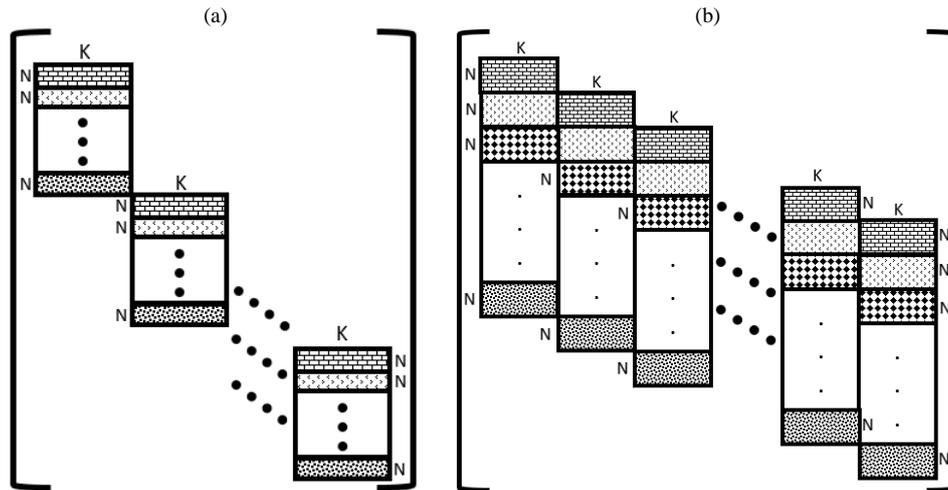


Figure 2: schematic of (a) non-overlapping matrix G_{aug} as in Eq. (3); (b) the overlapped matrix G_{aug} as in Eq. (7)

10

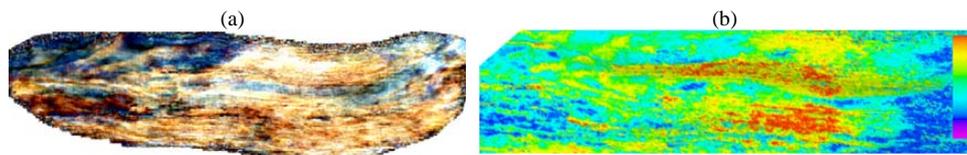


Figure 3: (a) Spectral decomposition of the Asmari reservoir; (b) The acoustic impedance image corresponding to (a)

5

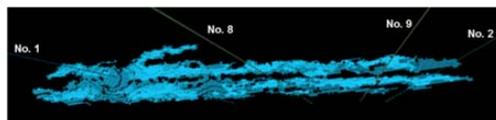


Figure 4: The gas fingering feature from the injection well No. 8

10

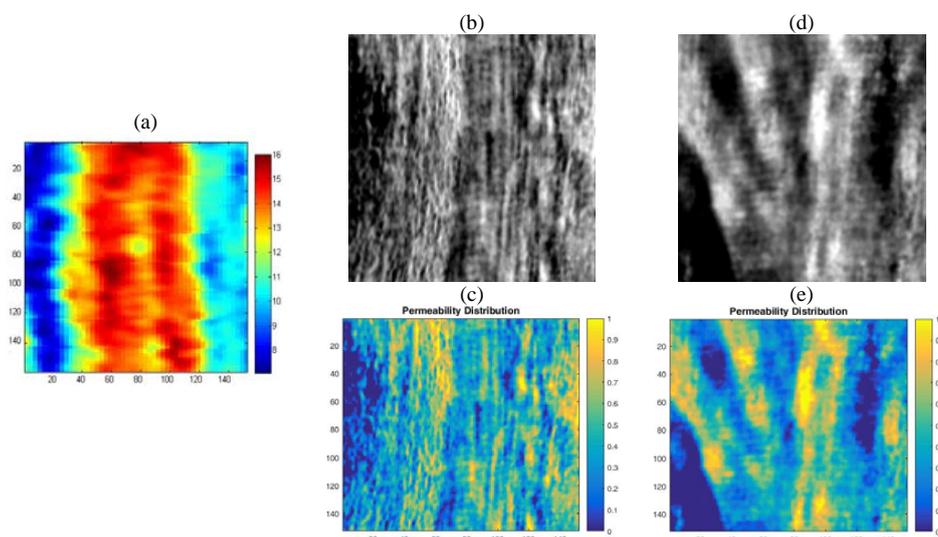


Figure 5: (a) Porosity distribution for the simulation model; (b) stationary fracture network training image; (c) permeability distribution corresponding to (a) for the simulation model; (d) nonstationary deltaic training image; (e) permeability distribution corresponding to (d) for the simulation model

15

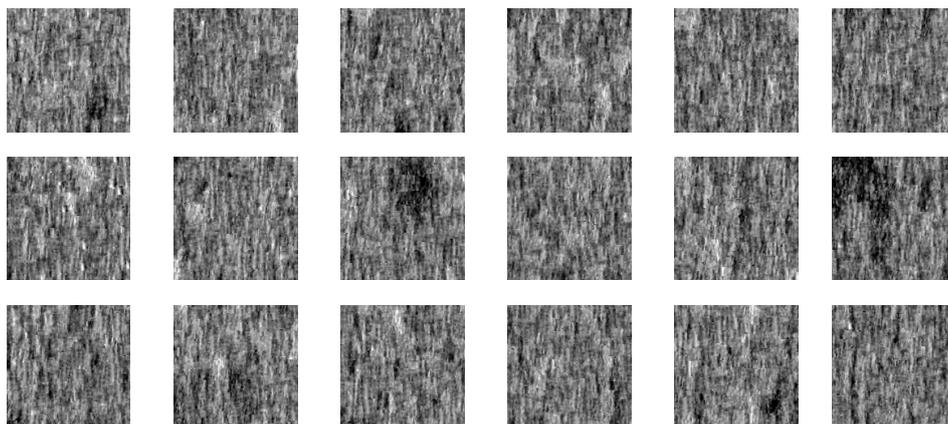


Figure 6: Few randomly selected DisPat MPS generated realizations for the stationary fracture network training image

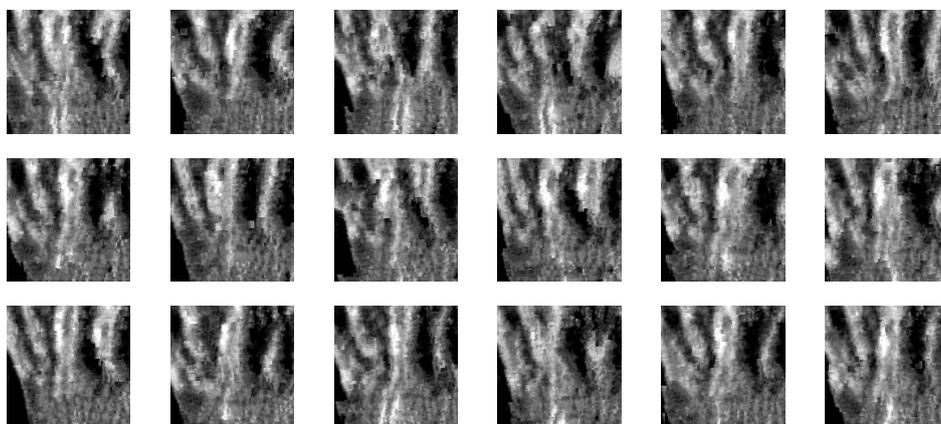


Figure 7: Few randomly selected DisPat MPS generated realizations for the nonstationary deltaic system training image

5

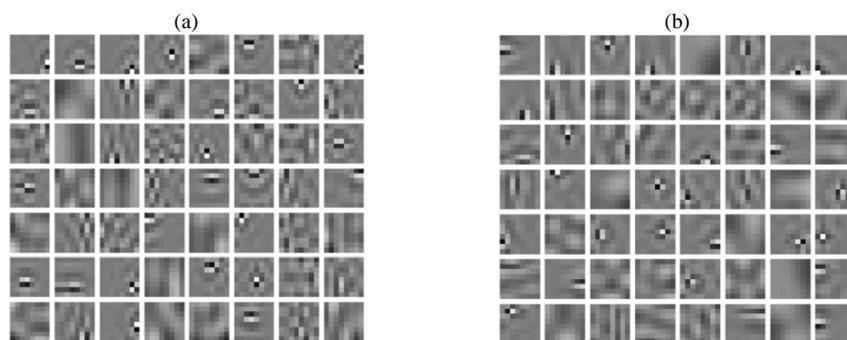


Figure 8: (a) Few atoms from the ILS-DLA-LARS trained dictionary for the fracture network training images;
(b) Same for the deltaic system training images

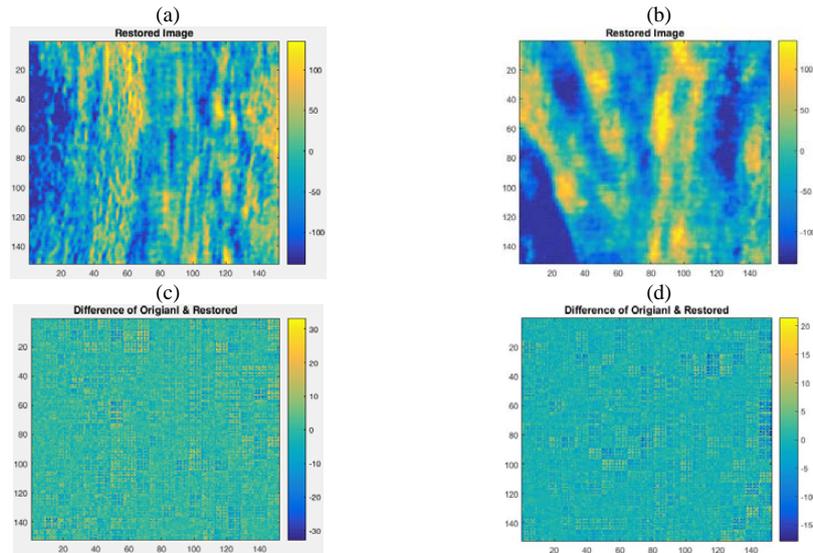


Figure 9: (a) The sparsity-based compressed permeability model for fracture network using ILS-DLA dictionary training with the LARS; (b) Same for deltaic system; (c) Difference between the compressed and the main model for the fracture network; (d) Same for the deltaic system

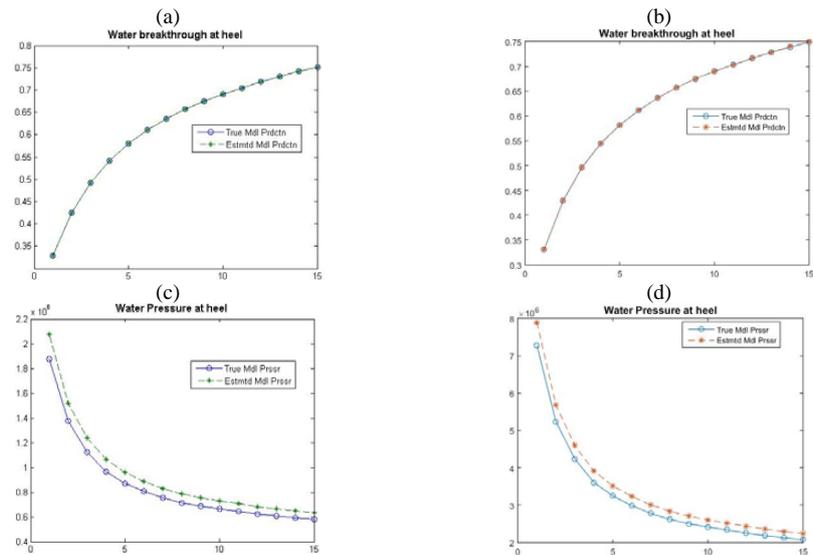


Figure 10: (a) Production profile at the observation well for the sparsity-based compressed fracture network model and its comparison with the main model; (b) Same for the deltaic system; (c) Pressure profile at the observation well for the compressed fracture network model and its comparison with the main model; (d) Same for the deltaic system

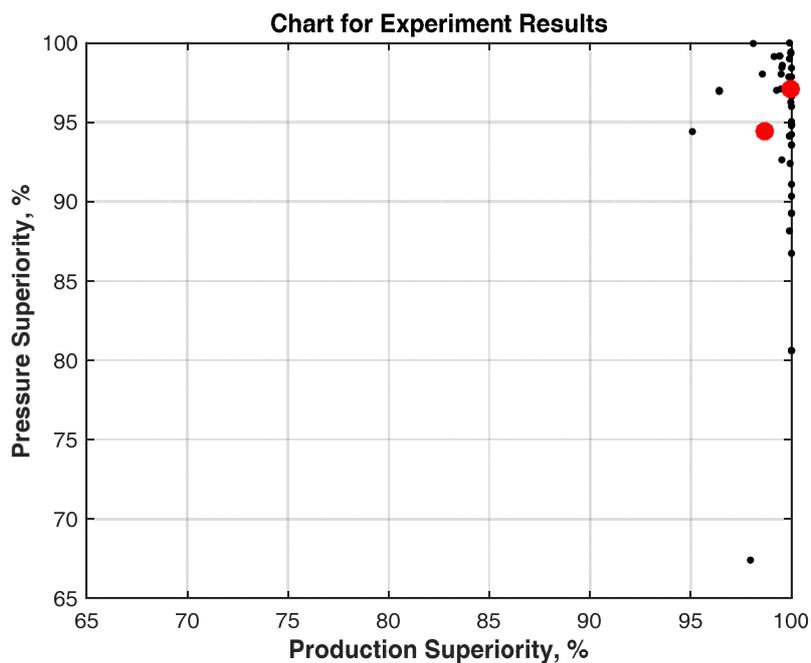


Figure 11: Results of other sets of experiments providing support for this study. The exaggerated red-filled circles indicate the results presented in this paper.