Abstract

Providing an overview of an ensemble of oil reservoir models could help users compare and analyze their characteristics. Approaches that show a single model at a time may hamper analysts’ understanding of the whole model set. In this paper, we propose two visualization approaches that show multiple reservoir models, simultaneously and on a single screen, with the goal of helping users to compare models and improve their understanding of ensemble characteristics. First, we calculate 2D models from the ensemble's 3D models. We then create two visualizations that represent ensembles of these 2D models. The Small Multiples approach lays out heatmaps of 2D models side-by-side on a grid. Pixelization approach shows $n$ 2D models in a single heatmap, where each cell $(i, j)$ contains $n$ subcells that represent values in the coordinate $(i, j)$ of each model. Both approaches display their elements (heatmaps and subcells) clustered by X-means, which may help analysts identify similarities and representative models in the ensemble. We used two types of distance matrices: based on Euclidean distance of models for a given property or, based on Euclidean distance of feature vectors of the 2D models. We tested our approaches within models based on Brazilian benchmark cases corresponding to a turbiditic reservoir (UNISIM-I-D/M/H) and a presalt carbonatic reservoir (UNISIM-II-D). As a result, the Small Multiples approach presented clusters of similar models for some properties of the ensembles we studied, e.g. eight clusters of porosity values in UNISIM-II-D's ensemble. This fact suggests that eight representative models can represent the ensemble, regarding porosity. Also, a Pixelization approach revealed patterns that happen in specific regions of all models of an ensemble, such as an abrupt change of porosity values in the northwest region of UNISIM-I-M's models. Both approaches have the potential to help analysts perceive situations that would be improbable to observe in a graph with only mean values for each cell. Therefore, our proposal can be helpful to users who need to deal with uncertainties and have an overview of ensembles of models for better understanding and decisionmaking, e.g. when they need to choose representative models for a process of decision analysis related to petroleum field development and management.

Introduction

Oil field development and management decisions depend on analyses of data related to reservoir models. Dealing with multiple models is very common as there are always uncertainties associated to the field.
Sampling techniques are used to select a set of models that cover all classes of uncertainties of each property. Even after assimilation data procedures, which reduce the size of the initial sampled set, the filtered ensemble of models (the models that reproduce observed production data) may still have hundreds of models to be analyzed. This number may increase if the analysis uses representative models of multiple production strategies, according to the 12-step methodology developed by Schiozer et al. (2019).

If the only way of inspecting this set is to interactively select one model at a time and analyze each of its layers, it then becomes unfeasible to obtain an overview of those models and their similarities. In an oil reservoir, each model may be considered as a set of: (a) three independent variables \((i, j, k)\), which define the coordinates of a reservoir grid block (or cell); and (b) a set of dependent variables, static and dynamic, which define block properties such as oil, water and gas saturation, net-to-gross, permeability, and porosity, among others. A user who aims to have an overview of a single property of a single model has two possibilities. The first is to interact with a 3D heatmap (with rotation and slicing capabilities in order to overcome occlusion problems). The second is to handle a bidimensional slice of the model that a user can interactively select (e.g., choosing to show the first layer of the model, in which \(k=1\)). Interaction is also necessary to select which property the block color should represent. In other words, the current approaches to provide an overview of a single model to a user involve a series of user actions, each followed by a new graphical version of the reservoir model. Therefore, the user iteratively creates a mental representation (a.k.a. mental model) of the reservoir model. The bigger the size of the reservoir, the harder the cognitive task of creating this mental model. This problem is even worse when a user deals with a set of models instead of a single one.

Alternatively, statistical approaches may reduce a set of models to a single summary model that represents a statistical measure of the set. E.g. each block \((i, j, k)\) in the summary model may represent mean (or standard deviation) values of its respective blocks \((i, j, k)\) in each model of the ensemble. However, statistical measures may hide subtleties in the data. A classical example is the Anscombe's quartet (Figure 1): four datasets in which the statistical measures are the same (e.g. mean, deviation and correlation measures) but whose curves are very distinct (Anscombe, 1973). Therefore, providing a visual inspection of the ensemble's raw data would be preferable for a user to create a mental model of the ensemble without bias, instead of visualizing statistical measures only.
This work proposes alternative ways for visualizing, on a single screen, the values of properties in multiple reservoir models with the goal of helping users to compare models and improve their understanding of ensemble characteristics. The Pixelization and Small Multiples approaches represent 2D visualizations of the 3D models. The Small Multiples approach lays out heatmaps of models (simplified from 3D to 2D) side-by-side on a grid. The Pixelization approach merges \( N \) 2D models in a single heatmap. Both techniques define layouts based on clustering models. We also used X-means and feature vectors of models as an alternative way to calculate clusters of models. Additionally, the clusters can help analysts identify representative models based on the ensemble.

The main contributions of this paper are:

a. Two complementary approaches to visualize properties of an ensemble of reservoir models on a single screen;
b. A definition of a feature vector for the values of a given property of a model;
c. An alternative way to calculate a distance matrix to use in clustering algorithms, based on Euclidean
distances of feature vectors of models.

We organized this work as follows: The "Theoretical Background" section summarizes Information
Visualization techniques related to multidimensional data visualization, and focuses on Small Multiples and
Pixelization. The "Methodology" section explains how we use Pixelization and Small Multiples approaches
to present multiple models. The "Results and Discussion" section presents a series of visualizations and
discusses how they may enable the understanding of the characteristics of an ensemble. Finally, the
"Conclusion and Future Works" section highlights the benefits of our visualization approaches and future
research directions.

**Theoretical Background**

*Information Visualization* is a research area that studies how to develop interactive graphics which are useful
to obtain information from a dataset (Mazza, 2009; Card et al., 1999). One crucial concept of this area is
the *visual mapping*, i.e., in which the spatial of graphical property (such as color, size, and shape) we will
use to represent each dataset variable (a.k.a. field or dimension).

One of the main concerns of this area is how to represent multidimensional datasets; in other words, how
to show a data table with more than three variables. Ward et al. (2015) state the following choices to make
in multidimensional visual analysis:

1. Use of graphical properties (*dimensional embedding*) to represent some of the variables, and not only
   spatial ones;
2. Selection of a subset of variables to represent (*dimension subsetting*);
3. Transformation of a high-dimensional dataset into a low-dimensional one while trying to retain the
   relationships between the data items (*dimensional reduction*); and
4. Using *multiple displays*, each one with some of the dimensions.

Ward et al. (2015) also define four classes of multidimensional data visualization techniques: point-based,
line-based, region-based and hybrid. *Point-based techniques* represent each tuple by a single mark (e.g. a
small bullet). A first example of these techniques is the *classical multidimensional scaling*. This dimension
reduction technique calculates coordinates in $\mathbb{R}^2$ for every tuple in $\mathbb{R}^N$ of the dataset. The distances between
each pair of coordinates in $\mathbb{R}^N$ mimic the dissimilarities of the tuples in the $\mathbb{R}^N$ space. *Scatterplot matrices*
is a second example of this type of technique. It creates $N \times N$ interactive scatterplots, each one with a possible
pair of the $N$ variables of the dataset.

*Line-based techniques* use lines to represent either dataset records or values of a same variable. Line
graphs of multiple series are a common way to use lines for linking values of a same variable in multiple
tuples. Another example is the parallel coordinates plot (Insemberg, 1985), which draws $N$ parallel axes
and a set of polyline, one for each record.

*Region-based techniques* use filled regions, whose color, size or shape represent values. We highlight
the heat map technique (a.k.a. matrix visualization) (Wu et al., 2008), which resembles the data table
organization. In this technique, table columns and rows represent variables and tuple identifications,
respectively; each cell color indicates the value of a single variable in a specific tuple. Distinct color maps
may help unveil distinct characteristics of the dataset.

*Hybrid techniques* mix some of the previous types. In this paper, we highlight Pixelization approaches,
a.k.a. pixel-based visualization techniques or dense pixel displays (Keim et al., 1994; Keim, 2000; Ward
et al., 2015). Given a dataset with $m$ records and $n$ variables, we summarize Pixelization approaches as
following steps:

1. Define $n$ regions on the screen, each one related to a variable;
2. Divide each region as a matrix with at least \( m \) cells;
3. Sort the records according to defined criteria (e.g. the ascendant order of a variable);
4. Choose a screen-filling layout (e.g. snake, spiral or Peano-Hilbert);
5. Choose a color map; and
6. For each variable \( v \), paint the cells of the respective region with the values of \( v \) for each record according to the ordering of the record, the layout and the color map defined in Steps 3 to 5.

The name "Pixelization" highlights that the cell size in Step 2 may have the size of a pixel. Therefore, a small region on the screen, such as an area with 100 \( \times \) 100 pixels, may show 10000 pixels, which corresponds to 10000 values of a single variable. It is a hybrid technique because each matrix is a heatmap (region-based technique) and other techniques define the position of each heatmap, such as a tabular display (again, region-based) or a multidimensional scaling (as if the heatmap were a single point in this point-based technique).

Figure 2 provides an illustrative example that represents a dataset with four variables (columns) and 25 tuples (rows) arranged in four heatmaps organized as a 5 \( \times \) 5 matrix each. We used rectangles instead of pixels. The colormap uses green cells for higher values, white for mean values, and red for lower values. The layout for region filling follows a snake (or left-right) filling layout. This technique enables the inference of positive and negative correlations among variables, due to their similarities and differences. For example, variables A (which defined the dataset order) and C have a positive correlation; B has a negative correlation to A and C; and there is no conclusion about correlations from A, B or C to D. The Pixelization uses multiple displays (one for each variable) and dimension embedding (value mapped to a color).

A good choice of a screen-filling layout (Step 4) is relevant. Given a list of elements to place on the screen, a good screen-filling layout defines that elements which are close to each other in this list are near each other on the screen (Ward et al., 2015). Figure 2 uses a left-to-right layout (a.k.a. snake layout). Figure 3 shows the layout ordering for a Peano-Hilbert (or Hilbert) curve - a recursive screenfilling layout technique that fills every cell of a square matrix exactly once (Ward et al., 2015; Castro and Burns, 2007). The Peano-Hilbert is a recursive layout, but works only for square matrices with size \( 2^i \times 2^i \), where \( i \) is the number of rows and columns of the layout matrix. There are generalizations of the Peano-Hilbert curve for non-square matrix layout, called Pseudo-Peano-Hilbert curves (e.g. Zhang et al., 2006). In this paper, we used both Peano-Hilbert and pseudo-Peano-Hilbert curves in our visualizations.
An alternative way of using multiple displays for multivariate data visualization is the *Small Multiples* technique (Tufte, 1990). We summarize it in the following steps:

1. Define \( r \) regions according to defined criteria (e.g. the values of one or two discrete variables);
2. Define the coordinates of these regions on the screen, according to the previously-defined criteria;
3. Define the data records to present in each region, according to the definition of each region in Step 1;
4. Define the same visualization technique and visual mapping to use for all regions; and
5. Plot each region's data with the technique and mapping from Step 4.

According to this definition, each region must be similar to the other ones in terms of how to show the data; therefore, the user perceives the differences among them and they may reveal relevant data variations regarding the division criteria of Step 1.

*Figure 4* exemplifies the use of small multiples to compare reservoir models of the UNISIM-I-D benchmark case. The comparison uses six input variables (therefore, \( r=6 \)) and two output variables (\( NPV \) and \( Wp \)). It is worth noting that the variable \( bl \), which indicates absence or presence of an east region of reservoir blocks in each model, seems to be related to \( NPV \) but not to \( Wp \). Given a common ground (the same pair of axes and the points at the same positions), the technique enable the users to analyze the behavior of the six input variables.
Methodology

Our methodology comprises the following steps, which we detail in this section:

1. Defining preprocessing and data transformations for the ensemble data;
2. Defining visual mapping for Small Multiples approach;
3. Defining visual mapping for Pixelization approach; and
4. Testing the visual mappings in ensembles.

Preprocessing and data transformations

Following Shneiderman’s Visual Information Seeking mantra (Shneiderman, 1996), at first we want to create an overview of the entire set of models for each property \( p \), aiming to provide a starting point for visual analysis. To create this overview, we need to convert each 3D model into 2D and calculate distance matrices of the models.

Creating 2D models. Let \( M^3D_p = \{m^3D_p(i, j, k)\} \), \( i \in [1, i_{max}], j \in [1, j_{max}], k \in [1, k_{max}] \) be the set of values of \( p \) in a 3D model. Coordinates \( i \) and \( j \) define positions in a horizontal layer; coordinate \( k \) defines the deepness of the layer. Let \( M^2D_p = \{m^2D_p(i, j)\} \), \( i \in [1, i_{max}], j \in [1, j_{max}] \) be the set of values of \( p \) in a 2D model. We define \( m^2D_p(i, j) = \text{mean}_{k \in [1, k_{max}]}(m^3D_p(i, j, k)) \). In other words, each cell \( (i, j) \) in the 2D model summarizes an entire column of 3D blocks that share the same values of \( i \) and \( j \). The mean function may be arithmetic, harmonic or geometric mean, according to user needs. The use of the arithmetic mean can be a good choice for most properties of the ensemble. However, the use of the harmonic mean for vertical permeability or fractured systems is suggested due to the nature of this property. This first preprocessing step generates a set of 2D models that will be the base of our visualizations.

Creating distance matrices. We also derive distance matrices from the original ensemble. In a first approach, define \( p_{max} \) as the maximum value of a property \( p \) in the entire ensemble. Let \( M^3DNorm_p \) be a
normalized version of $M^3D_p$, i.e. $m^3DNorm(i, j, k) = \frac{m^3D(i, j, k)}{P_{\text{max}}}$. Let $M^3DNormLin_p$ be a linearized version of $M^3DNorm$. We define the distance $d^2_p(M^{(r)}, M^{(s)})$ between two models $M^{(r)}$ and $M^{(s)}$ for a given property $p$ as the squared Euclidean distance between $M^3DNormLin_p$ and $M^3DNormLin_s$.

Given an ensemble $\omega = \{M^{(1)}, M^{(2)}, \ldots, M^{(k)}\}$, the distance matrix of the entire ensemble $\omega$ for the property $p$ is $\Delta_{p,\omega} = [x_{r,s}]$, $x_{r,s} = \sqrt{d^2_p(M^{(r)}, M^{(s)})}$, $r \in [1, |\omega|], s \in [1, |\omega|]$. We define the distance matrix of all models of the ensemble as $\Delta_{\omega} = \left[ \sqrt{\sum_{p=1}^{|P|} (w_i d^2_p(M^{(r)}, M^{(s)}))} \right]$, $r \in [1, |\omega|], s \in [1, |\omega|]$, where $P = (p_1, p_2, \ldots)$ is the set of properties, and $w_i$ is a weight defined to the property $p_i$.

**Creating distance matrices based on feature vectors.** Calculating $\Delta_{\omega}$ may be a time-consuming activity. E.g., our initial implementations spent days to calculate it for the UNISIM-I-D benchmark case. Therefore, we defined a faster procedure for calculating an approximation of this matrix, based on feature vectors, which took a few minutes to compute. A feature vector stores relevant properties of an object. Classification and clustering procedures can compare feature vectors instead of their respective (more complicated) objects, and this approach tends to spend less computational time. A common use of feature vectors is to summarize relevant information about an image as, for example, image comparison (Torres et al., 2003).

We define a feature vector of $M^2D_p$, as follows. First, define 3 classes of values of $M^2D$: lower values (less than or equal to percentile 33), middle values (between percentiles 33 and 67), and higher values (greater than or equal to percentile 67). $C_{\text{set}, \text{class}} = [c_1, c_2, \ldots, c_{\text{set}, \text{class}}]$ is a vector that counts the amount of values that belongs to class inside set; class may be "lower", "middle" or "higher", and set may be "rows" or "columns". E.g., for a 2D model with 20 rows, $C_{\text{rows}, \text{higher}}$ is a vector with 20 elements and each element counts the number of higher values in their respective rows of $M^2D_p$. We define $F_p = [C_{\text{rows}, \text{higher}}, C_{\text{rows}, \text{middle}}, C_{\text{rows}, \text{lower}}, C_{\text{columns}, \text{higher}}, C_{\text{columns}, \text{middle}}, C_{\text{columns}, \text{lower}}]$, i.e., a concatenation of vectors with count values.

We define the distance $d^2_{FV}(M^{(r)}, M^{(s)})$ between two models $M^{(r)}$ and $M^{(s)}$ for a given property $p$ as the Euclidean distance between $F_p(M^{(r)})$ and $F_p(M^{(s)})$. The distance matrix of the entire ensemble $\omega$ for the property $p$ is $\Delta_{p,\omega}^{FV} = [d^2_{FV}(M^{(r)}, M^{(s)})]$, $r \in [1, |\omega|], s \in [1, |\omega|]$. We define the distance matrix of all models of the ensemble as $\Delta_{\omega}^{FV} = \sqrt{\sum_{p=1}^{|P|} (w_i d^2_{FV}(M^{(r)}, M^{(s)}))}$, where $w_i$ is the weight defined to the property $p_i$. Therefore, we generate $\Delta_{p,\omega}$ and $\Delta_{\omega}^{FV}$ for each property $p$. In this paper, we explore how to use both for clustering purposes. Finding evidence to support the use of the latter instead of the former, or even the use of $\Delta_{\omega}^{FV}$ instead of $\Delta_{\omega}$, is a step outside the scope of this paper.

**Clustering models.** Defining clusters of models is relevant in order to enhance similarities and differences among them. The literature identifies many clustering methods, from which we highlight hierarchical clustering and K-means as examples. Both need a user-defined criterion to define the number of clusters; e.g. the number itself, or an analysis of distance between clusters to merge. In this work, we propose the use of X-means (Pelleg and Moore, 2000), a clustering algorithm derived from K-means. Instead of receiving as input the number of clusters (K) to return, X-means calculates the ideal number of clusters for a given dataset. This number must lie in a range provided as input.
Visual mapping for Small Multiples approach

We propose two approaches to visualize an ensemble $\omega$ of 2D models for a given property $p$. Both use as main concept the idea of a heatmap - a matrix whose cell colors represent values of $p$. We adopted a color scale that maps values of $p$ to a smooth transition of color hues from blue (lowest values) to green (mean values) and then to red (highest values), with fixed saturation and brightness. The choice is due to consistency with other visualizations provided by software frequently used in reservoir characterization.

In our Small Multiples approach, given an ensemble $\omega$ and a property $p$, we represent a model $M^2_p$ by a small heatmap. We place each heatmap into a cell of a $n \times m$ grid, where $nm \geq |\omega|$. We define the grid dimension in such a way that the picture dimension becomes closer to a golden rectangle, i.e.

$$\frac{n j_{\text{max}}}{m i_{\text{max}}} = \frac{1}{1.618}$$

We consider it relevant to place similar models next to each other, aiming to simplify the comparison among them. In order to achieve this, we first use X-means to cluster models, based on $\Delta_{p,\omega}$ or $\Delta^{FV}_{p,\omega}$. After that, we sort these clusters by size in descendant order and place each model of each cluster into the grid, according to a pseudo-Peano-Hilbert curve. Also, we outline each cluster to differentiate it from the others. Figure 5 illustrates these characteristics in a visualization of a synthetic and instructional ensemble with 16 models with 13 cells each.

![Figure 5](image.png)

Figure 5—Example of small multiples approach visualizing values of a fictitious propriety of 16 models, clustered into 3 groups (white lines). Numbers are shown for instructional reasons only.

Visual mapping for Pixelization approach.

In our Pixelization approach, we also represent the property $p$ of an ensemble $\omega$ with a grid of heatmaps, with the same color scale, but in a different organization. The grid of this visualization has size $j_{\text{max}} \times i_{\text{max}}$, i.e., it has the same dimension of a single 2D model. Each cell $(i, j)$ of this grid has a heatmap with $|\omega|$ subcells. Each subcell represents the value of $p$ for the cell $(i, j)$ of one model of $\omega$.

We opted for defining that a cell should be a square in order to maintain the aspect ratio of the picture similar to the aspect ratio of a single model. Therefore, we define a cell as a $n \times n$ heatmap. Given that the heatmap must have at least $|\omega|$ subcells, we defined

$$n = \sqrt{|\omega|}$$

In order to unveil similarities we sort models in the same way as in the Small Multiples approach. However, we use the order in this visualization to sort the $|\omega|$ subcells inside the heatmap, according to a Peano-Hilbert curve.

The result is a visualization which looks like a single 2D model, but that represents an entire ensemble and attempts to highlight the global similarities in a local level of detail.
Figure 6 exemplifies this approach with the same ensemble from Figure 5, with 13 cells, with 16 values each, and each value from a model of the ensemble. It is important to note that the top-right cell has low variability in all models. Other cells seem to be similar, such as two of the three leftmost cells.

Figure 6—Example of pixelization approach using the same ensemble from Figure 5. Numbers are shown for instructional reasons only.

Testing the visual mappings in ensembles
We opted to validate our visual mappings within Brazilian reservoir models based on UNISIM benchmark cases, where UNISIM-I corresponds to a turbiditic reservoir (UNISIM-I-D, UNISIM-I-H and UNISIM-I-M) and UNISIM-II is related to a pre-salt carbonatic reservoir (UNISIM-II-D). Those cases are freely available at UNISIM benchmark cases’ web site.

The test consists of (1) creating Small Multiples and Pixelization visualizations for these benchmarks, for properties such as oil saturation (SO), water saturation (SW), permeability (PERMI), net-to-gross (NTG), and porosity (POR); and (2) analyzing the results to discover patterns, outliers or tendencies that could be relevant for a better understanding of the ensembles.

For Pixelization approaches, we also added the location of producer and injector wells, provided as part of the production strategies applied to the reservoirs in the benchmarks or from optimization processes.

Results and Discussion
In this section, we present and discuss some of the visualizations that we obtained as the result of applying our methodology. We group them by benchmark case.

UNISIM-I-D Benchmark Case.
For UNISIM-I-D, we show the Pixelization and Small Multiples visualizations for oil saturation (SO) of the 214 models selected after an assimilation data process. The Small Multiples approach in Figure 7 reveals nine clusters of models. The clustering step used \( \Delta_{p,o} \) as the distance matrix and found nine clusters. For comparison purposes, Figure 8 used \( \Delta_{p,o}^{FE} \) as the distance matrix and found seven clusters. Both figures reveal up to three clusters whose models do not have an east region (a structural uncertainty) when compared to the remaining models. Other clusters seem to have high similarity in their west regions, with big differences related to the oil saturation in the east region. It is worth noting that an analyst probably would not notice these clusters in a graph with only mean values for each cell.
Figure 7—Small multiples visualization of oil saturation in the 214 models for the UNISIM-I-D benchmark case at the end of the production history. Clustering based on the use of $\Delta So_{\omega}$ as distance matrix. 9 clusters were found.

Figure 8—Small multiples visualization of oil saturation in the 214 models for the UNISIM-I-D benchmark case at the end of the production history. Clustering based on the use of $\Delta_{SO_{\omega}}$ as distance matrix. 7 clusters were found.
Figure 9 presents the use of the Pixelization approach for oil saturation, using $\Delta_{SO_{\omega}}^{FV}$ as distance matrix. The absence of the east region in some models is also perceivable in this visualization due to many small black areas around the east of the map. The west and central regions seem to have less uncertainty than the east, based on the oil saturation map. It is important to note that in this visualization we added the names and the location of wells. Dot rectangles indicate injector wells, and straight rectangles represent producer wells.

**UNISIM-I-M Benchmark Case**

For UNISIM-I-M, we highlight a Pixelization of permeability (Figure 10) for 48 models selected after an assimilation data process, using $\Delta_{SO_{\omega}}^{FV}$ as distance matrix for clustering. The visualization reveals a red region next to wells PROD021 and INJ019 with an abrupt change of values (from red to green values), which suggests that the models were manually edited. It was due to manipulation of petrophysical properties since there is no barrier in that area.
UNISIM-I-H Benchmark Case

Figure 11 represents the Pixelization of the porosity values for 407 models selected after an assimilation data process. We highlight the clustering at the northwest region, whose cells are green on the left and blue on the right. This indicates the presence of two distinct set of models, one set with porosity close to the mean and the other with below average porosity.

UNISIM-II-D Benchmark Case

For the UNISIM-II-D we show a Small Multiples visualization of the porosity of the 199 models selected after an assimilation data process (Figure 12). This visualization is based on the use of $\Delta_{\text{POR},\omega}^{FV}$.
Conclusion and Future Work

Analyzing data related to multiple models with uncertainties is an important task for decision-making in oil field development and management activities. In this paper, we provide two complementary approaches so that analysts can understand the characteristics of an ensemble and its uncertainties. Our proposals - Small Multiples and Pixelization visualizations - not only show mean values of a given property, but the raw data itself. A clustering procedure grouped data from similar models, aiming to enhance the perception of similar models. Visualizing the clustered models helps reveal subtleties that other common visualization approaches would not present to the analysts. Moreover, we presented a way to use feature vectors as a summary of the values of a given property of a reservoir model. We used these vectors to compare pairs of models and then to construct distance matrices, which we used for clustering purposes.

Future work involves:

a. Adding interactivity to the proposed visualizations;
b. Integrating these visualizations to a tool where they can be used for decision making in real cases;
c. Comparing the distance matrices $\Delta_{p,\omega}$ and $\Delta_{p,\rho}$, as well as $\Delta_{\omega}$ and $\Delta_{\rho}$;
d. Studying other possible feature vectors to use for computing distance matrices; and
e. Defining a broader clustering approach that could help define representative models, while taking into consideration all properties instead of a single one.

Acknowledgements

We developed this work with the support of Petrobras within the ANP R&D "commitment to research and development investments" and Energi Simulation. The authors would like to thank the support of the Center of Petroleum Studies (CEPETRO-UNICAMP/Brazil), the Department of Energy (DE-FEM-UNICAMP/Brazil) and the Research Group in Reservoir Simulation and Management (UNISIM-UNICAMP/Brazil).
References


Castro, J., and Burns, S. Online Data Visualization of Multidimensional Databases Using the Hilbert Space–Filling Curve.


