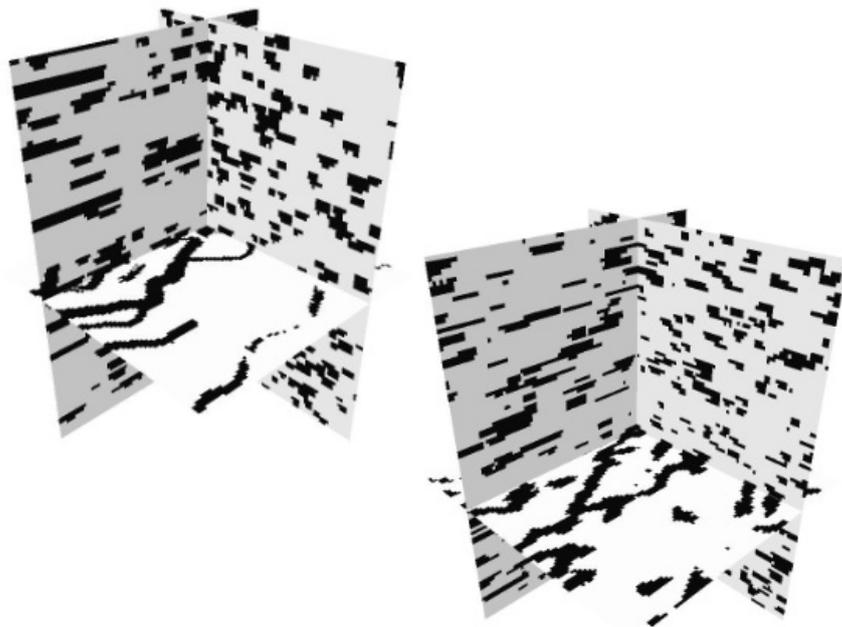


Markov mesh model specification for facies modeling



Note no
Author
Date

SAND/09/10
Marita Stien, Odd Kolbjørnsen
27th January 2011

Norwegian Computing Center

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modeling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Title	Markov mesh model specification for facies modeling
Author	Marita Stien, Odd Kolbjørnsen
Date	27th January 2011
Publication number	SAND/09/10

Abstract

Spatial continuity of facies is one of the key factors controlling flow in reservoir models. Traditional pixel-based methods like truncated Gaussian random fields and indicator simulation are only based on two point statistics, which is not sufficient to capture the complex facies structures. Current methods for multi-point statistics, either lack a consistent statistical model specification or they are too computer intensive to be applicable. We propose a Markov mesh model based on generalized linear models for geological facies modeling. The approach defines a consistent statistical model which is facilitated with efficient estimation of model parameters and generation of realizations. In our presentation we include formulation of the general framework, model specification in two and three dimension, and detail on how the parameters can be estimated from a training image. We illustrate the method using multiple training images, including binary and trinary images and simulations in two and three dimensions. We also do a thorough comparison to the snesim approach. We find that the current model formulation is applicable for multiple training images and compares favorable to the snesim approach in our test examples. The method is highly memory efficient.

Keywords	Markov mesh models, sequential simulation, generalized linear models
Target group	All employees, MPS project partners
Availability	Open
Project	Multipoint
Project number	808002
Research field	
Number of pages	22
© Copyright	Norwegian Computing Center

Contents

1	Introduction	7
2	Markov mesh models	8
3	Model specification	8
3.1	Specification of the neighborhood functions	9
3.1.1	2D specification.	9
3.1.2	3D specification.	11
4	Volume fraction.	12
5	Examples	13
5.1	2D binary models.	13
5.2	Comparison with snesim	13
5.2.1	Effect of volume fraction steering	16
5.3	2D trinary models	16
5.4	3D model	18
6	Conclusions	18
A	Generalized linear models	20
B	Parameter reduction.	20
	References	21

1 Introduction

Reservoir models are commonly described by a two step approach by first defining the geometry of the facies and then populate this model with petrophysical properties, see Damsleth et al. (1992). Simulation studies show that the facies model often is one of the main sources of variability in flow, see for instance Skorstad et al. (2005). The spatial distribution of facies is therefore a crucial part of any reservoir model. The use of multi-point statistics for geological facies modeling was proposed nearly two decades ago (Guardiano and Srivastava, 1993). Since then several methods have been developed and tested. At large there has been two paths of development. The statistical model driven approach and the algorithmic approach. Markov random fields (Tjelmeland and Besag, 1998), has been the preferred statistical model. The problem with these models is that they are highly time consuming, both in terms of model estimation and simulation. The algorithmically driven development has the goal to formulate a simulation procedure that reproduces pattern at a limited template. This approach had a break through by the introduction of search trees (Strebelle, 2000). These methods have been criticized for their lack of consistency since the statistical model depends on the simulation path. A more serious concern with the algorithmic approach is however the strong dependence to pattern frequencies in the training image; there is not a problem with patterns that are seen in the training image, but how the method treats patterns that are not present in the training image. Current practice is to reduce the size of the pattern, but there is obviously room for more advanced approaches. In this respect, methods using statistical models, have an advantage over the algorithmic methods. The model interpolate between observed patterns to compute the probability of patterns that are not present in the training image.

We propose to model the facies dependencies through a Markov mesh model (Abend et al., 1965). This is a sub class of Markov random fields which is defined through a unilateral path (Daly, 2005). The probability model is defined using the framework of generalized linear models, see McCullagh and Nelder (1989). This type of model is also discussed in Cressie and Davidson (1998), but we are explicit in the formulation of the model and extend it to three dimensions. The formulation enables fast estimation of model parameters through iterated weighted least squares, and fast simulation by the sequential definition. The parametrization is suited to model phenomena with a high degree of spatial continuity, such as for facies structures. It captures the consistency of the modeling approach, the speed of the algorithmic approach during simulation, and is memory efficient.

This work focus on how the statistical model can be formulated and how to simulate unconditional realizations, the challenge of conditional realizations is discussed in Kjønsgberg and Kolbjørnsen (2008).

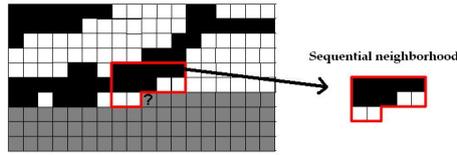


Figure 1. The illustration displays a snapshot of an unfinished simulation, where the grey cells have not yet been simulated. The sequential neighborhood is represented by the cells within the red border.

2 Markov mesh models

Markov mesh models are defined by a unilateral path and a conditional probability for the value in a cell given the cell values in a sequential neighborhood. The sequential neighborhood is a subset of previously simulated cells, illustrated in Figure 1 on a two dimensional grid.

Consider a finite, regular grid in two or higher dimensions, and let the one-dimensional index i label the cells of the grid. The set of all cells is $\{1, 2, \dots, N\}$, where cell value x_i can take K different facies values, $x_i \in (1, K)$. We write the conditional probability for facies at the cell i as

$$\pi(x_i | x_{j < i}) = \pi(x_i | x_{\Gamma_i}), \quad (1)$$

where x_{Γ_i} is the set of values in the sequential neighborhood of cell i . Markov mesh models are fully specified through the conditional probabilities in (1), such that the joint probability is

$$\pi(x_1, x_2, \dots, x_N) = \prod_{i=1}^N \pi(x_i | x_{\Gamma_i}). \quad (2)$$

Simulation from the model is done by following the unilateral path, $i = 1, 2, \dots, N$, throughout the grid. For each cell the facies value is drawn according to the conditional probability $\pi(x_i | x_{\Gamma_i})$. Each cell is visited once, and the resulting grid configuration follows the joint probability distribution in (2).

3 Model specification

The statistical model is defined by parameterizing the conditional probabilities in expression (1). Our model is based on generalized linear models (GLM), see McCullagh and Nelder (1989). The formulation is chosen such that the parameters are efficient to estimate and simple to interpret. We do not claim that our model is unique in this respect, there exist many alternative model formulations with similar characteristics.

The idea in GLM is that the distribution of a response variable depends on a linear combination of explanatory variables, through a non-linear link function. In our application

we let the facies x_i be the response variable, and the explanatory variables be functions of the sequential neighborhood. The facies x_i is one of K categories, and is encoded with binary variables x_i^k , such that $x_i^k = 1$ if $x_i = k$ and 0 otherwise. Further we let \vec{z} be a $(P + 1) \times 1$ vector of explanatory variables, with elements that are functions of cells from the sequential neighborhood. We propose particular functions below, but for now we write $z_{ij} = f_j(x_{\Gamma_i})$ for some $j \in (1, P)$. The conditional probability in expression (1) is then:

$$\pi(x_i | \vec{z}_i, \vec{\theta}^1, \dots, \vec{\theta}^P) = \frac{\prod_k \exp\{\vec{z}_i^T \vec{\theta}^k x_i^k\}}{\sum_{j=1}^K \exp\{\vec{z}_i^T \vec{\theta}^j\}}.$$

The joint probability in (2) is further:

$$\pi(x_1, \dots, x_N) = \prod_{i=1}^N \frac{\prod_k \exp\{\vec{z}_i^T \vec{\theta}^k x_i^k\}}{\sum_{j=1}^K \exp\{\vec{z}_i^T \vec{\theta}^j\}}. \quad (3)$$

Interpreted as a likelihood for the model parameters this expression is a GLM. The maximum likelihood estimation of the parameters in the Markov mesh formulation can therefore be solved with the iterative weighted least squares scheme. Details of our implementation is found in appendix A.

Although (3) is identical to the likelihood of a GLM, the assumptions that lead to them are very different. In GLM the assumption is independence whereas in Markov mesh we use a sequential formulation. Thus even though the maximum likelihood estimate is identical, other properties of the estimators in GLM does not hold in general. In our application to facies modeling this has an unfortunate effect on the volume fraction. We account for this by a post-processing the parameter estimates explained in section 4.

3.1 Specification of the neighborhood functions

We specify the neighborhood functions, $f_j(x_{\Gamma_i})$ for $j = 1, \dots, P$ for a 2D model initially, and extend this model to 3D subsequently. The continuity of facies and transitions to other facies are the most important features of geological structures, our model description therefore focus on these features.

A multi-point interaction of order l is the interaction between the reference cell and a function of the values of $l - 1$ cells in its sequential neighborhood. A two-point interaction thereby refers to the interaction between the reference cell and the value of one cell in its sequential neighborhood.

3.1.1 2D specification

The two-point interactions are the simplest, and we include all two-point interactions in a subset of the sequential neighborhood in our model. Figure 2 shows how this can be described by the two lengths l_x and l_y to give interactions with $2 \cdot l_x \cdot l_y + l_x + l_y$ cells. For each cell j among these we include an indicator function $f^k(x_j)$ for every facies k , the function is 1 if cell j has facies k and 0 otherwise. This yields one parameter for each facies value of each cell in the sequential neighborhood, resulting in $K(2 \cdot l_x \cdot l_y + l_x + l_y)$ parameters.

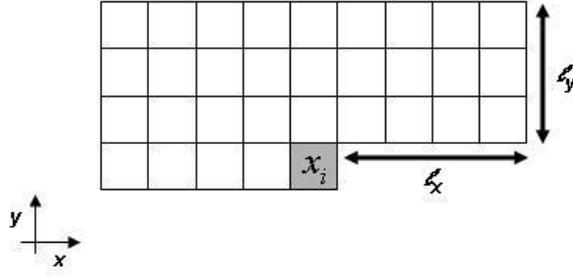


Figure 2. Illustration of the two-point interaction neighborhood. l_x and l_y yield the span of the neighbourhood.



Figure 3. All possible combinations of these 5 cells are included in the model.

The challenge with higher-point statistic is that there are too many possibilities. It is not possible to extract all properties since this would give the problem with missing patterns, as is seen in the traditional snesim approach. However, at the same time all relevant statistics should be considered. The impact of the four nearest cells γ_i^4 , illustrated in Figure 3, are the most important, therefore all combinations of these are considered resulting in K^5 parameters.

For multi-point interactions at longer range, we focus on continuity and transitions of facies. Therefore we include multi-point patterns where all cells have identical facies. The set of cells in these patterns are chosen carefully to capture the shape and extension of the facies object. We limit our selections to a set of directions, see Figure 4. In each of these directions we first include the interaction with the two nearest cells, which are three-point interaction terms. Then we increase with one cell at the time until we reach a limit L . Illustration of these interaction terms are given in Figure 5. Let $x_{\gamma_i}^{l-1}$ be a set of $l-1$ neighbors in an l -point interaction term, then the indicator function are expressed as

$$f^k(x_{\gamma_i}^{l-1}) = \begin{cases} 1 & \text{if all } x_j \in x_{\gamma_i}^{l-1} = k, \\ 0 & \text{otherwise.} \end{cases}$$

For each facies and direction one such indicator function is included in our model up to the highest interaction term of length L , this result in $8 \cdot (L-2)K$ terms.

Our explicit parametrization is easy to interpret; two-point statistics measure direct dependency between cells; the nearest cells indicate preferences for particular structures at the minimum scale; and the indicators of facies continuity promote continuity if the parameter corresponding to the same facies is large or it promote transition if the parameter corresponding to an other facies is large.

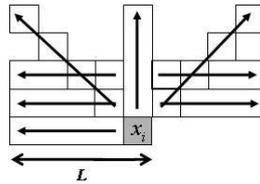


Figure 4. Illustration of the strips of cells where higher-point interactions are considered. Arrows indicates the directions and in which order the number of interaction terms increases.

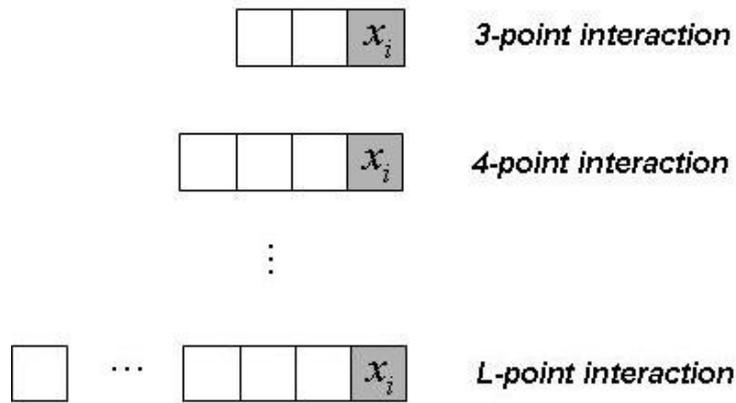


Figure 5. Example of higher-point interactions that are included in one of a set of directions.

Some of the functions described are redundant, for instance it is not necessary to include two point interactions with the nearest four cells since these are covered when all configurations are considered. An other situation which might occur is that a multi-point pattern described in one of the explanatory variables, is not present in the training image; for instance all combinations of the four closest cells need not be present, or all facies need not be continuous for all lengths included. Problems related to these aspects are solved by introducing parameter reduction in the estimation, see appendix A.

3.1.2 3D specification

In three dimensions the number of cells in the dependency structure increases significantly. It is far more challenging to capture the main features of the facies structures whilst keeping a low number of interaction terms.

The advantage with sequential simulation is that all cells within the sequential neighborhood are simulated. This is in contrast to snesim which has unestablished patterns in its template. In the snesim approach the template therefore need to be full in order to account for all possible combinations. In our approach we take advantage of the large redundancy of information in the sequential neighborhood. This means that we can ignore some cells and still have good information about the neighborhood. By systematically selecting the cells, we are able to keep the number of parameters from exploding.

In our selection we consider those cells located in the 2D orthogonal slices intersecting in the reference cell i . These are cells from layers above the current layer, and cells from earlier in the path in the current layer. For each of the three 2D slices we adopt the 2D model

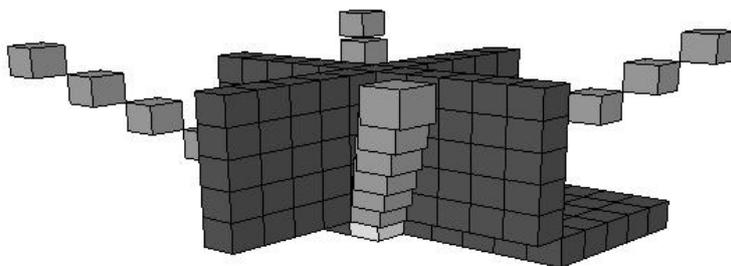


Figure 6. An illustration of which cell that are included in the 3D model specification. These are cells from three 2D slices and from four diagonal lines of cells.

above, that is, we include a similar set of interaction terms for each slice. In addition we extend the neighborhood beyond these 2D slices by considering cells going diagonally out in all three directions. We include four such diagonal lines of cells from which we add the same set of interaction terms as illustrated in Figure 5. A 3D illustration of the neighborhood is displayed in Figure 6.

In geological structures there is much similarity between consecutive layers in all directions. Therefore, we believe our choice of neighborhood to be sufficient enough to capture the most important correlations.

An advantage with the 2D cross-section of neighboring cells is that it yields simpler interpretation of the parameters since we can relate to 2D patterns.

4 Volume fraction

Reproduction of the correction facies fraction from the training image is of great importance for our application. This is not a problem in standard GLM, as long as the explanatory variables are generated from the same distribution as was used in the estimation. However, since our model is defined sequentially, the response variable will be an explanatory variable when we move along the unilateral path. The explanatory variables in our model are therefore not from the distribution used for estimation, unless we are able to fully reproduce all statistics of the training image.

To reproduce the volume fraction, we adjust the estimated parameters to meet our requirement. Advantageously, the model formulation enables us to select parameters that corresponds to continuity of facies as discussed above.

We propose an iterative method where the volume fraction of a realization is measured, and if the value is deviating from the training image the selected parameters are adjusted by a small value. A new realization is generated based on the adjusted parameter values, the volume fraction is computed and the parameters are again adjusted if necessary. This process is continued until the volume fraction of the realization is sufficiently close to the training image.

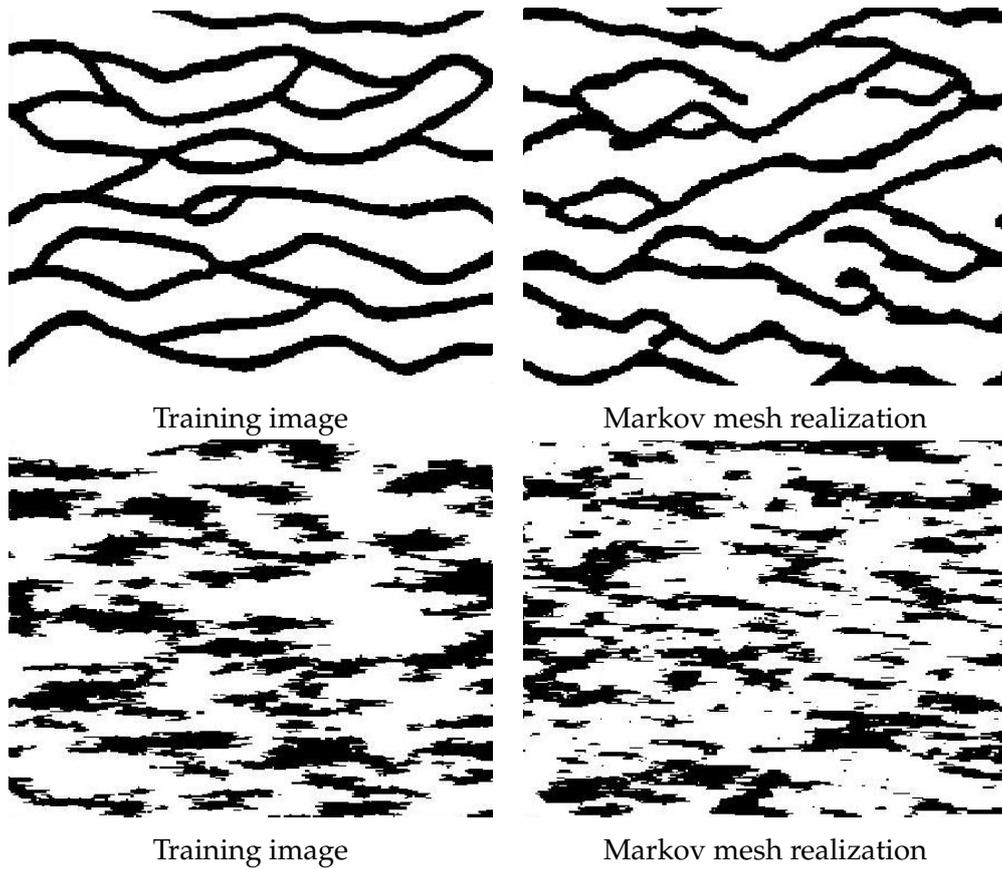


Figure 7. 2D training images and realizations from our 2D model.

5 Examples

First we consider two 2D binary training images, where one of them is used for a more thorough comparison to the snesim approach. This is done in terms of visual inspection, statistical analysis of properties within the realizations, and effect of volume fraction steering. Next we consider a 2D trinary and a 3D binary training image, and these are visually compared to snesim.

5.1 2D binary models

Figure 7 displays two training images and the corresponding realization from the 2D Markov mesh model. We have used a two-point interaction neighborhood of extension $l_x = l_y = 5$ and higher point interactions of maximum size $L = 8$ in all directions.

By visual inspection of the results it is clear that the model reproduces similar features as in the training images, although they are generally more rugged. The simulation direction is revealed by the slight skewness of the pattern in the realizations.

5.2 Comparison with snesim

In the snesim algorithm we define the template size to be 60, and set the servo system factor to fully reproduce the volume fraction, see Liu (2006). The Markov mesh model is

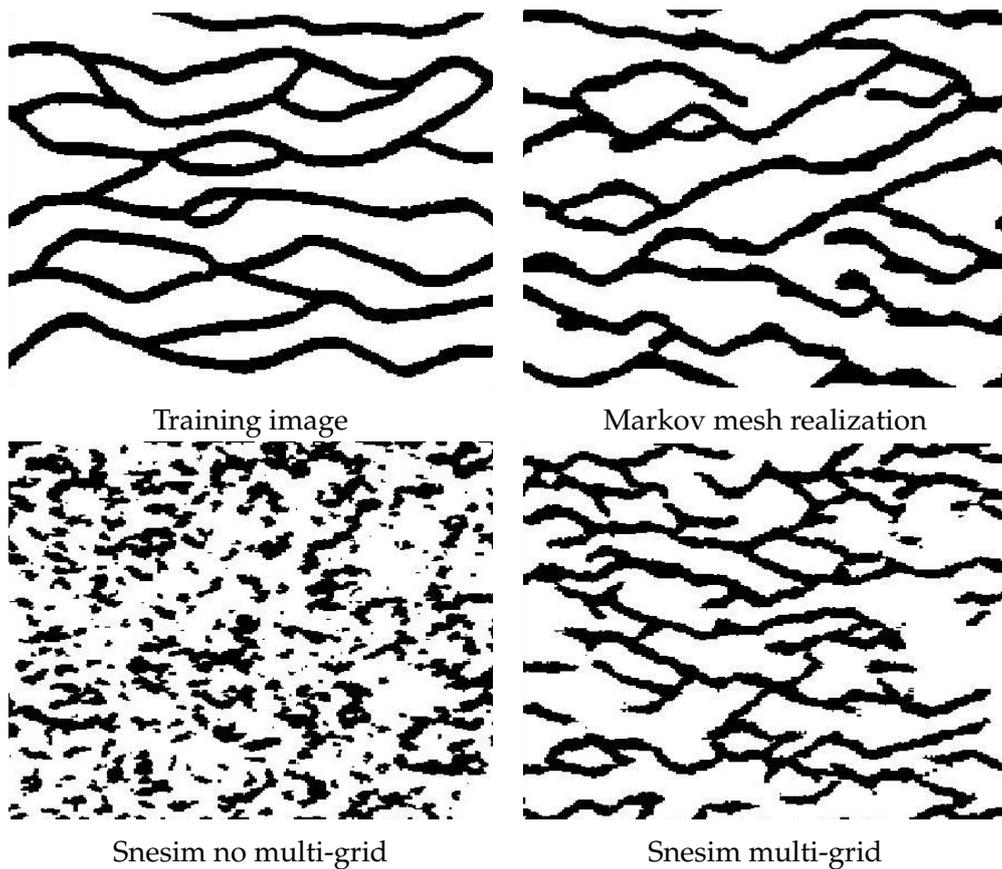


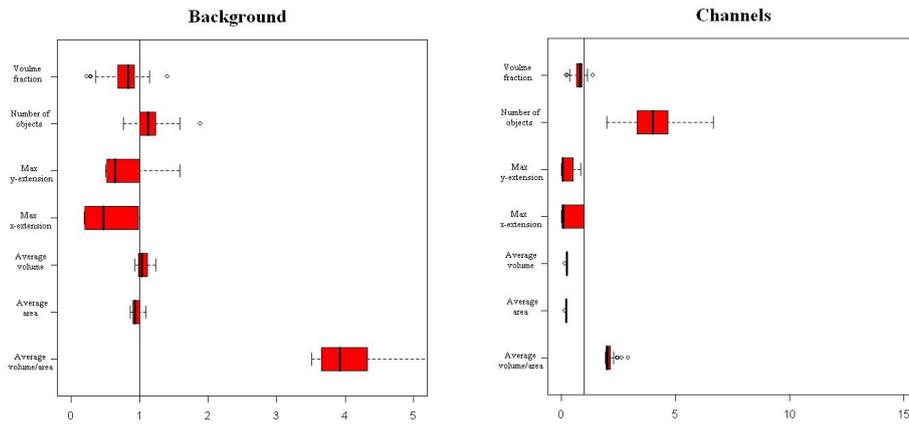
Figure 8. 2D binary training image with realizations from the Markov mesh model and the snesim algorithm.

defined on the original grid. It is therefore natural to compare it to the snesim defined on the same grid. It is however common to use a multi-grid approach with the snesim algorithm, NBNBref multigrid, so we include results using three multi-grids.

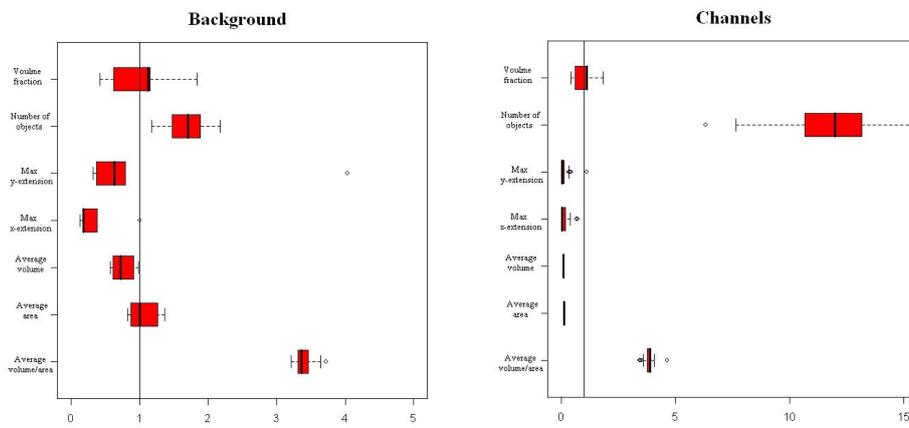
Figure 8 shows the results (a) the training image, (b) a realization from our model, (c) a snesim realization on one grid and (d) a snesim realization using three multi-grids. We clearly see the need for multi-grid in the snesim approach. The figure also suggest that the use of multi-grids in the Markov mesh formulation might improve the results even further.

To check how the models reproduce the correct statistics we focus on features in the realizations. Important features of geological structures that should be reproduced are the range of the dependency, the size and shape of objects, the number of objects and the volume fraction. In Soleng et al. (2006) a program is described which computes these statistics from realizations. The program detects the various facies objects and computes their volumes, surface areas and their extensions in each direction.

We apply the facies properties program in order to compare our model with the multi-grid snesim approach. It does not make sense to compare to snesim on one grid. We run the program on the training image, and on 100 realizations from both methods. Statistics are displayed in form of box-plots in Figure 9. The straight vertical line represents the



Markov mesh realizations



Snesim realizations

Figure 9. Statistical analysis of Markov mesh and snesim realizations

training image, and the boxes span out data from the realizations. The leftmost box-plots represent the background facies and the rightmost represents the channel facies.

The number of channel objects is higher for snesim realizations because of all the loose ends. Consequently, the number of background objects increases and the extensions, average area and average volume of the objects decreases. The latter measure in the box plot is the average of the volume divided by area for each object. This gives an indication of the smoothness of the edges of the objects. Channel objects from the Markov mesh model are slightly smoother than from snesim. The difference between the models for the background facies is much smaller, though the plot gives a different impression because of the difference in scale.

The general impression left by the comparison is that the distributions for the Markov mesh model more often encapsulate the true value than the snesim realizations.

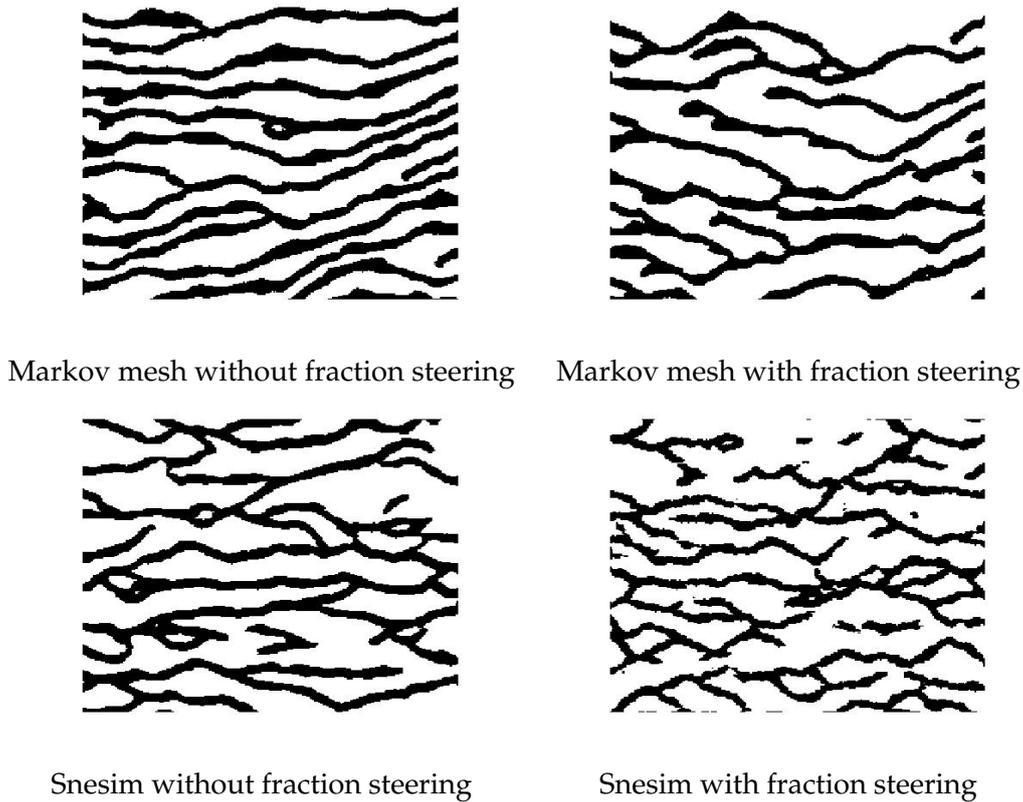


Figure 10. Markov mesh and Snesim realizations with and without volume fraction steering.

5.2.1 Effect of volume fraction steering

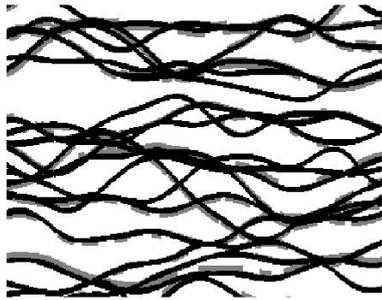
We run both our model and the snesim algorithm with and without volume fraction steering to see its effect. We use the training image with channels from Figure 7, which has a 0.28 fraction of channels. Results are displayed in Figure 10

The volume fraction before steering is 0.37 for the Markov mesh model and 0.33 for snesim. After steering it is 0.29 for both methods. Note that for this particular training image snesim fits the volume fraction quite well without steering.

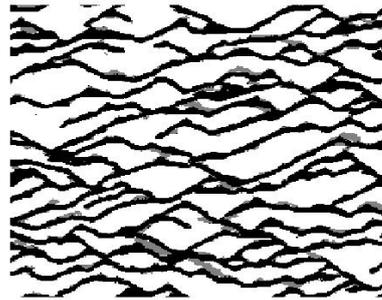
When the Markov mesh model does reproduce the facies fraction, we adjust the parameters based on their interpretation, and fit a model that yields more background facies without losing the continuity of the channels. For snesim, the volume fraction steering is not concerned with the geological properties of the training image, only the global and local facies fractions during the simulation. This results in the many loose end channels that appear in the realizations.

5.3 2D trinary models

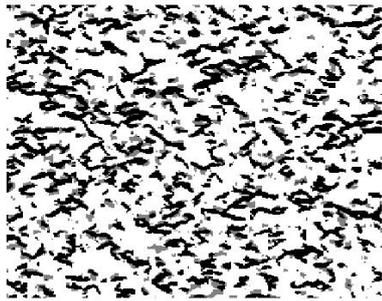
Here we give an example of a training image with 3 facies, background, channels and crevasses. Again we display results of snesim with and without multi-grid, see Figure 11. We use a template of size 60 and 3 multi-grids for snesim. For the Markov mesh model we use a two-point interaction neighborhood $l_x = 7$ and $l_y = 4$. The higher point interaction neighborhood is set to length $L = 7$ for all directions.



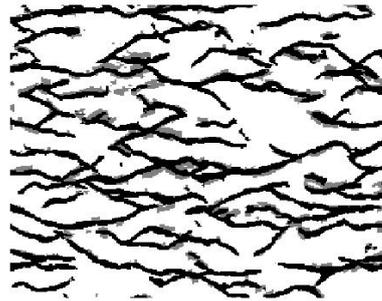
Training image



Markov mesh realization



Snesim no multigrid



Snesim multigrid

Figure 11. 2D trinary training image with realization from the Markov mesh model and the Snesim algorithm.

The Markov mesh model produces realizations with properties that appears to be more smooth than the snesim algorithm with multi-grid. Note that we display only parts of the whole training image and the realizations to highlight the structure. The original training image is four times the size of the figure displayed.

5.4 3D model

In three dimensions we use a training image with channels, displayed by three cross-sections in Figure 12. This training image is generated by an object model. Parameter choice for the model is set to $l_x = l_y = 4$ for the two-point interaction neighborhood and a length $L = 6$ for all directions of the higher-point interaction neighborhood. For snesim we use a template of size 80 with equal radius in all directions, and 3 multi-grids.

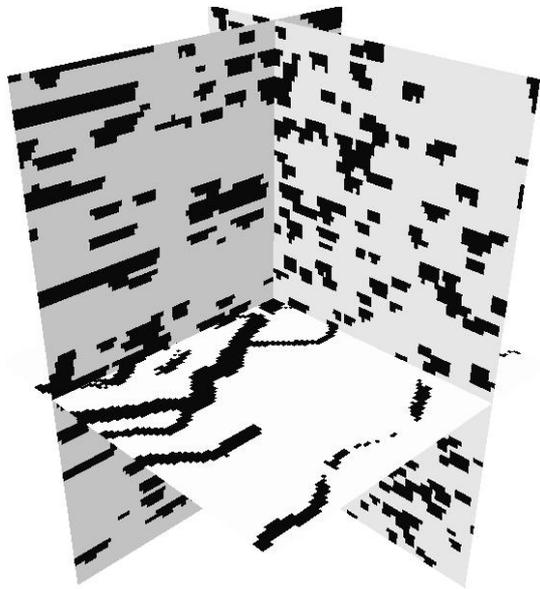
Simulation results are promising, see Figure 12, but a more thorough analysis indicate that the model has problem preserving the continuity of the channels. However, compared to snesim the shape and continuity of the channel objects seems slightly improved.

6 Conclusions

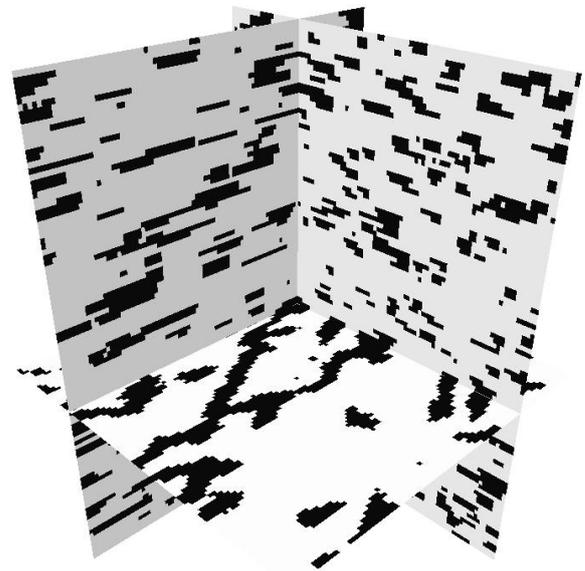
We have developed a Markov mesh model by using the framework of GLM. This approach yields a consistent model that enables efficient parameter estimation and fast simulation. The parametrization of the model is chosen to capture the continuity of geological structures. We have further adapted the approach such that it preserves the volume fractions of facies by increasing the probability of facies continuity.

Test examples show good pattern reproduction. In 2D the model capture the curvilinearity and continuity of the channel objects and at the same time reproduce the correct volume fraction. In one test case we observe skewness in the simulation pattern which might be caused by the sequential simulation path. In 3D, the model also perform reasonable, but it tends to make too small objects.

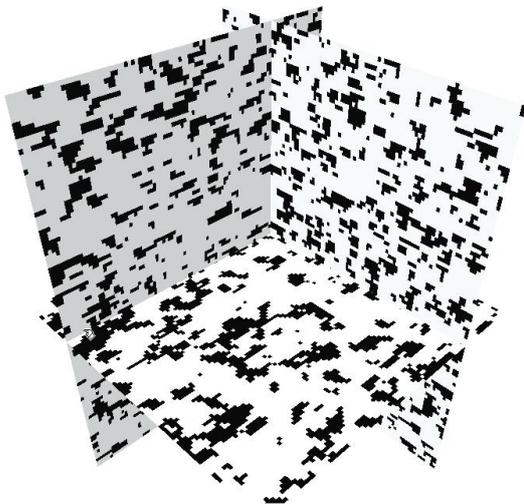
When compared to the snesim approach, our model yields substantially better results when defined in the same grid. The results obtained are comparable with those obtained by snesim using multi-grids. In a test case our approach gives a better reproduction of the statistics of the training image than what is observed in snesim using multi-grids.



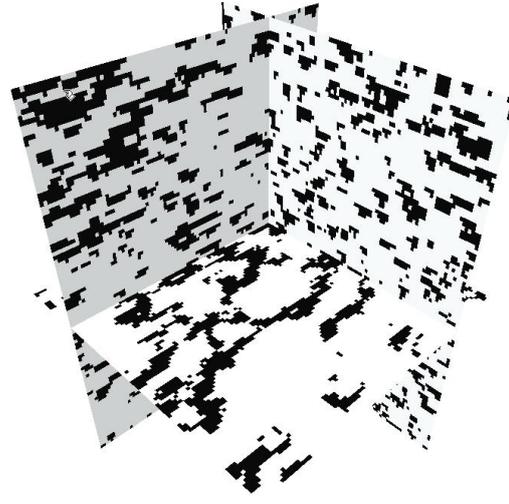
Training image



Markov mesh realization



Snesim no multi-grid



Snesim multi-grid

Figure 12. 3D binary training image with realization from the Markov mesh model and the snesim algorithm.

A Generalized linear models

In standard generalized linear models we assume that we have N independent observations of the model and get the likelihood function

$$L(\vec{\theta}^1, \dots, \vec{\theta}^K; \vec{Z}, \vec{X}) = \prod_{i=1}^N \frac{\prod_k \exp\{\vec{z}_i^T \vec{\theta}^k x_i^k\}}{\sum_{j=1}^K \exp\{\vec{z}_i^T \vec{\theta}^j\}} \quad (\text{A.1})$$

where \vec{X} is a $N \times K$ matrix of observations where the i^{th} row contains 0-1 coding for the i^{th} observation, and \vec{Z} is the design matrix of dimensions $N \times (P + 1)$ where i^{th} row is the set of explanatory variables corresponding to the i^{th} observation.

Standard derivations yield the following system of equations to be solved

$$\vec{Z}^T \vec{x}^k = \vec{Z}^T \vec{\mu}^k(\vec{\theta}^1, \dots, \vec{\theta}^K), \text{ for } k = 1, \dots, K,$$

where the vector $\vec{\mu}^k(\vec{\theta}^1, \dots, \vec{\theta}^K)$ is the $N \times 1$ vector of $\mu_i^k(\vec{\theta}^1, \dots, \vec{\theta}^K)$, while $\vec{x}^k = [x_1^k, x_2^k, \dots, x_N^k]^T$. The above expressions are solved using a gradient search, which yields the iterated weighted least squares where each iteration is given by

$$\vec{\theta}_{m+1}^k = \vec{\theta}_m^k + (\vec{Z}^T \vec{W}_m^k \vec{Z})^{-1} \vec{Z}^T (\vec{x}^k - \vec{\mu}^k(\vec{\theta}^1, \dots, \vec{\theta}^K))$$

for $k = 1, \dots, K$. Here \vec{W}_m^k is an $N \times N$ matrix of weights given by:

$$\vec{W}_m^k = \text{diag}\{(1 - \mu_1^k(\vec{\theta}_m))\mu_1^k(\vec{\theta}_m), \dots, (1 - \mu_N^k(\vec{\theta}_m))\mu_N^k(\vec{\theta}_m)\}.$$

This is the exact same equations which are used in our approach.

B Parameter reduction

We reduce the number of parameters by extracting the principal components of the explanatory variables. This also solves the problems that occur when the design matrix is singular.

In principal component analysis we find the eigenvalues and eigenvectors of the empirical covariance matrix through the equality

$$\vec{V} \vec{D} \vec{V}^T = \vec{Z}^T \vec{Z},$$

where \vec{V} is a $P \times P$ matrix of eigenvectors and \vec{D} is a $P \times P$ diagonal matrix of eigenvalues. We select the $\tilde{P} < P$ eigenvectors $\tilde{\vec{V}}$ corresponding to the \tilde{P} largest eigenvalues, and use these to generate the reduced $N \times \tilde{P}$ matrix $\tilde{\vec{Z}}$ of principal components

$$\tilde{\vec{Z}} = \vec{Z} \tilde{\vec{V}}.$$

From this set of reduced variables we estimate the corresponding parameters $\{\tilde{\theta}^1, \dots, \tilde{\theta}^K\}$. If we want a set of parameters which apply to the original variables, we find this by defining

$$\vec{\theta}^k = \vec{V} \tilde{\theta}^k.$$

It is equivalent to compute $\tilde{Z} \tilde{\theta}^k$ or $\vec{Z} \vec{\theta}^k$ since we have the relation:

$$\tilde{Z} \tilde{\theta}^k = \vec{Z} \vec{V} \tilde{\theta}^k = \vec{Z} \vec{\theta}^k.$$

References

- Abend, K., Harley, T., and Kanal, L. (1965). Classification of binary random patterns. *IEEE Transactions on Information Theory*, IT-11:538–544.
- Cressie, N. and Davidson, J. (1998). Image analysis with partially ordered markov models. *Computational Statistics and Data Analysis*, 29(1):1–26.
- Daly, C. (2005). *Higher Order Models using Entropy, Markov Random Fields and Sequential Simulation*, volume 1, pages 215–224. Springer.
- Damsleth, E., Tjølsen, C., Omre, H., and Haldorsen, H. (1992). A two-stage stochastic model applied to a north sea reservoir. *SPE JPT*, 44(4):402–408.
- Guardiano and Srivastava (1993). *Multivariate Geostatistics: Beyond Bivariate Moments*, volume 1, pages 133–144. Kluwer, Dordrecht.
- Kjønsberg, H. and Kolbjørnsen, O. (2008). *Markov Mesh Simulations with Data Conditioning through Indicator Kriging*, volume 1, pages 257–266. Kluwer, Dordrecht.
- Liu, Y. (2006). Using the snesim program for multiple-point statistical simulation. *Computer and Geosciences*, 32:1544–1563.
- McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Chapman and Hall, 2 edition.
- Skorstad, A., Kolbjørnsen, O., Fjellvoll, B., and Howell, J. (2005). *Sensitivity of oil production to petrophysical heterogeneities*. Springer, Dordrecht, The Netherlands.
- Soleng, H., Syversveen, A., and Kolbjørnsen, O. (2006). Comparing facies realizations: Defining metrics on realization space. *Proceedings of the 10th European Conference in Mathematics of Oil Recovery*, ??:A014.
- Strebelle, S. (2000). *Multivariate Geostatistics: Beyond Bivariate Moments*, volume 1, page ??? ??
- Tjelmeland, H. and Besag, J. (1998). Markov random fields with higher-order interactions. *Scandinavian Journal of Statistics*, 25:415–433.

