Synthesis and simulation of digital images by tree-indexed Markov chains

F. Michel Dekking[1], Cor Kraaikamp, Amro M. Elfeki[2], Johannes Bruining
Faculty of Information Technology and Systems, Faculty of Applied Earth Sciences,
Delft University of Technology, P.O. Box 5028, Delft, The Netherlands

ABSTRACT

Quad trees and dyadic trees are hierarchical data structures that are used to represent spatial data or images. With a collection of images, a tree-indexed Markov chain can be generated by letting the transition probabilities from one level to the next level be estimated by the empirical transition probabilities. Depending on the original collection of images various goals may be achieved. One may for instance synthesize new structures from given structures. Another example is the modeling of heterogeneous structures at multi-scales for geological characterization.

Keywords: Synthesis of digital images, tree-indexed Markov chain, Simulation of images, multi-scale.

1. INTRODUCTION

Quad trees and dyadic trees are hierarchical data structures used to represent spatial data or images. They are based on a principle of recursive decomposition of an image into its corresponding scales. Each level in the hierarchical structure corresponds to a particular scale and each node at a given scale is connected to a node at the preceding coarser scale and to several descendent nodes at the next finer scale. This type of representation is commonly used in many fields such as computer science and data compression [4]. The structure of the paper will be as follows. First an introduction to the quad trees are given followed by an introduction to the tree-indexed Markov chains on quad trees. Some simulations are presented using various models on the quad trees. Then an introduction to the dyadic trees is given with some simulations and comparisons with the quad tree simulations.

2. QUAD TREES

A two-dimensional image can be represented by K scales (levels). At a particular level, L_M, such that $0\leq M \leq K$, the corresponding number of nodes (pixels) at this scale is $2^M\times2^M$. There is a factor 4 between the number of grid cells at each scale and the previous coarser one. This yields the quad tree structure over all scales of an image. The procedure is simply based on the successive subdivision of an image into four equal-sized quadrants. In case of binary images, which contains either black (B) or white (W) colors, if the image does not consist entirely of blacks or entirely whites, it is subdivided into quadrants colored gray (G), sub-quadrants, and so on, until quadrants are obtained that consist entirely of blacks or whites. Figure 1 shows a resolution of an image by the quad tree. In general, a tree is a connected graph without loops or cycles and with a distinguished vertex. The vertex that is the first node in the tree is called the *" root"*. It is denoted by the symbol Λ and corresponds to the entire image (M=0). The descendents of a node are the *" children"*. Each child represents a quadrant of the region that is represented by its father.

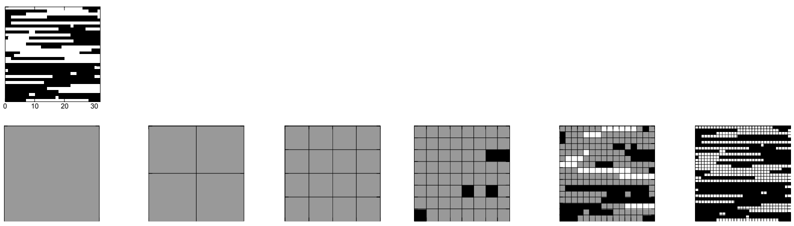


Figure 1. Scale resolution of an image with 32 by 32 pixels (K=5, N=32, left top of the figure is the image, bottom row is scales M = 0, 1, 2, 3, 4 and 5 respectively from left to right)

[1] Correspondence E-mail: F.M. Dekking@math.tudelft.nl
[2] On leave from Civil Engineering Dept., Faculty of Engineering Mansoura University, Mansoura, Egypt.

3. TREE-INDEXED MARKOV CHAINS

Images can be randomized by randomly labeling the corresponding quad trees. A natural way to accomplish this is by using a Markov chain. A Markov chain on a tree describes a scale-to-scale transition. Formally this should be called a tree-indexed Markov chain. For vertices u and v on the tree, one can write $u \le v$ if u is on the unique path from v to the root Λ. The set of vertices of the tree will be denoted by T. For any vertex w in T which is not the root Λ, we denote the father of w by $\uparrow w$, i.e., the unique vertex connected to w with $\uparrow w \le w$. A tree-indexed Markov chain (indexed by a tree T) is a collection $\{X_w\colon w \in T\}$ of random variables taking values in a finite set Γ of states, satisfying the (tree) Markov property (see [1]). Let $w \in T$, $w \ne \Lambda$, and let $v = \uparrow w$. We call

$$p_\Lambda(\alpha) = P(X_\Lambda = \alpha) \qquad \text{and} \qquad p_{v,w}(\alpha,\beta) = P(X_w = \beta \mid X_v = \alpha), \quad \alpha,\beta \in \Gamma,$$

the initial distribution and the transition probabilities of the chain.

In Dekking et al.[1] it was shown that for a tree-indexed Markov chain X_T, and for $v \in T$, the probability $P(X_v = Ʊ)$ can be expressed in the initial distribution and transition probabilities, proceeding just as in the case of ordinary Markov chains. To be explicit: for each $v \in T$ there is a unique path of vertices $\Lambda, v_1, v_2, \ldots, v_{n-1}, v$ from the root Λ to v and

$$P(X_v = \alpha) = \sum_{\alpha_0 \in \Gamma} \sum_{\beta_1 \in \Gamma} \cdots \sum_{\beta_{n-1} \in \Gamma} p_\Lambda(\alpha_0) p_{\Lambda,v_1}(\alpha_0,\beta_1) \ldots p_{v_{n-1},v}(\beta_{n-1},\alpha). \tag{1}$$

For an arbitrary T we define its levels $L_M = L_M(T)$, by $L_0 = \{\Lambda\}$ and $L_{M+1} = \{w \in T\colon \uparrow w \in L_M\}$ for $M = 0,1,2,\ldots,K$. If $w \in L_M$ we write Lev $(w) = M$. With an $2^K \times 2^K$ pixels we associate the 4-ary tree $T_4(K)$, i.e., all level K vertices do not have children, and for each v with $\mathrm{Lev}(v) < K$ one has $\#\{w\colon \uparrow w = v\} = 4$ (see Figure 2).

To randomize the quad tree of a $2^K \times 2^K$ pixels image by a $T_4(K)$–indexed Markov chain we already saw that we should have a state space $\triangleq = \{B, W, G\}$ representing the colors white, black and gray. Furthermore it is required that $p_{v,w}(W, W) = p_{v,w}(B, B) = 1$, for all $T_4(K) \setminus \{\Lambda\}$, $v = \uparrow w$. This corresponds to the fact that the algorithm stops when the pixels in a sub-square are either all white or all black. With this definition the tree itself is not image dependent: the randomness resides in the transitions from G to B, W and G. Although the whole idea of a tree-indexed Markov chain on quad tree is conceptually simple and clear, some care has to be taken. The problem is that it is possible (since children do not "know" each others colors) that a "gray" father has four black children or four white children, which in violation with the definition of the father being in a state labeled "gray". The way out of this problem is to interpret quadruples of vertices having the same father as a single vertex. This gives a $T_4(K\text{-}1)$–indexed Markov chain with state space $\triangleq^4 = \{B, W, G\}^4$ and transition probabilities,

$$p_{\mathbf{v},\mathbf{w}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = p_{(v_1,v_2,v_3,v_4),(w_1,w_2,w_3,w_4)}(\alpha_1,\alpha_2,\alpha_3,\alpha_4),(\beta_1,\beta_2,\beta_3,\beta_4), \quad \alpha_i,\ \beta_j \in \Gamma. \tag{2}$$

with $\uparrow w_i = v_j$ for some j and $\uparrow v_1 = \uparrow v_2 = \uparrow v_3 = \uparrow v_4$. This model is also known as the 4-4 model. Other variations of the so-called 1-1 model will be discussed in the next sections.

3.1 Models on the Quad Trees

There are various models can be described on the quad tree. The first model is the 1-1 model, which does not fit into the framework of the tree-indexed Markov chain as explained above. The second model is the 1-4 model (which is a particular case of the 4-4 model above), where one looks at the joint distribution of the states in the four children. This means that one looks at how gray at a certain level in the tree is going to a combination of black, white and gray at the next level in a specific order. For example, in Figure 3 (left image) one can notice that G at level M=0 is going to $WBGB$ at the next level M=1 with this ordering from top to bottom and from left to right (i.e. NW, SW, NE, SE). This model guarantees that G goes always to a combination of black and white and not to pure black or to pure white.

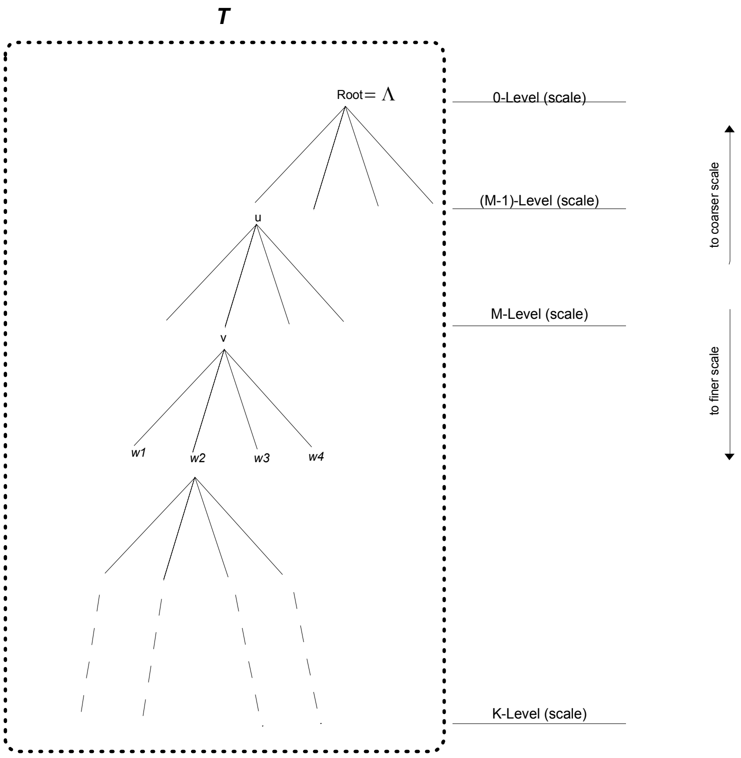


Figure 2. Definition of the scale structure of the quad tree and the symbols used in the text (each node in the tree has four rays).

3.2 Estimation of the Empirical Transition Probabilities from Data

The transition probabilities, that are required to perform simulations, are estimated from the data. Suppose there are M_{tot} images and the corresponding tree of each image is built. To be more precise, these M_{tot} data trees are realizations of a tree-indexed Markov chain X_T with transition probabilities $p_{v,w}$ (Ʊ, …) for $w \in T$ and Ʊ, … ∈ ≜. Here it is always assumed that $v = \uparrow w$. Let $w \in T$ and $1 \leq m \leq M_{tot}$ the color of vertex w in the mth data tree be denoted by X_w^m. One can define for Ʊ, … ∈ ≜, $w \in T$

$$N_w(\alpha) = \#\{X_w^m = \alpha\}, \quad N_{v,w}(\alpha,\beta) = \#\{X_v^m = \alpha, \;\; X_w^m = \beta\}. \tag{3}$$

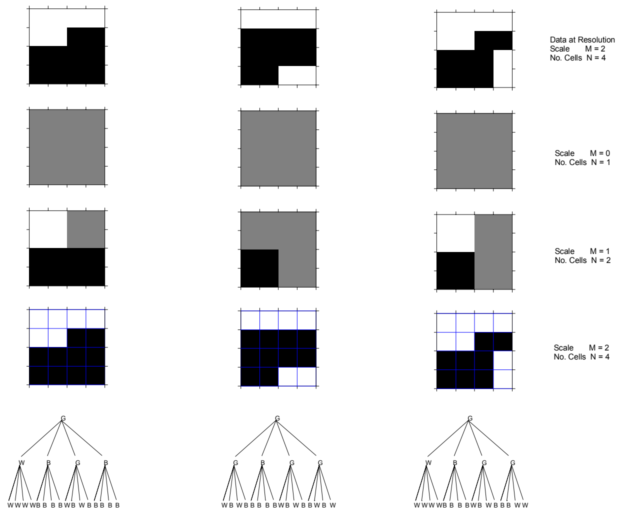
Furthermore, one can define the empirical transition probabilities $\hat{p}_{v,w}$ (Ʊ, …) by

$$\begin{aligned} \hat{p}_{v,w}(\alpha,\beta) &= N_{v,w}(\alpha,\beta)/N_v(\alpha), \quad if \quad N_v(\alpha) \neq 0 \\ \hat{p}_{v,w}(\alpha,\beta) &= 0, \qquad\qquad\qquad\qquad\; if \quad N_v(\alpha) = 0 \end{aligned} \tag{4}$$

and the initial empirical probabilities are estimated by,

$$\hat{p}_{\Lambda}(\alpha) = N_{\Lambda}(\alpha)/M_{tot}. \tag{5}$$

Dekking et al.[1] showed that these empirical transition probabilities $\hat{p}_{v,w}(\mho, \ldots)$ and initial probabilities $\hat{p}_{\Lambda}(\mho)$ are in some sense the "right" estimators for the probabilities $p_{v,w}(\mho, \ldots)$ and $p_{\Lambda}(\mho)$ respectively. The main theorem[1] shows that the empirical transition and initial probabilities are the maximum likelihood estimators for $p_{v,w}(\mho, \ldots)$ and $p_{\Lambda}(\mho)$ respectively. Figure 3 shows, for a simple example, how these empirical probabilities are estimated from a given data set. Firstly, for each given image a tree is built based on a scaling by a factor four until the pixel level is reached. Secondly, in each branch in the tree one looks at all possible transition from the given images in the data. For instance, in Figure 3, the first level is gray for all the given images. In the second level, the square is divided into four small sub-squares with *WBGB* in the first left image, *GBGG* in the middle image and *WBGG* in the right image. The transition probability from *G* at level *M*=0 to *WBGB*, *GBGG* or *WBGG* at level *M*=1is 1/3. One property on that tree is the transition from *B* at any level will go to *BBBB* at the next level with probability one. Similarly, we have a probability one for a transition from *W* to *WWWW*.



Transition Matrix Over Scales:

Scale M = 0 ----> 1

	WBGB	*GBGG*	*WBGG*
G	*1/3*	*1/3*	*1/3*

Scale M = 1 ----> 2

	WBWB	*BBBB*	*WWWW*
G	*1*	*0*	*0*
B	*0*	*1*	*0*
W	*0*	*0*	*1*

	WBWB	*BBBB*	*WWWW*
G	*1*	*0*	*0*
B	*0*	*1*	*0*

	BBBB	*WWWW*
B	*1*	*0*
W	*0*	*1*

	BWBW	*BBBB*	*BBWW*
G	*1/2*	*0*	*1/2*
B	*0*	*1*	*0*

Figure 3. Sample images and calculation of their empirical transition probabilies over the various scales in case of quad tree representation (1-4 model).

3.3 Test Examples of the Quad Tree (1-4 Model)

The first test is a simulation of a layered system. These tests do not use the full power of the tree-indexed Markov chain formalism but make the reader understand how the randomness works in the method. The data of this test is given in Figure 4 (top row, first two images from the left). In this example, the input data are two possible layered sequences. One of the data starts from the top with a black layer and the other starts on top with a white layer. The tree representation of this example is gray at all levels until scale *M*=6 which is the image resolution. At the resolution scale of the image there is either *G* goes to *BWBW* or *WBWB*. Each of these combinations has a probability 0.5. The simulation procedure switches between these two possible combinations and produces the realizations in the bottom row of Figure 4 (first two images). The second test is a simulation of a checkerboard system. The data for such test is given in the top row of Figure 4 (third and fourth images). The simulation results, shown in Figure 4 bottom row, produce some correlation in the system that is not present in the original data. The reason is the following. The tree distribution of this type of data is gray at all levels up to the resolution scale of the image where black and white appeared. The transition in this example is that *G* is going to *WBBW* or *BWWB*. Then the simulation switches between these two possibilities each with a probability 0.5. Correlation is produced in the simulations due to the fact that the procedure is sampling from the two data images. The procedure generates the combination *BWWB* as a neighbor of the combination *WBBW*, which leads to some short range correlation.

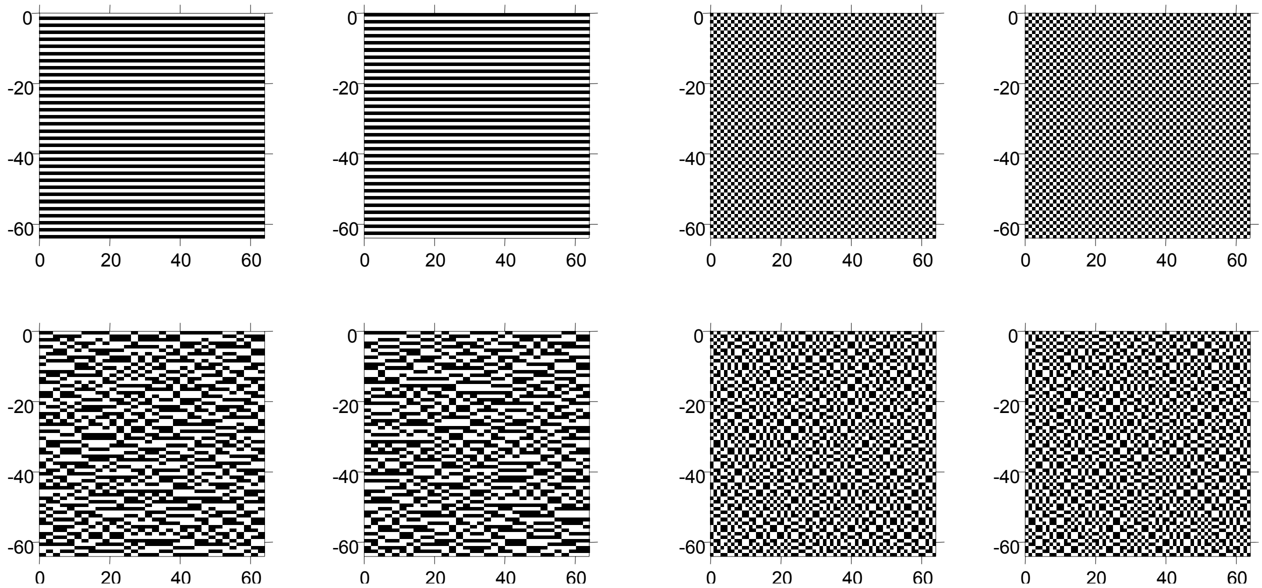


Figure 4. Test Examples of the quad tree: the top row consists of the data, the bottom row consists of the simulations. Left part: layered system, right part: checkerboard system (image resolution *K*=6, *N*=64).

4. SYNTHETIC DATA USED IN THE SIMULATIONS

The four authors were cooperating in a project to model subsurface heterogeneity at various scales. In this context the coupled Markov chain model, developed by Elfeki [2,3] was used to generate synthetic data for the tree-indexed Markov chain technique. A brief description of the coupled Markov chain model is given below. The coupled Markov chain model is a stochastic technique that couples two one-dimensional Markov chains. The first one is used to describe the sequence of variation in states in the vertical direction and the second chain describes the sequence of variation in the horizontal direction. The two chains are coupled in the sense that, a state of a cell in the domain is dependent on the state of two cells, the one on top and the other on the left of the current cell. This dependence is described in terms of transition probabilities from the two chains. Some examples of input data that is generated by the coupled Markov Chain model are shown in the following sections.

Figure 5 (left most column) shows two images that are generated by the coupled Markov chain model. The input parameters (transition probabilities) for generating the bottom image in the first column (from the left) are presented in Table 1 (for the large-scale structure). The elements of the transition probability matrix for the horizontal direction, p^H_{ij} which means the probability of a given state, *i,* is following another state, *j*, are given in the tables (e.g. in Table 1 p^H_{BB} =0.98 and p^H_{BW} =0.02 and so on.). A similar transition probability matrix is used in the vertical direction with p^V_{ij} (e.g. in Table 1 p^V_{BB} =0.80 and p^V_{BW} =0.20 and so on.).

Table 1. Input parameters to generate synthetic data by the coupled Markov chain model (large-scale structure in Figure 5) for the quad tree simulation.

Horizontal transition probability matrix			Vertical transition probability matrix		
State	*B*	*W*	State	*B*	*W*

B	0.98	0.02	*B*	0.80	0.20
W	0.02	0.98	*W*	0.20	0.80

5. SIMULATIONS WITH THE QUAD TREE (1-4 MODEL)

One of the applications of the tree-indexed Markov chain is to handle data at various scales and merge these data in a statistical sense to produce many realizations that possesses information from the given data.

5.1 Simulations with Original and Smoothed Quad Tree

An example of merging two different structures is presented. A large-scale tilted layered system (Figure 5, left bottom image) and a fine-scale structure (Figure 5 left column top image) are considered. The simulations are performed with the original quad tree (described in section 2) and with what is called a smoothed quad tree. The smoothed quad tree is similar to the original quad tree except that it tries to smooth noisy data. The reason to do so is to let the large-scale features to be more pronounced in the simulations. The original idea of the quad tree is that if a square contains any percentage of black and white it is considered to be gray. While in the smoothed quad tree, the idea simply is that if a square contains a higher percentage of black than white then it is considered to be black and vice versa. Two degrees of smoothing are considered. These degrees are the following:

1. Degree of smoothing no. 1: It is considered that, if a square contains more than 87.5% *W* is considered to be *W* and if the square contains more that 87.5% *B,* it is considered to be *B*.
2. Degree of smoothing no. 2: It is considered that, if a square contains more than 75% *W* is considered to be *W* and if the square contains more that 75% *B,* it is considered to be *B*.

According to these degrees of smoothing, simulations have been performed and compared with original quad tree simulations. The simulation with original quad tree is displayed in the Figure 5 (2nd column from the left). The simulations with smoothing effects are displayed in the third and right columns of Figure 5 for the two degrees of smoothing that are mentioned above. One would recognize that the simulations for the degree of smoothing no.1 show no significant difference between this case and the simulation by the original quad tree. The reason is that the degree of smoothing in case no.1 is relatively small that has no significant influence. However, in case of smoothing no.2 (Figure 5 right column) the simulations show more black and white spots than in the simulations with the original quad tree and case no.1. That means the smoothing effect is stronger in case no. 2.

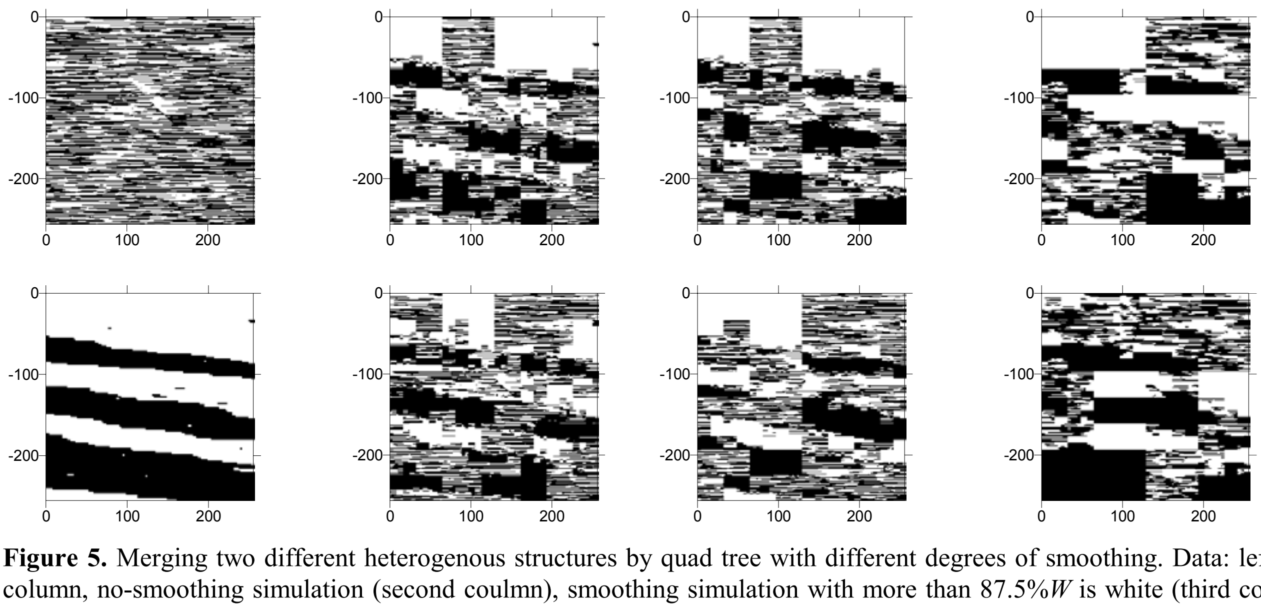


Figure 5. Merging two different heterogenous structures by quad tree with different degrees of smoothing. Data: left most column, no-smoothing simulation (second coulmn), smoothing simulation with more than 87.5%*W* is white (third column), smoothing simulation with more than 75%*W* is white right column. (Image resolution *K*=8, *N*=256).

6. DYADIC TREES

The so-called "*Dyadic trees*" are a variation on the quad trees used in the previous sections. In the dyadic tree, there is a factor of 2 between any scale and the previous coarser one. In this representation any node in the tree has two descendent nodes at the next finer scale and one parent node at the preceding coarser scale. The tree is used to perform separately scaling in the vertical direction (V-scaling) and to perform scaling in the horizontal direction (H-scaling). Although there are many other possibilities we concentrate here on a particular order in which the vertical and horizontal scaling are performed. Firstly, the image is decomposed into its corresponding scales in the vertical direction by a dyadic tree with K_y levels until the pixel level in the vertical direction is reached (we call this the pixel line). Secondly the features that do not appear in the vertical direction will appear when the scaling in the horizontal direction is performed. The strips that are appear in a gray color are scaled horizontally in K_x steps in the horizontal direction until the pixel level is reached. An image that is scaled by the dyadic tree is shown in Figure 6. In the language of section 3 a tree-indexed Markov chain on a dyadic tree is simply a 2-ary tree $T_2(K_x+K_y)$. Mathematically speaking, there is no difference in properties between quad trees and dyadic trees. In particular the empirical transition and initial probabilities are the maximum likelihood estimators for $p_{v,w}(\mho, \ldots)$ and $p_\Lambda(\mho)$ respectively.

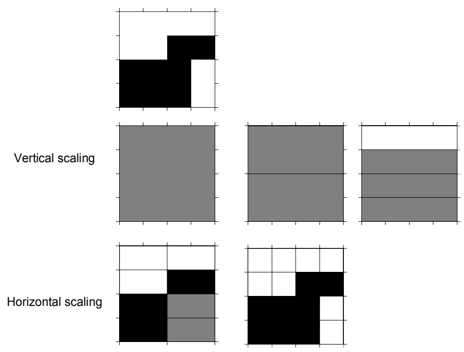


Figure 6. Scale decomposition of an image by the dyadic tree with 8 by 8 pixels (top left is the image, second row is the vertical scaling M_y=0, 1, 2 at M_x=0 and bottom row is the horizontal scaling M_x=1, 2.

6.1 Models on the Dyadic Tree

The first model is called 1-2V 1-2H model. It is similar to what is considered earlier in the quad tree (1-4 model). The 1-2V 1-2H model means: division of the image into two horizontal strips until the pixel line is reached in the vertical direction. Then each strip is divided into two cells in the horizontal direction until the pixel line is reached. In Figure 7 (left image), at the vertical scale M_y=0 and the horizontal M_x=0 the color is *G*. This node in the tree goes to two horizontal strips at the next level (M_y=1 and M_x=0) with colors *GB* from top to bottom and so on. Then the procedure stops in that direction because the pixel line is reached. The scaling procedure starts in the horizontal direction. The top white *W* strip goes to two white children *WW*, while the next gray *G* strip goes to *WB*.

6.2 Test Cases of the Dyadic Tree

The first test is simulation of a layered system. This is similar to what has been performed earlier in the quad tree. Figure 8 (left part) first and second rows show the data the results respectively. The simulation results are quite different from the quad tree case (see Figure 4). In this situation the tree will show *G* at all scales up to the pixel level in the vertical direction where the strips will be either *B* or *W*. From the two image data set *G* goes either to *WB* or *BW* with probability 0.5. The combination *WB* or *BW* will be repeated in the simulations and sometimes *WB* will come to be in contact with *BW*. The horizontal scaling does not play a role in this particular test because all the colors *B* or *W* will already have appeared in the vertical scaling. Another test is also made to simulate a checkerboard system. Figure 8 (right part) shows data and results of this test. The simulation results produce some correlation in the system that is not present in the original data. The correlation shows different patterns in comparison with the case showed by the quad tree. This is because of the different tree representation.

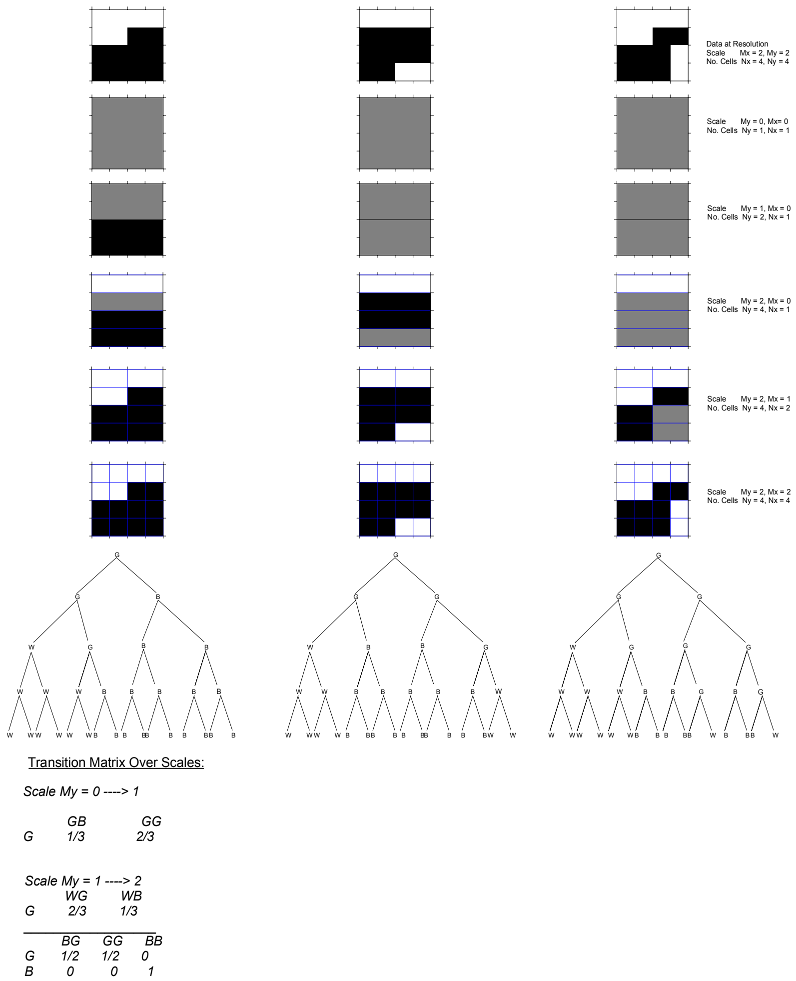


Figure 7. Sample images and calculation of their transition probabilities over the various scales in case of dyadic tree representation (1-2V 1-2H model), calculations are shown in this example up to scale M_y=2, M_x=0, similar calculations can be done for further scales.

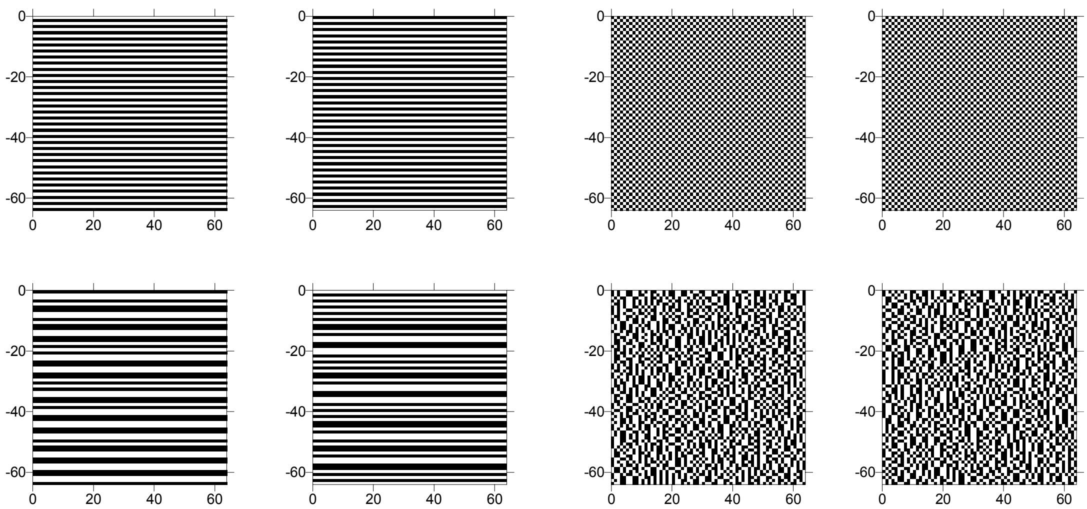


Figure 8. Test cases of the dyadic tree: the top row consists of the data and bottom row consists of the simulations. Left part: layered system, right part: a checkerboard system, (image resolution K_x=6, N_x=64, K_y=6 and N_y=64).

7. SIMULATIONS WITH DYADIC TREES

Some simulations with the dyadic tree are performed in this section. The data that has been used in the quad tree is used once again here for comparison reasons.

7.1 Merging Large-scale Structures with Fine-scale Structures

In this example, simulations with original and smoothed dyadic trees are performed to merge large-scale structures with fine-scale structures. This simulation is similar to the one performed using quad tree (see Figure 5). Figure 9 shows the data (1st column from the left) and the simulations (2nd, 3rd and 4th columns) for this example. The results in this example are rather good from geological point of view. The interpretation of the results is as follows. The technique tries to embed the fine-scale structure into the large-scale structure to produce two-scale heterogeneous structure. Therefore, what is observed in the simulations is intrusion of the fine-scale structure that has specific characteristics (i.e. long correlation in the horizontal direction and short correlation in the vertical direction) in the large-scale structure. Simulations with a smoothed dyadic tree have also been performed (see Figure 9, 3rd and 4th columns for degree of smoothing no.1 and degree of smoothing no.2 respectively). One would recognize in case of degree of smoothing no. 1 (figure 9, 3rd column), that there is no significant difference between this simulation and the simulation by the original dyadic tree (no-smoothing effects, figure 9 in the second row). However, the simulations for the degree of smoothing no.2 show more black and white spots (figure 9 bottom row) than in the simulations with the original dyadic tree. The degree of smoothing no.1 has relatively no influence in comparison with smoothing degree no. 2.

7.2 Merging Heterogeneous Structures Using the Multi-Gray Concept Versus the Single Gray Concept

In all the previous simulations, a single-gray concept is used to describe any degree of mixture between black and white. It is thought that a distinction between different degrees of mixture of black and white would produce different simulation results. In this example three different degree of gray are distinguished:

1. Single gray: means a rectangle that contains any degree of mixture of black and white between more than 0%B up to less than 100%B is called gray.
2. Two grays: mean a rectangle that contains more than 0%B and less than or equal 50%B is called gray1 while a rectangle that contains more than 50%B and less than 100%B is called gray2.
3. Four grays: mean a rectangle that contains more than 0%B and less than or equal 25%B is called gray1. A rectangle that contains more than 25%B and less than or equal 50%B is called gray2. A rectangle that contains more than 50%B and less than or equal 75%B is called gray3. A rectangle that contains more than 75%B and less than 100%B is called gray4.

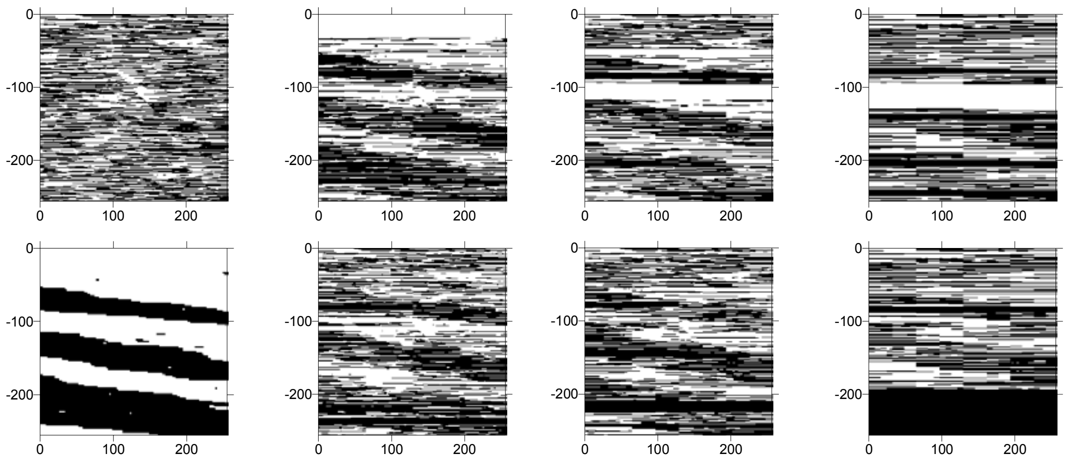


Figure 9. Comparison of merging two different heterogeneous structures using smoothing effects. Data (left column), no smoothing simulation (2nd column from the left), degree of smoothing no.1 more than 87.5%W is considered white (3rd column from the left) and degree of smoothing no.2 more than 75%W is considered white (right column). Image resolution K_x=8, N_x=256, K_y=8 and N_y=256.

Figure 10 shows results of a numerical experiment that implement the concept of multi-grays. The left column of Figure 10 is the data set used. The next three columns of images show the performed simulations using single gray, two grays and four grays respectively. The simulations show no difference between using single-gray and multi-gray concept. The reason is due to that fact that the information that is presented in the tree either stored by single-gray or multi-gray up to the pixel level is location dependent. Then when realizations are drawn from the tree at certain level is strongly dependent on that location in the tree whatever is coded by single or multi-gray concept.

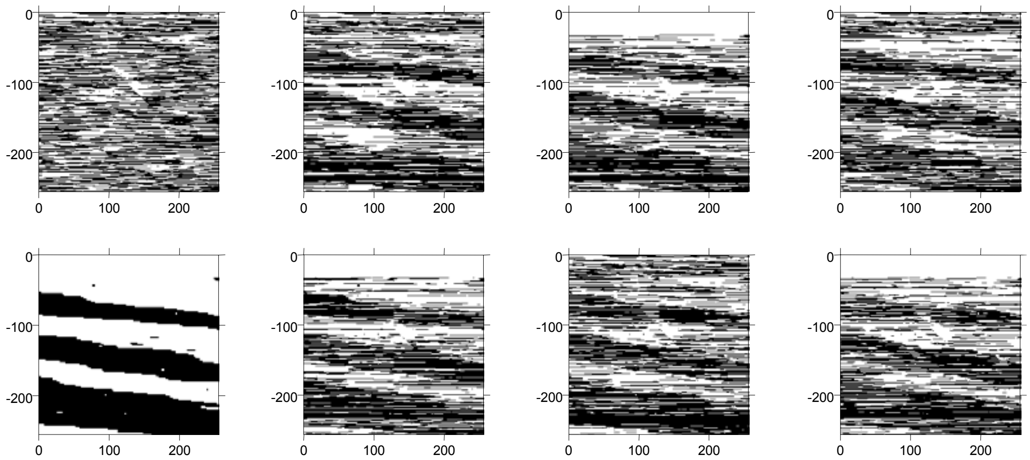


Figure 10. Comparision of merging two different structure using single-gray and multi-gray concepts. Data (1st column from the left), single-gray simulation (2nd column), two-grays simulation (3rd column) and four-grays simulation (right column), image resolution K_x=8, N_x=256, K_y=8 and N_y=256.

8. COMPARISONS BETWEEN QUAD TREES AND DYADIC TREES

Another set of numerical experiments has been performed using different tree representation. Figure 11 shows some examples of thin horizontal layers and thin vertical layers. Another example with inclined thin layers at 45 degrees and at 135 degrees. Figure 12 shows another examples with isotropic versus an-isotopic structures.

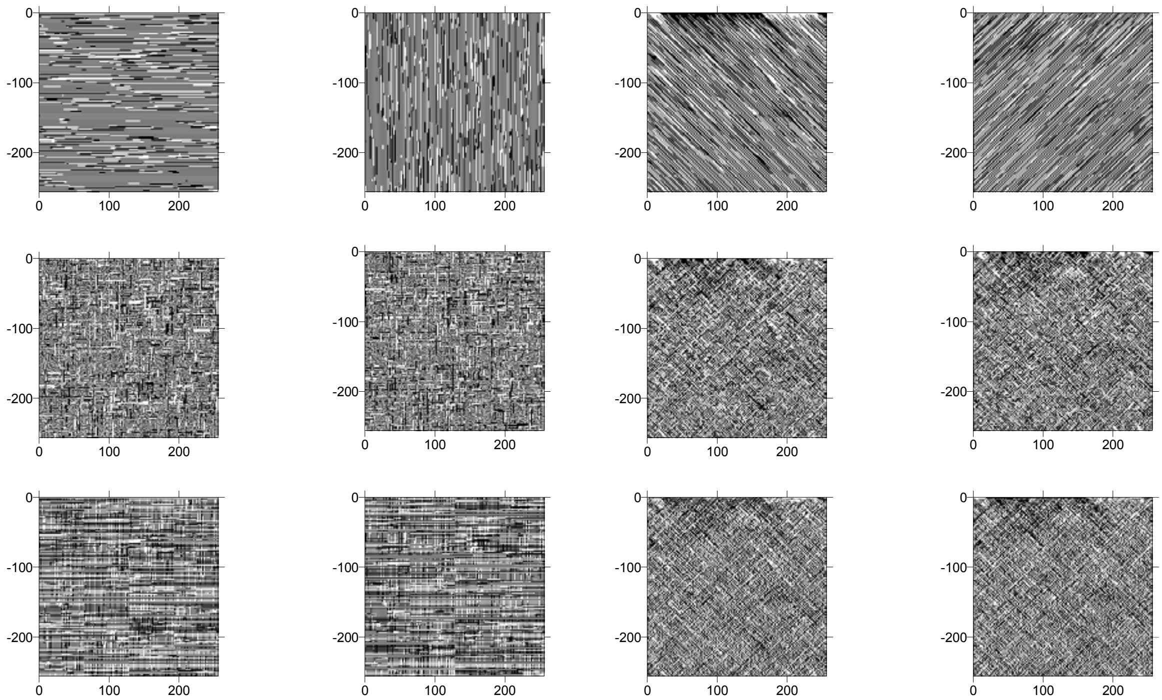


Figure 11. Comparision of quad tree and dyadic tree simulations (top row is the data, middle row is the simulations with the quad tree and bottom row is simulation with the dyadic tree (image resolution K_x=8, N_x=256, K_y=8 and N_y=256).

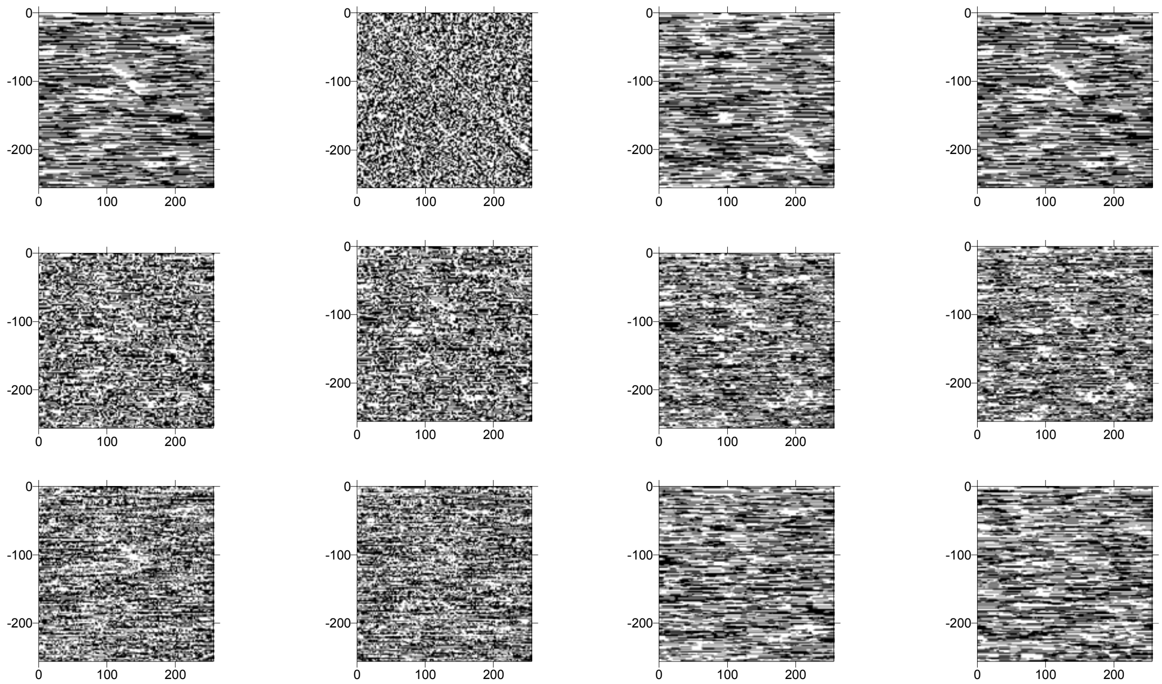


Figure 12. Comparision of quad tree and dyadic tree simulations (top row is the data, middle row is the simulations with the quad tree and bottom row is simulation with the dyadic tree (image resolution K_x=8, N_x=256, K_y=8 and N_y=256).

7.3 Weighed Input Data

In this example, an experiment is performed to study the influence of weighting the input data. The reason to use this option is to show the possibility to give more weight to the structure that it is considered to be more dominant. Figure 13 shows results of this experiment. The two structures in the most left column are the data. Inclined fine-scale structure (bottom image) and large-scale structure (top image). The second column from the left is the simulation results with only equi-probable input data i.e. probability 0.5 either for the top image or for the bottom image. While in the third column we have simulation results from data that contains two input images of the fine-scale structure and one image of the large-scale structure. The weights in this case are 2/3 for the fine-scale structure and 1/3 for the large-scale structure. In the right column an experiment is performed with three data images of the fine-scale structure and one data image of the large-scale structure. The weights in this case are 0.75 for the fine structure and 0.25 for the large-scale structure. The simulation results show that increasing the weight of the fine structure leads to smearing of the large-scale structure.

CONCLUSIONS

A framework for the manipulation and simulation of binary images has been developed. The framework is based on hierarchical representation of binary images that can be straightforwardly extended to gray value images. This technique is attractive for many applications. A description of the technique with some simulations on synthetic data is presented. Computer codes written in FORTRAN language have been developed to implement this methodology. The programs are highly flexible, and permit the user to insert his ideas. A series of numerical experiments has been carried out to investigate the applicability of this methodology. The following conclusions can be drawn from the experiments:

1. The proposed methodology is capable of merging different heterogeneity patterns at various scales.
2. Stationary fields, nested and compound structures can be addressed by this methodology.

3. A comparison between different tree descriptions (quad tree based representation versus dyadic tree based representation) has been addressed.
4. In binary images that contains black and white colors, a comparison between the single-gray and the multi-gray concept has lead to the conclusion that there is no significant difference in the simulation results when using either of the two concepts.
5. Both quad tree and dyadic tree show different simulation patterns in some cases and no differences in other cases. This behavior is dependent on the type of data.
6. Smoothed trees have been introduced as a deviation from the original trees to show a way out for smoothing noisy data.
7. Weighted input data provide an option to make some of the structures to be more pronounced than the other. This option could be useful for some applications.

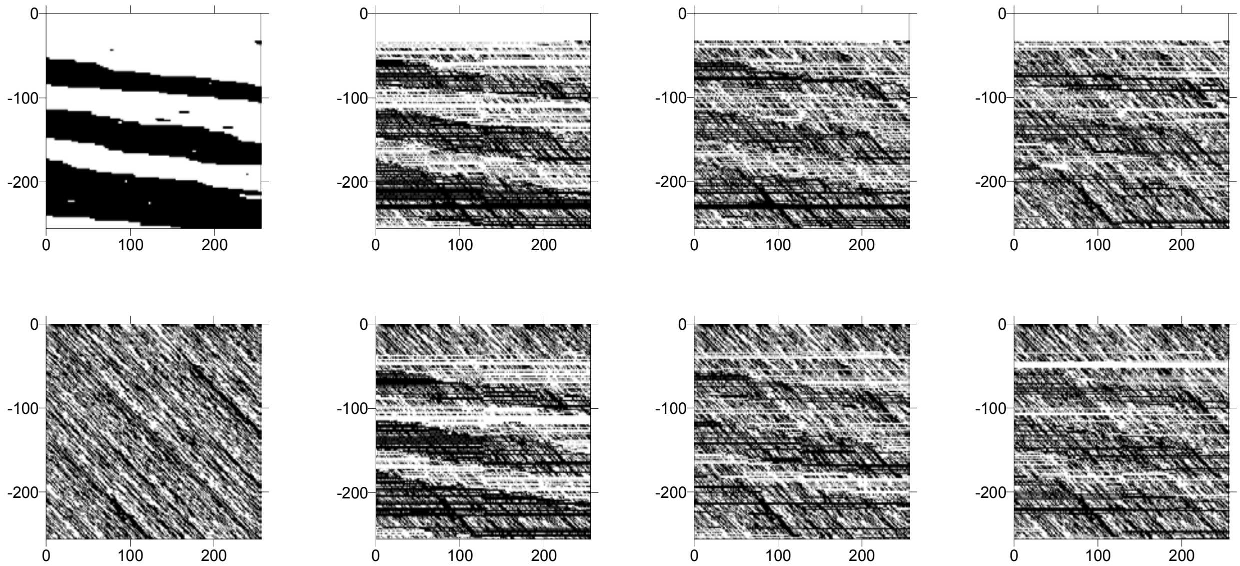


Figure 13. The influence of weighting the input data. Left column is the data, second, third and fourth columns are simulations with 1/2-1/2 probability, 2/3-1/3 probability and 3/4-1/4 probability respectivley. (Image resolution: K_x=8, N_x=256, K_y=8 and N_y=256).

ACKNOWLEDGMENTS

This study is financially supported by the DIOC project of Delft University of Technology.

REFERENCES

1. F. M. Dekking, C. Kraaikamp and J.G. Schouten, “Binary images and inhomogeneous tree-indexed Markov chains,” to appear in *Rev. Roumaine Math. Pures Appl.*, 1999.

2. A.M.M. Elfeki, “*Stochastic characterization of geological heterogeneity and its impact on groundwater contaminant transport,*” Ph. D. Thesis. Delft University of Technology, Delft, The Netherlands, 1996.

3. A.M.M. Elfeki, G.J.M. Uffink, and F.B.J. Barends, “Stochastic simulation of heterogeneous geological formations using soft information, with an application to groundwater. In *Groundwater Quality: Remediation and Protection, QG’95*., ed. by Kovar, K. and Krasny, IAHS Publication No. 225, 1995.

4. H. Samet, *Applications of Spatial Data Structures*, Addison-Wesely Publishing Company. Inc, Reading, 1990.