Herriot-Watt University

Institute of Petroleum Engineering

Supervisor(s)-Name(s)
-from HW:
-from HW ASC: Margarita Eliseykina

MSc Reservoir Evaluation and Management
Project Report 2005/2006

Author: Anatoly Chernoivanov

Title: Implementation of SIMPAT M algorithm
(SIMulation with PATterns Modified)

Table of contents

Introduction

The aim of this work is to develop a geostatistical sequential simulation algorithm SIMPAT M based on Patterns drawn from a Training image. The algorithm is derived from works of scientists from Stanford University [1, 2] (A good review of the sequential simulation methods avialible is also given by them). The algorithm takes a facies distribution in 2D or 3D space as its input Training image and yields an alternative facies distribution as its output realization, which differs from the input distribution, but has characteristic details of the latest (Fig. 1). For example, if the input of the algorithm SIMPAT M is a distribution of channel sandstones in shaly floodplain, the output should be also the sandstone channels in shaly floodplain with the same facies (sand to shale) ratio.

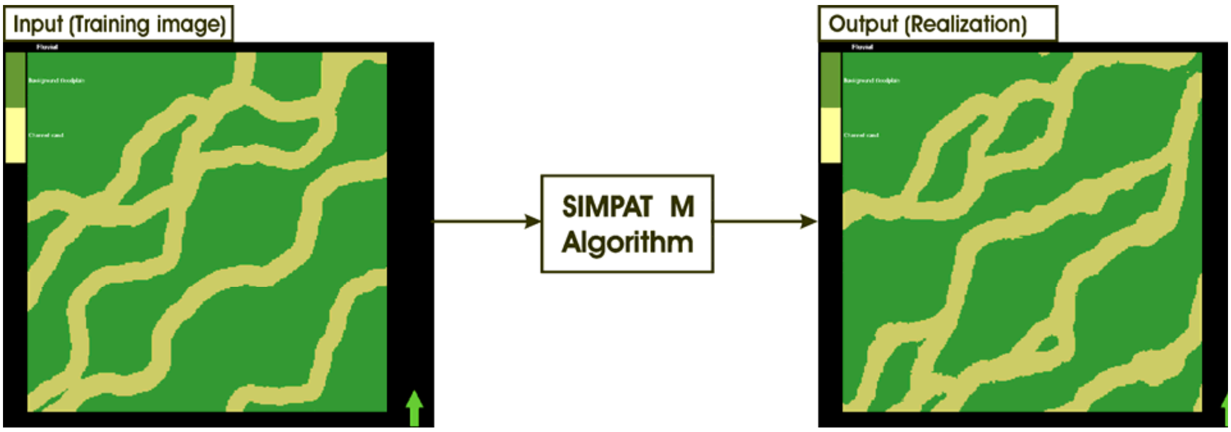

Fig. 1, A functional scheme of the SIMPAT M algorithm, from left to right: input data (image 250x250x1 cells of fluvial channels) – the algorithm – output data (image 250x250x1 cells)

A novelty of this work is concluded in a new approach of the data organization and assiciated methods of search for the appropriate Pattern to be inserted into the Realization during the sequential simulation. The issue is the following.

It is well known, that the fastest search is related to the tree-like organization of elements, to be searched for. It is pretty easy to implement the tree for numbers, since they have a sequence, but images have no any sequence. Even if we represent an image as a multidimensional number, the sequense of such numbers will not have useful meaning, as the closeness between these numbers is not the same as the similarity of images itself. For example, in an image we can change just one high bit, what replaces the image far from the place it is in the sequence of images. It is not good.

So that, it have been desided to use the visual nature of images and develop methods, building tree-like Data Base of Patterns with the aid of the coarsened versions of Pattern images drawn directly from the initial Training image. The future development of the methods considered will allow to improve the algorithm preformance, especially if they will be used in conjunction with the methods implemented with the basic SIMPAT algorithm. In addition, an effective communication with the authors of the the SIMPAT algorithm, Guven Burc Arpat in particular, approved the usefulness of this work as a research one.

To simplify the process of debugging and testing base functions of the algorithm, 2D training images are used in this work. These images represent 2- facies distribution, where the first facies is floodplain shale and the second one is channel sandstone. The architecture of the program allows to develop an extension version of the current SIMPAT M algorithm implementation, dealing with 3D multiple facies distribution.

The implementation of the algorithm is written in C# programming language, since it is the most comfortble for author and potential Customer. The program can read input file of Training image and write output file of Realization using the data format *.VIP avialible in the Schlumberger Petrel Package.

Common scheme of the SIMPAT M algorithm iterations

The common scheme of the SIMPAT M algorithm iterations looks like the following. At the every step of a basic cycle of the algorithm, a place for the data event is choosen at random at space of the initially empty Realization to fill the place with the values comprising a Pattern from the Training image (Fig. 2). Data event is defined as the set of values of Realization, which are within the boundaries of a Template, placed within the Realization. Template defines the shape and size of Patterns used as well as that of Data event (which are equal for both).

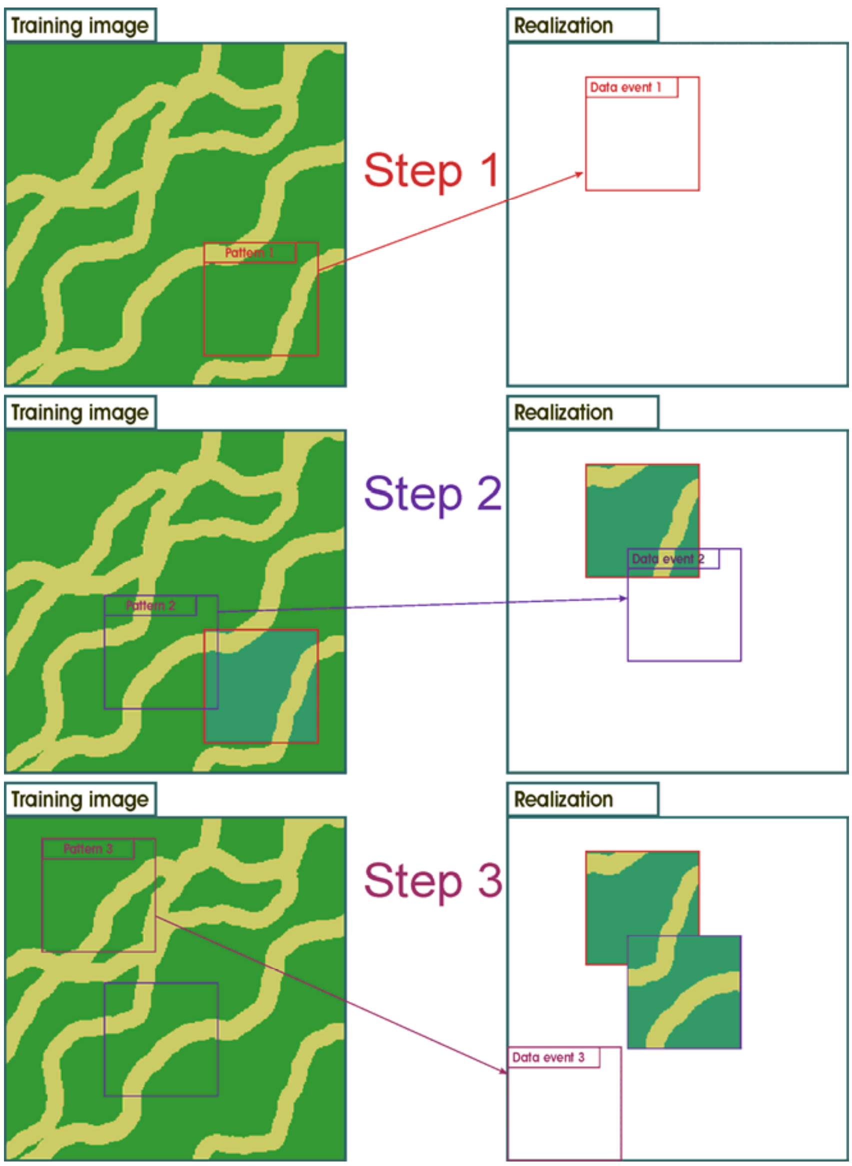

Fig. 2, The basic cycle of the SIMPAT M algorithm

If the Data event contains only empty values (Fig. 2, Step 1 and Step 3), a Pattern of the same shape and size is selected almost at random from the Training image. “Almost at random” means, that the facies ratio is controlled during the selection, and only those Patterns are selected, which restore the ratio if it have been violated during the previous steps of the simulation. When the Data event contains non-empty values (Fig. 2, Step 2), the search for a Pattern is conducted. The whole Training image is scanned until the Pattern selected repeats the values of the Data event with a minimum of error.

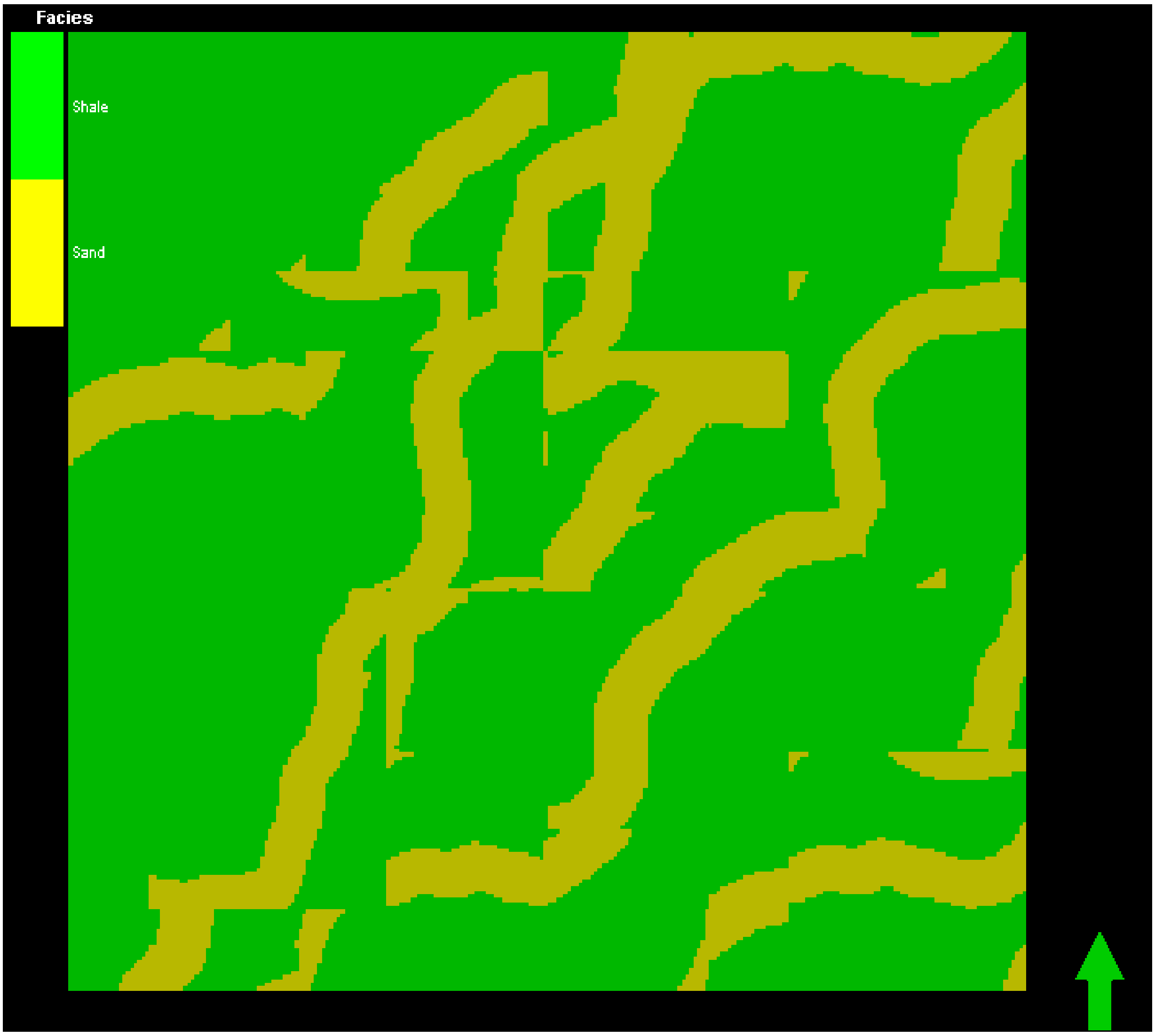

Fig. 3, The result of a SIMPAT M algorithm iteration using the largest Template. The small-scale detales are failed, the shape of the channels are highly distorted in an unnatural manner, what is absent in the initial Training image.

At the first iteration of the SIMPAT M algorithm, the Template is choosen large enough to account for the large-scale detales of the training image. These detales are, for example, the maximal distance between any two neighbour fluvial channels. In the SIMPAT M algorithm, the procedure of calculating such a Template is implemented. Nevertheless, because of the random character of the Data event selection and limited number of Patterns, it is ofthen happen, that even the best Pattern close to Data event (similar to it) fails small-scale detales evident in the Training image at the finish of the iteration (Fig. 3).

To escape artefact presented in the Realization after the first iteration of the SIMPAT M algorithm (Fig. 3), the next iterations are conducted, using smaller Templates. The basic cycle of the algorithm is repeated but a smaller template is used and the Realization at the start of each consequent iteration has already filled with non-empty values, calculated at the previous iteration (Fig. 3). Thus, the basic cycle of the SIMPAT M algorithm (Fig. 2) is repeated several times for different Templates to account for large-scale detales as well as small-scale ones.

Pattern search and comparison with a Data event

The most time-consuming operation of the SIMPAT M algorithm is the search for the Pattern, which has the least distance to the appropriate Data Event (Fig. 2, Step 2). The measure of the distance is defined as the square root of the sum of the square differences between the non-empty cell values of the Data event and that of the Pattern (1).

$$Dis\tan ce = \sqrt{\sum_{i=1}^{N}(P_i - D_i)^2} \qquad (1),$$

where N is the number of the non-empty cells in the Data event,

P_i is the value of the Pattern cell i

D_i is the value of the Data event cell i

The great search time is related to the necessity to scan the whole Training image, and paste each of N values into the formula (1) to calculate the distance between the Data event and Pattern. To improve the quality of the distance calculation and make the calculation result be close to what is expected by a human [1], the Chamfer distance transformation is applied to the Training image before the SIMPAT M algorithm begins its iterations (Fig. 4). All operations within the SIMPAT M algorithm are conducted on the Chamfer transformed images.

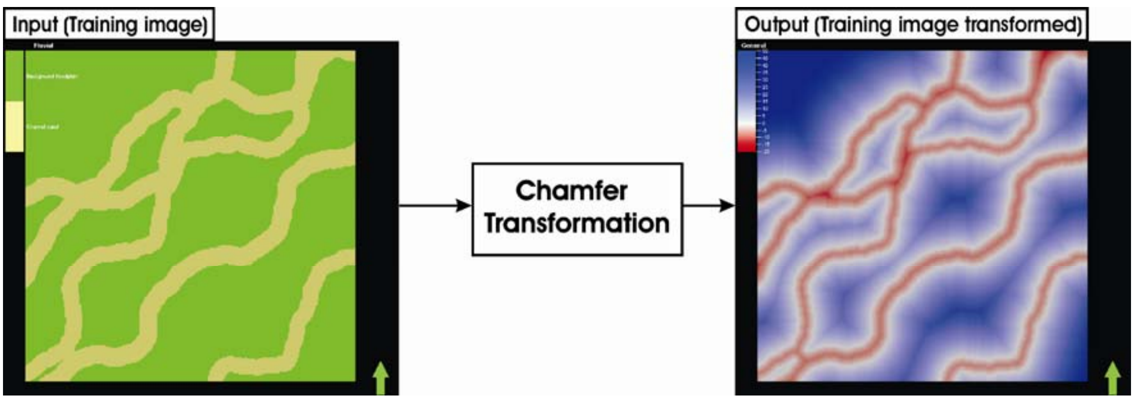

Fig. 4, A functional scheme of the Chamfer Transformation procedure. Blue colour (on the right), positive values, mark a minimal distance to the nearest channel. Red colour, negative values, mark minimal distance to the nearest channel boundary.

One of the methods to decrease the amount of elements in the sum (1) is to skip a sertain number of elements during the calculation and to fill the Realization, using cascading grids and dual sparse templates [1] (Fig. 5, 6). The step of the cascading grid is equal to the number of skipped cells plus 1 during the Pattern to Data event comparison. So that, the large-scale details are taken into account during the iterations.

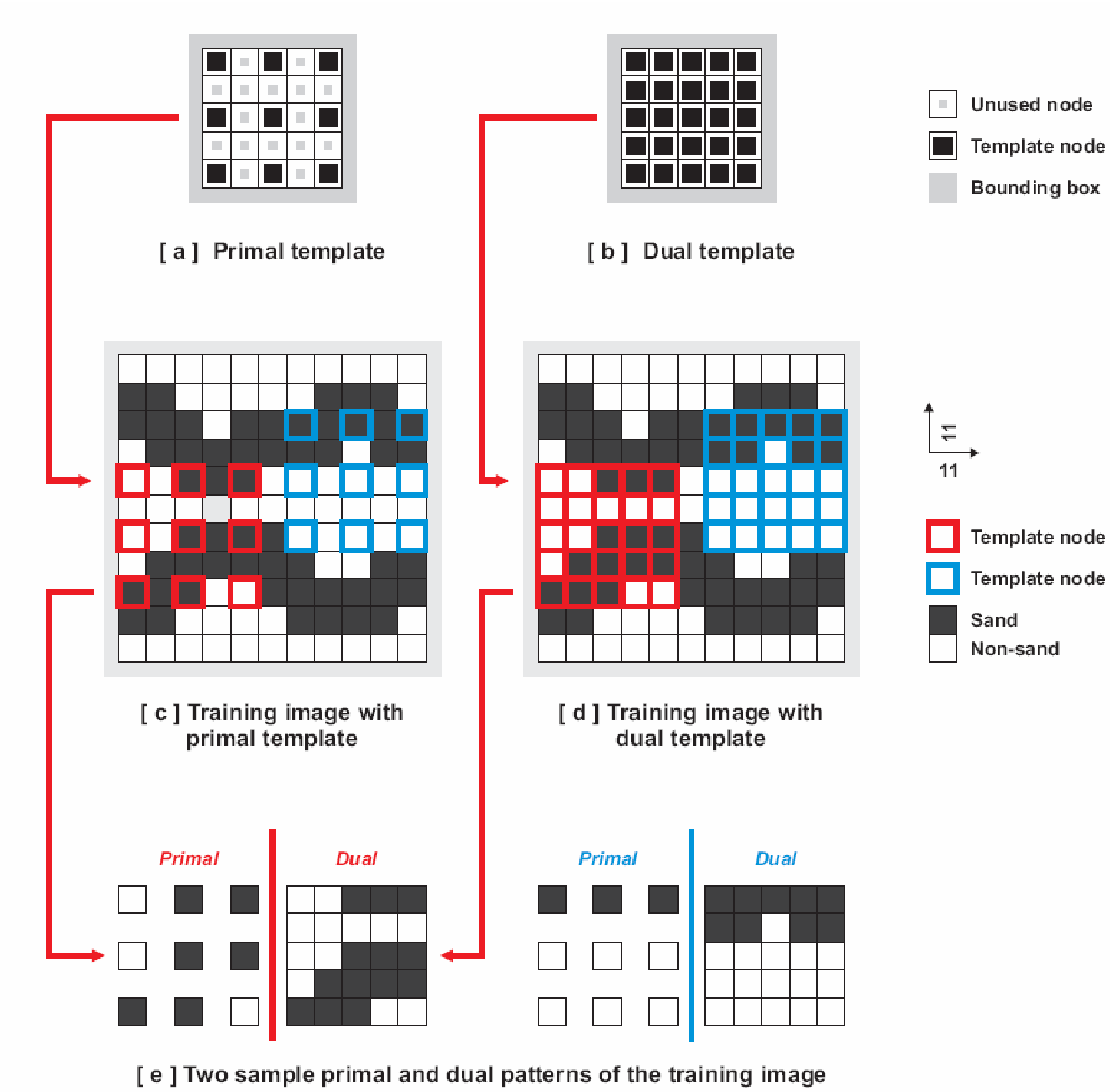

Fig. 5, Guven Burc Arpat, 2005, A scheme of the extraction of dula templates from a Training image.

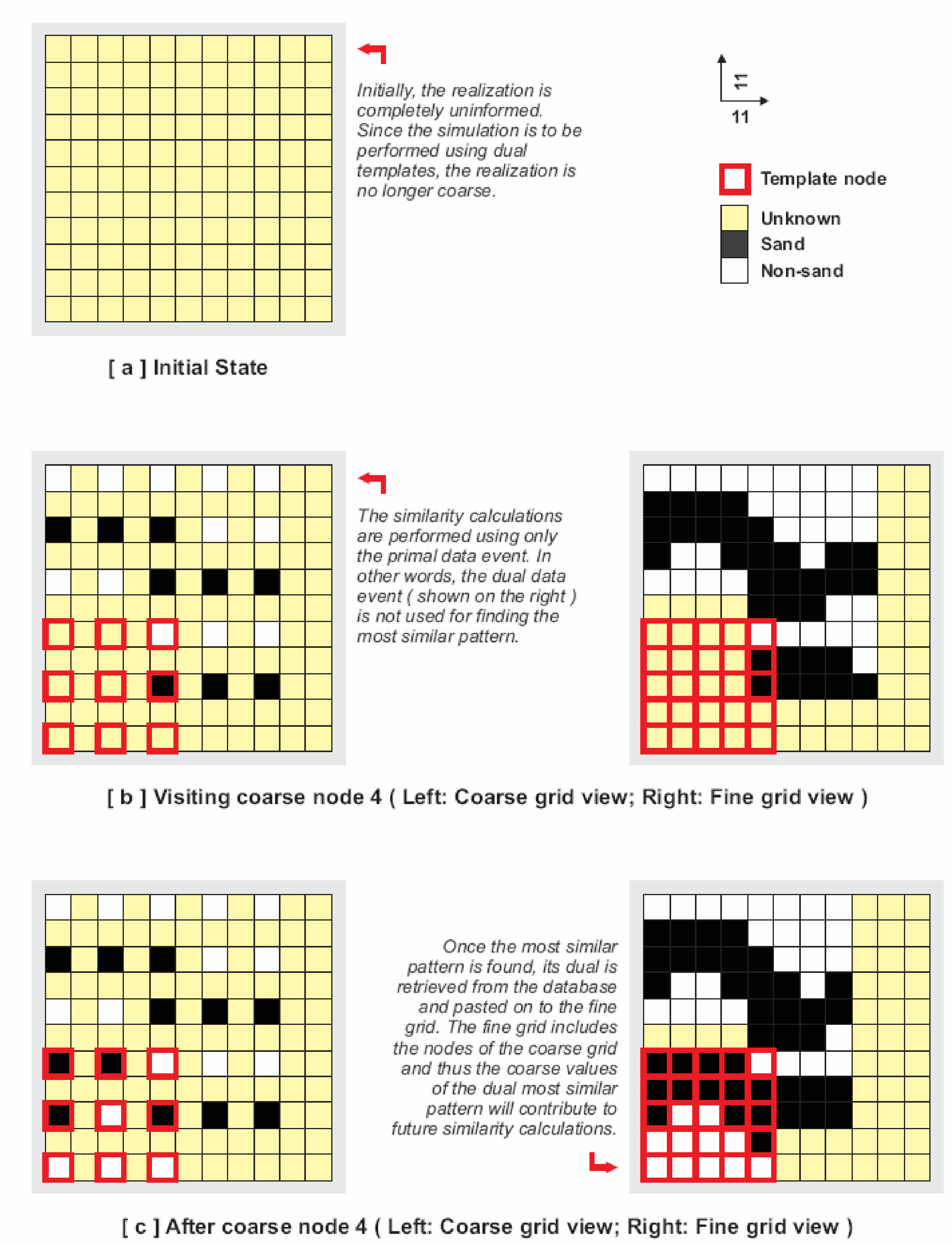

Fig. 6, Guven Burc Arpat, 2005, A scheme of the basic cycle of the SIMPAT algorithm, using the method of dula templates.

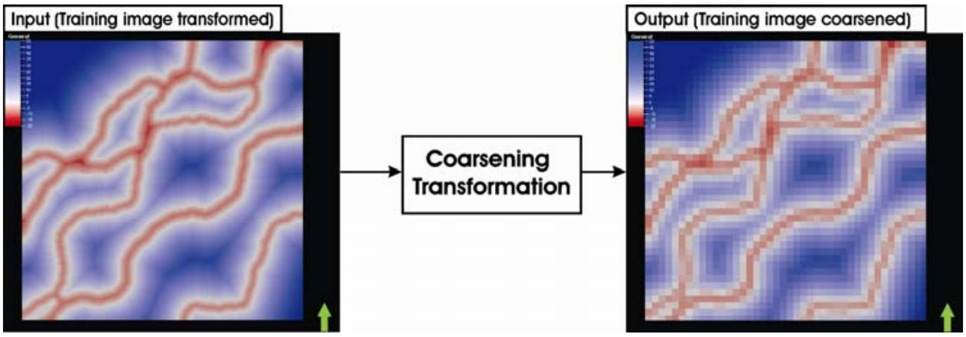

Fig. 7, A functional scheme of the Coarsening transformation procedure.

In this work, it is suggested an alternative method reducing the amount of the calculations. The method is based on the Coarsenong transformation procedure. The value of each coarse cell (Fig. 7, on the right) is a weighed average of fine cell values (Fig. 7, on the left), intersected with the coarse cell. The weight of the averaging is defined as a volume or area of the Cartesian product of a fine and coarse cells. This transformation allows to reduce the number of cells, although some information from the non-coarsened image is lost, fine detales disappear.

Before the execution of the basic cycle of the SIMPAT M algorithm, the copy of the initial Training image is created. This copy is coarsened with the aid of the Coarsening transformation procedure (Fig. 7), and then the Pattern Data Base constructor takes the coarsened copy together with the initial Training image as its input (Fig. 8). It should be noted, that the coarse grid step is selected in a way to make the sum of a number of steps equal to the length or width of the Template for the Patterns and Data events, to simplify the logic of the program.

The following happens during the construction of the Pattern Data Base. Each pattern from the initial Training image (Fig. 8, on the top left) is linked to the nearest Pattern from the coarsened Training image (Fig. 8, on the bottom left). As a result, The Pattern Data Base contains an array of references to Coarse Patterns (Fig. 8, on the bottom right), each of which includes a number of references to fine Patterns (Fig. 8, on the top right) extracted directly from the initial Training image. This is the simplies tree-like organization of the Pattern Data base.

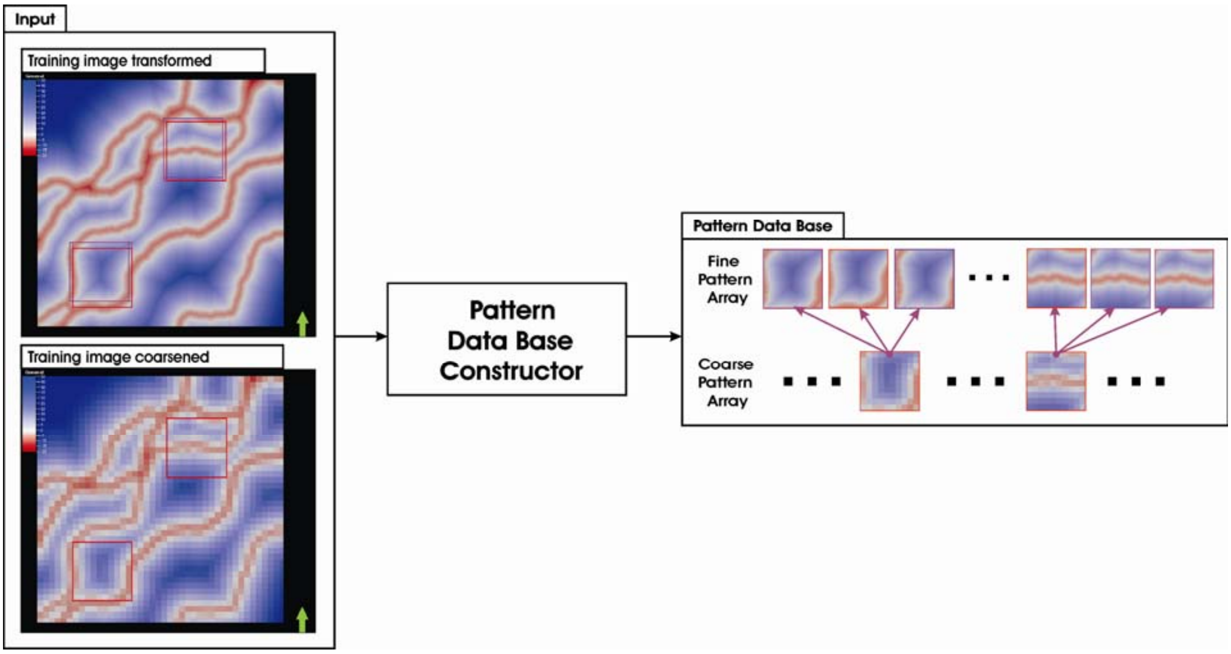

Fig. 8, A functional scheme of the Pattern Data Base Constructor.

After the construction of the Pattern Data Base, the iterations of the algorithm SIMPAT M are sequentially started. During each iteration (Fig. 9), a place for the Data event is chosen at random in the Coarse Realization, to insert into it the most appropriate Pattern. As it have been considered above, if the Data event contains only empty values, the appropriate Pattern is selected at random from a set of Patterns, which support the facies ratio. In case of the non-empty values in the Data event (Fig. 9, action number 3), the Pattern, which is the closest to the Data event, is selected from the array of coarse Patterns (Fig. 8, on the bottom right). This selection is several times quicker than that applied directly to Fine Data event and Fine Pattern array. The high speed of the selection (or search) is related to:

a) a lower number of coarse Patterns relative to that of fine ones;

b) a several times lower number of cells in each of coarse Patterns.

As the appropriate Pattern is found, it is inserted into the coarse Realization and the search for the fine Pattern is started to find the Pattern which is the closest to the fine Data event. This search can be even more quickly than that considered just above, because the Pattern is selected not from the whole array of the fine Patters, but from that small subset of the array, in which each Pattern is linked to the coarse Pattern found previously (Fig. 9, the links or references are marked with violet pointers). This scheme is analogue to that of dual templates, mentioned above (Fig. 6), and coarse patterns play the role of dual template Patterns (Fig. 5).

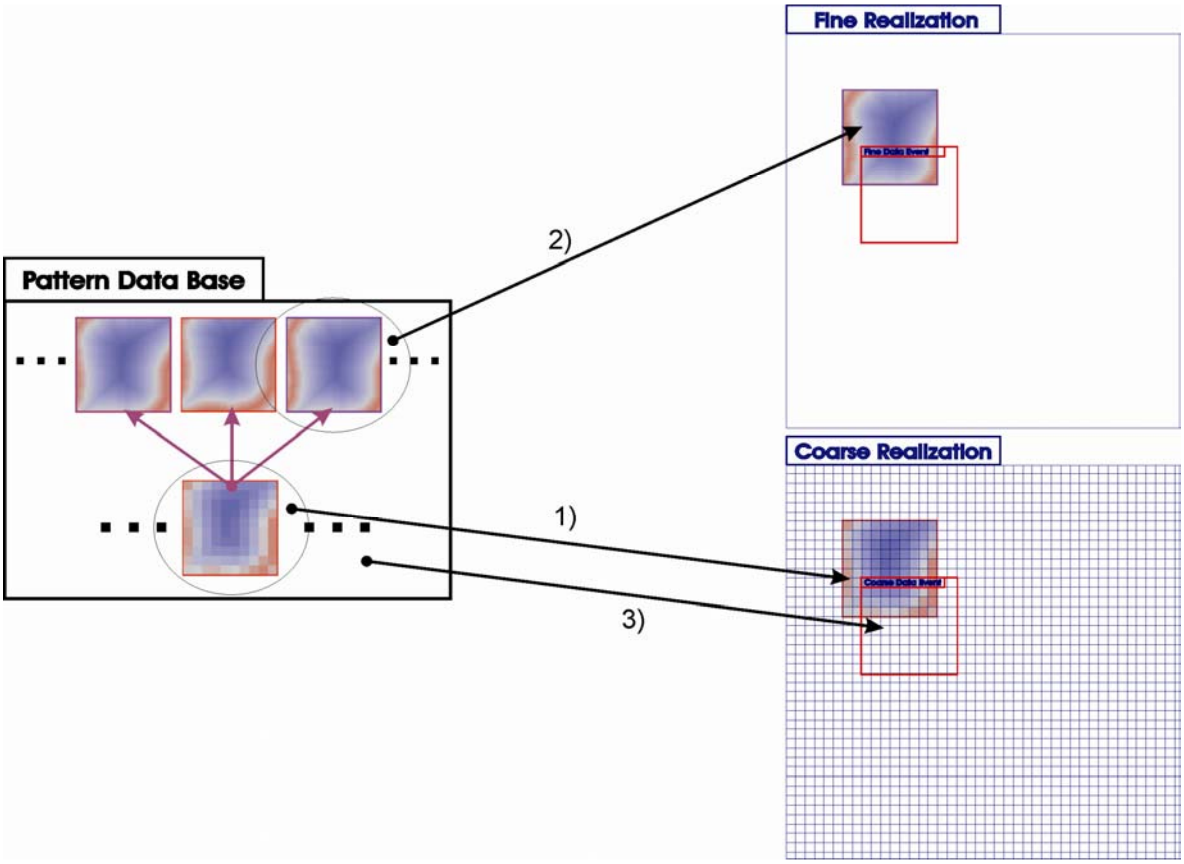

Fig. 9, A scheme of the basic cycle of the SIMPAT M algorithm, using the method based on the Coarsening transformation preocedure. 1), 2), 3) are the numbers of the sequential cycle actions.

During the first iteration, the coarse and fine Realizations, which contain initially empty values, are filled with the values from appropriate Patterns of the largest Template size. Nevertheless, an effective search for appropriate Patterns does not resolve the problem of the small—scale distortions, mentioned above (Fig. 3). To solve the problem, we should use smaller and smaller Templates. So that, all consequent iterations are utilized to solve the problem.

The fine Realization from the previous iteration is directly taken by the next iteration as the input, whereas the coarse realization for the next iteration is obtained from the fine

Realization from the previous iteration with the aid of the Coarsening transformation procedure. The coarsening is conducted, using the same grid step as that applied for the Pattern Data Base construction used in the next iteration (Fig. 10, 11, 12). The Pattern Data Base is constructed at the start of each iteration, what is related to a smaller Template used. Thus, the size of the Patterns used and grid step of the coarse Realization and coarse Training image, from which Patterns are extracted, should be in agreement with the Template size used.

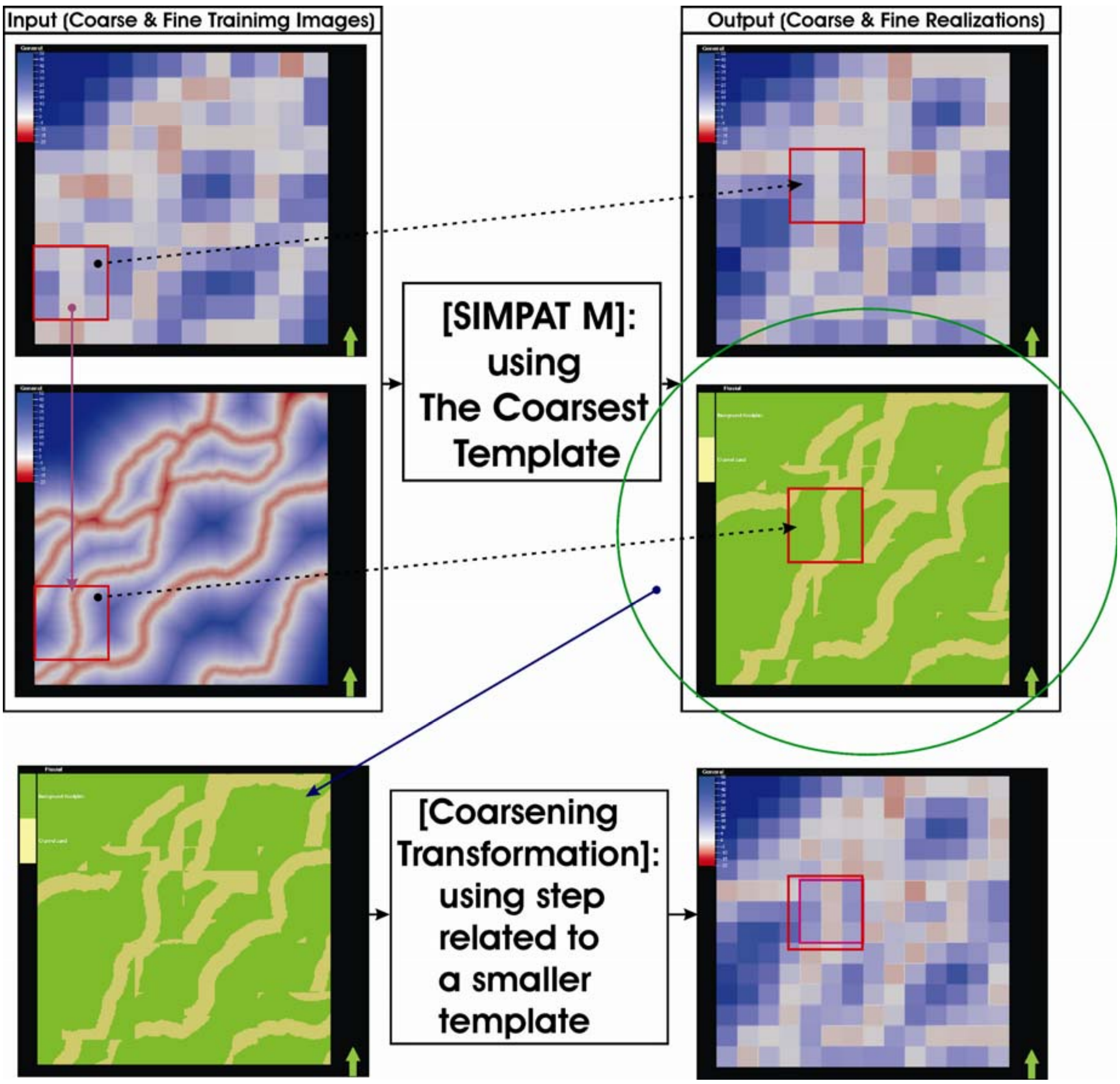

Fig. 10, An extended scheme, representing the functional scheme of the basic cycle of the SIMPAT M algorithm and the functional scheme of the Coarsening transformation, performed on the fine Realization after the previous iteration and before consequent one.

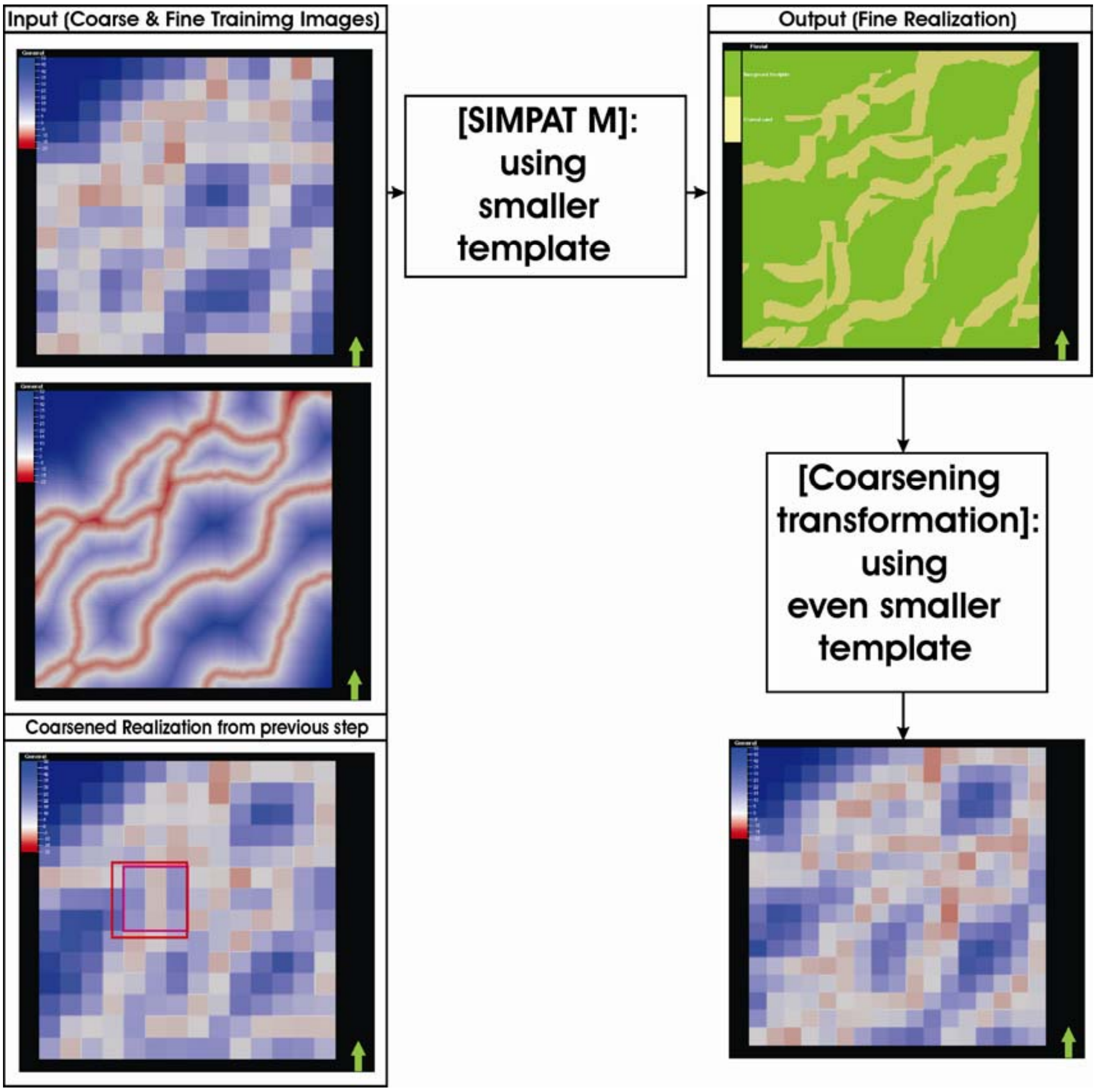

Fig. 11, An extended scheme of an iteration of the algorithm SIMPAT M, using the smaller template relative to that in the previous iteration (Fig. 10).

Compare the coarse Realizations (on the bottom left and right), the Realization on the right have the better quality, wherease channels have less distortion relative to the result obtained on the previous step.

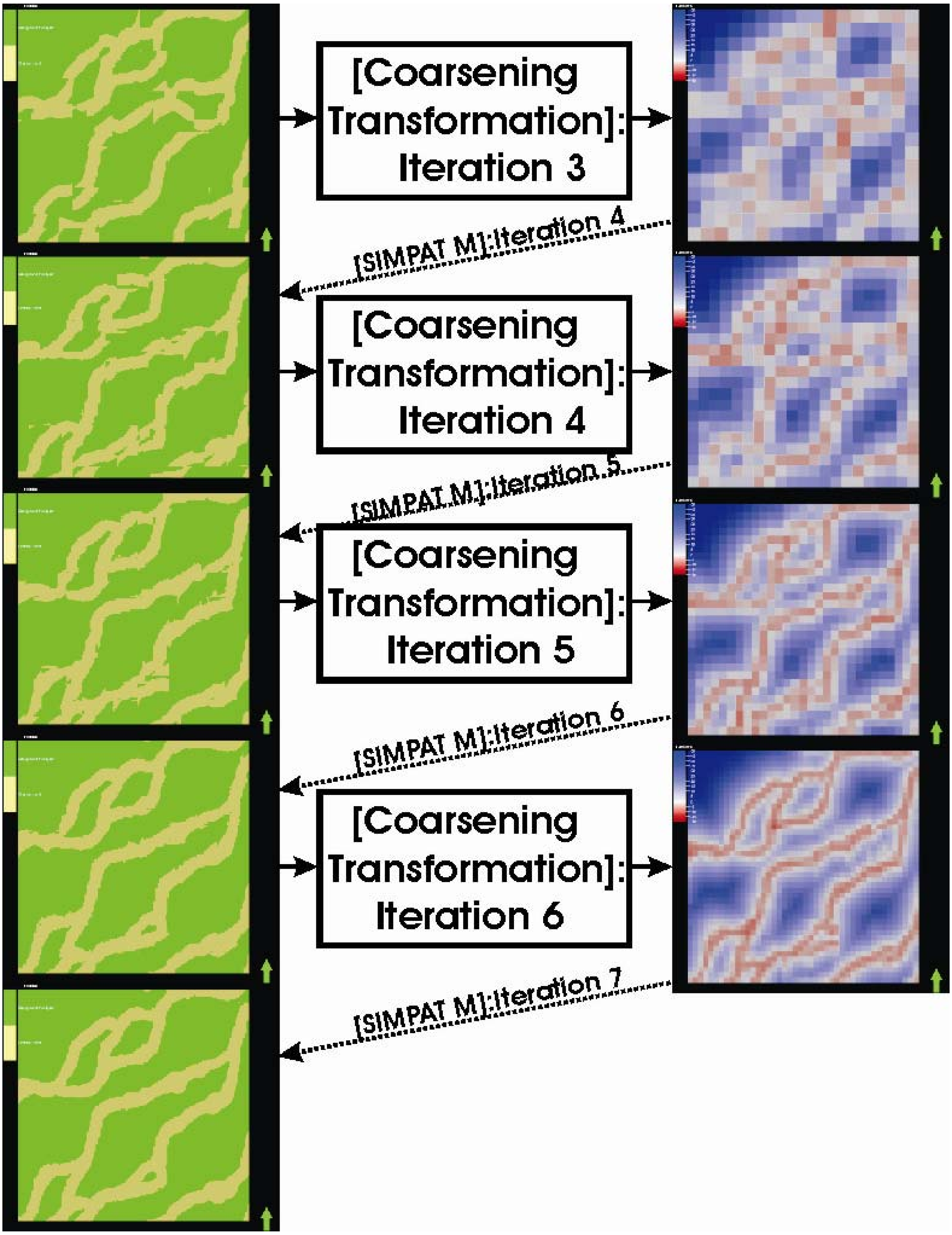

Fig. 12, Sequential intermediate steps between the iterations of the basic cycle of the algorithm SIMPAT M (Fig. 9).

SIMPAT M algorithm performance

The calculation time of the SIMPAT M algorithm is several percent greater than that of the basic SIMPAT algorithm, e.g. SIMPAT M constructs the final Realization of 250x250x1 cells from the Training image of the same size spending about 8 minutes on the calculations, whereas the basic SIMPAT algorithm produces the same task in 7 minutes. Nevertheless, the time for the SIMPAT M can be reduced with the aid of using the Pattern Data Base, having the tree depth greater than 2. Greater tree depth allows to decrease the number of comparisons between a Data event and a Pattern searched, which is espetially actual for big Templates, which contain a lot of cells, and have many fine Patterns linked to them.

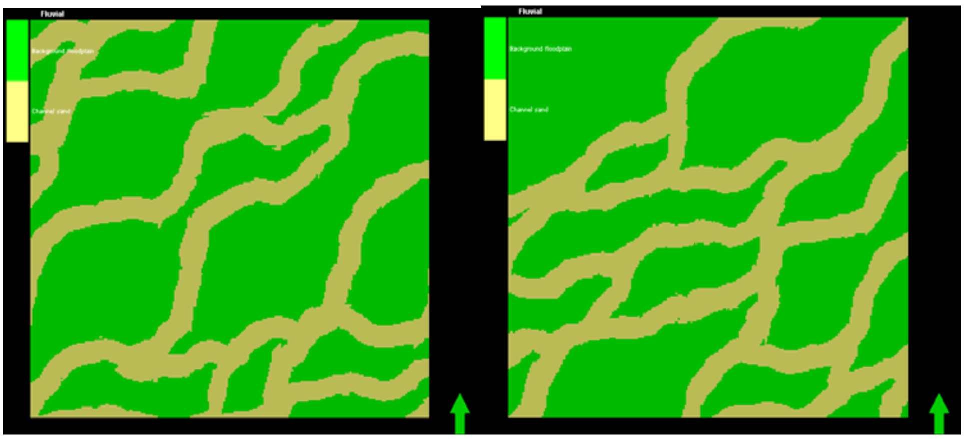

Fig. 13, SIMPAT Realization is on the left, facies ratio is 0.35. SIMPAT M Realization is on the right (another realization), and facies ratio is 0.34. Initial training image has the facies ratio equal to 0.29

The Realizations for both algorithms are rarely “different”, since they contain the same characteristic channels, the same facies ratio (Fig. 13). The facies ratio of the Training image is about 0.29. The difference between the facies ratio of Training image and that of a Realization could be explained by the fact, that coarse Patterns are interlayered imperfectly because of Pattern insufficiency (Fig. 10, 11, 12), what make the algorithm link parts of different channels somehow in the consequent iterations, adding more sand for the links between two channels.

It should be stressed also, that SIMPAT M algorithm is not a concurent one to the basic SIMPAT algorithm. The SIMPAT M algorithm uses an alternative method of the Pattern Data Base organization and the search procedure, which can be used in conjunction with that of the basic SIMPAT algorithm. This direction of the development could give a significant increase in performance of the algorithm.

Summary

It has been implemented the basic SIMPAT algorithm, based on the dual template technique, skipping a number of cells during the comparison between a Pattern from the Pattern Data Base and a Data event drawn from a partially informed Realization.

It has been implemented the SIMPAT M algorithm, based on a new technique (probably actively used in the Image processing and Pattern recognition), based on the Coarsening Transformation procedure and tree-like Pattern Data Base (with a tree depth equal to 2).

The comparison between the algorithms shows a good performance (a little calculation time) for the SIMPAT M algorithm, but SIMPAT one is still quicker. Probably the future development will reduce the calculation time for the SIMPAT M algorithm (see below).

Prospects

The main purpose of the following work is to create a quick algorithm for the facies modelling based on the Training images. So that, one of the foreground tasks is to improve the algorithm performance, and 3D hierarchical multifacies simulation is better to model after the cernel functions of the algorithm will be finally developed and optimized.

The optimization is planned to conduct in two ways. One way to reduce the calculation time is to increase the tree depth from 2 (used now) to 3 or more for large Templates. Probably it will be necessary to implement a function, calculationg the most optimal tree configuration (tree depth, level of coarsening at each “floor” of the tree, etc). Another way and/or additional step of the optimization is to merge the technique of dual templates [1] and alternative one, considered in the following work.

It is also planned to develop the extension version of the SIMPAT M algorithm based on an idea of the hierarchical facies simulation [2]. According to that idea one or two facies, e.g. floodplain shales, comprizes background facies, which are modelled at the end of the facies simulation, filling the empty space between the basic facies such as fluvial channels.

References

[1] Guven Burc Arpat, *Sequential Simulation with Patterns*, PhD thesis, Stanford University, U.S.A., January 2005
http://pangea.stanford.edu/~jcaers/theses/thesisBurcArpat.pdf

[2] A. Maharaja, A. Journel, *Hierarchical simulation of multiple-facies reservoirs using multiple-point geostatistics*, SPE 95574, 2005 SPE Annual Technical Conference and Exhibition, Dallas, Texas, U.S.A., 9-12 October 2005
http://www.spe.org/elibinfo/eLibrary_Papers/spe/2005/05ATCE/SPE-95574-MS/SPE-95574-MS.htm