

Titre: Une nouvelle approche pour la reconstruction statistique des images : le rapiéçage de motifs stochastiques
Title: images : le rapiéçage de motifs stochastiques

Auteur: Ayoub El Ouassini
Author:

Date: 2006

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: El Ouassini, A. (2006). Une nouvelle approche pour la reconstruction statistique des images : le rapiéçage de motifs stochastiques [Master's thesis, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/7881/>
Citation:

Document en libre accès dans PolyPublie Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7881/>
PolyPublie URL:

Directeurs de recherche: Antoine Saucier, & Denis Marcotte
Advisors:

Programme: Unspecified
Program:

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

52

This reproduction is the best copy available.

UMI[®]

UNIVERSITÉ DE MONTRÉAL

UNE NOUVELLE APPROCHE POUR LA RECONSTRUCTION STATISTIQUE DES
IMAGES : LE RAPIÉÇAGE DE MOTIFS STOCHASTIQUES

AYOUB EL OUASSINI

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)

MAI 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 978-0-494-19297-9

Our file *Notre référence*

ISBN: 978-0-494-19297-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

**
Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

UNE NOUVELLE APPROCHE POUR LA RECONSTRUCTION STATISTIQUE DES
IMAGES : LE RAPIÉÇAGE DE MOTIFS STOCHASTIQUES

présenté par: EL OUASSINI Ayoub

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. BOYER Francois-Raymond, Ph.D., président

M. SAUCIER Antoine, Ph.D., membre et directeur de recherche

M. MARCOTTE Denis, Ph.D., membre et codirecteur de recherche

M. FAVIS, Basil D., Ph.D., membre et codirecteur de recherche

M. LEFEBVRE Mario, Ph.D., membre

REMERCIEMENTS

Je tiens à remercier en premier lieu mon directeur de recherche M. Antoine Saucier pour m'avoir donné l'opportunité de travailler sur ce projet original. Je le remercie pour sa disponibilité, son encadrement et l'attention particulière qu'il m'a accordée tout au long de ma maîtrise.

Je tiens également à remercier mon codirecteur de recherche M. Denis Marcotte pour l'aide précieuse qu'il m'a apportée lors de mes travaux ainsi que pour sa personnalité agréable.

Ma reconnaissance se porte également envers mon codirecteur M. Basil Favis pour son amabilité et pour nous avoir présenté le problème qui est à l'origine de ce projet de maîtrise. Je remercie également M. Pierre Sarazin du laboratoire de recherche en polymère appliquée pour sa collaboration.

Je tiens à remercier l'ensemble de mes directeurs de recherche pour leur soutien financier.

Un remerciement à l'ensemble des professeurs de l'École Polytechnique que j'ai eu l'honneur de rencontrer.

Finalement, je tiens à remercier ma famille ainsi que mes amis pour leur soutien pendant les périodes difficiles et avec qui j'ai partagé des moments très agréables.

RÉSUMÉ

Dans ce projet, on propose deux nouvelles approches de reconstruction stochastique d'images bidimensionnelles. Notre objectif est de produire des simulations ayant les mêmes propriétés statistiques que celles observées dans l'image de référence. En particulier, un de nos principaux objectifs est la reproduction de la microgéométrie complexe des alliages de polymère à phase cocontinue. La caractérisation géométrique de ces alliages est cruciale car elle détermine les propriétés physiques du matériau. On accordera une attention particulière à la propriété de connectivité qui détermine la perméabilité et la conductivité électrique de l'alliage.

Nous avons développé dans un premier temps une approche multi-échelle qui vise à reproduire les mêmes propriétés que l'image de référence et ce à toutes les échelles de l'image. Plus précisément cette approche reproduit aux échelles diadiques de l'image les distributions de probabilité des porosités arrondies de l'image de référence. Cependant cette première approche a plusieurs inconvénients, le plus important étant la perte d'information liée au seuillage des porosités à toutes les échelles de l'image. Nous avons choisi d'écartier cette première approche à cause de la qualité des simulations obtenues à partir d'une image de polymère. Ces simulations montrent en effet une perte de porosité importante comparativement à l'image de référence. Toutefois l'approche multi-échelle reste prometteuse car bien des possibilités n'ont pas encore été explorées.

Par la suite nous avons développé une seconde approche mono-échelle plus simple qui a produit de biens meilleurs résultats que la précédente. Cette approche se base sur l'estimation de probabilités conditionnelles entre blocs de pixels adjacents de l'image de référence. Les simulations sont générées en juxtaposant séquentiellement les blocs de pixels observés dans l'image de référence, d'où le terme rapiéçage de motifs stochastique. Les différents blocs simulés sont choisis conformément aux probabilités

conditionnelles estimées. Nous avons comparé notre approche à l'algorithme de simulation multipoints de Strebelle sur plusieurs types d'images de complexité croissante. Pour les images composées de petits objets (damiers, disques, bâtonnets) relativement à la taille du bloc, notre approche a produit de meilleurs résultats que celle de Strebelle. Cependant la qualité des simulations produites se détériore lorsque la taille des objets à reproduire dépasse celle du bloc utilisé. Pour les images contenant des objets de grande taille relativement à la taille de l'image, tels que les canaux, notre approche performe aussi bien que l'algorithme de Strebelle. Cependant les simulations d'images d'un alliage de polymère bien que ressemblantes à l'image de référence ne reproduisent pas correctement sa distribution de longueur de cordes (CLD). L'algorithme de Strebelle par contre reproduit correctement la CLD de l'image de référence.

ABSTRACT

We present two new stochastic simulation approaches in which we focus on the accurate reproduction of the geometrical properties observed in the reference image. In particular, our approaches aim at reproducing correctly the complex micro- geometry of co-continuous polymer blends. We give a special attention to the connectivity property because it determines important physical properties of the material.

We developed a multiscale approach which aims at reproducing the same properties at each scale of the image. The multiscale approach reproduces the distribution probabilities of porosities at the dyadic scales of the reference image. The main disadvantage of this approach is the loss of information due to the use of a threshold on the porosities. We decided to draw aside this approach because of the simulation quality obtained on polymer images. Indeed, we observed an important porosity loss in these simulations compared to the reference image porosity. However, the multiscale approach remains a promising approach and many possibilities were not explored yet.

The second approach that we developed is simpler and produces much better results than the previous one. Our method is based on the analysis of statistical relationships between adjacent square pieces of image called blocks. We estimate the transition probabilities between adjacent blocks of pixels in a training image. The simulations are constructed by juxtaposing one by one square blocks of pixels, hence the term patchwork simulations. The probabilistic link between adjacent blocks of pixels is made via the transition probabilities. We compared the performance of patchwork simulations with Strebelle's multipoint simulation algorithm on several types of images of increasing complexity. For images composed of clusters which are small with respect to the block size (e.g. squares, discs, sticks), our patchwork approach produces better results than Strebelle's method. The most noticeable improvement is that the cluster geometry is

usually reproduced accurately. The accuracy of the patchwork approach is limited primarily by the block size. Clusters which are significantly larger than the block size are usually not reproduced accurately. For training images containing objects which are much larger than the block size, e.g. channels which span the entire image, the patchwork approach performs as good as Strebelle's algorithm. However, simulations of polymer blends could not reproduce accurately the cord length distribution (CLD) of the reference image. On the contrary, Strebelle's algorithm succeeds in reproducing the CLD of the reference image.

TABLE DES MATIÈRES

REMERCIEMENTS.....	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES FIGURES	xii
LISTE DES ABRÉVIATIONS	xiv
LISTE DES ANNEXES	xv
CHAPITRE 1 INTRODUCTION.....	1
CHAPITRE 2 REVUE CRITIQUE DE LITTÉRATURE	3
CHAPITRE 3 ORGANISATION DU TRAVAIL.....	6
CHAPITRE 4 Article 1: A patchwork approach to stochastic simulation: a route towards the analysis of morphology in multiphase systems.....	7
4.1 Introduction.....	9
4.2 Description of the patchwork simulation method	12
4.2.1 Preliminary definitions	12
4.2.2 Transition probability matrices	13
4.2.3 Sequential simulation algorithm.....	14

4.2.3.1	Initialization	15
4.2.3.2	Simulating the first line of blocks with the first transition type.....	15
4.2.3.3	Simulating the blocks in the first column with the second transition type	17
4.2.3.4	Simulating the inner blocks of the image with the third transition type..	
	17
4.2.3.5	The case of unobserved states	17
4.3	Simulation results.....	22
4.3.1	Simulations using checkerboards as training images	22
4.3.2	Simulations using training images composed of discs	24
4.3.2.1	Discs of constant diameters.....	24
4.3.2.2	Discs of variable diameters	28
4.3.3	Simulations using training images composed of sticks	29
4.3.4	Simulations using a training image containing channels.....	33
4.3.5	Simulations based on an electron microscope polymer blend image.....	35
4.4	Discussion.....	40
4.5	Conclusion	41
References.....		43

CHAPITRE 5 APPROCHE MULTI-ÉCHELLE	47
5.1 Description de la méthode	47
5.1.1 Définitions préliminaires	48
5.1.2 Description des matrices de probabilités de transition	50
5.1.3 Fonctionnement du modèle.....	52
5.2 Résultats.....	56
5.2.1 Simulations tests.....	56
5.2.2 Simulations de disques	59
5.2.3 Simulations avec une image de polymère.....	61
5.3 Synthèse	63
CHAPITRE 6 SYNTHÈSE et CONCLUSION	64
6.1 Discussion.....	64
6.2 Conclusion et Recommandations.....	66
RÉFÉRENCES BIBLIOGRAPHIQUES.....	68
ANNEXES	73

LISTE DES FIGURES

Figure 4.1	Transition probabilities between adjacent blocks of size n pixels 13
Figure 4.2	The three configurations of start and arrival states used in the patchwork.. algorithm 14
Figure 4.3	Illustration of the use in a simulation of the three transition types 16
Figure 4.4	Spatial weight distributions used for the metric..... 19
Figure 4.5	Comparison between three simulations using different approaches for the correction of unobserved states 21
Figure 4.6	Simulations using checkerboards with varying mesh size as training images 23
Figure 4.7	Simulations based on training images composed of discs with a fixed diameter D 25
Figure 4.8	Patchwork simulations based on training images composed of discs with a fixed diameter D 27
Figure 4.9	Simulations based on training images composed of discs with a varying diameter D 29
Figure 4.10	Simulations based on training images composed of sticks of a fixed length L 31
Figure 4.11	Simulations based on training images composed of sticks of varying length L 32
Figure 4.12	Simulations of channels..... 34

Figure 4.13	Electron microscope image of a co-continuous polymer blend.....	36
Figure 4.14	Simulations using a polymer blend microscopic image as a training image	37
Figure 4.15	Cord length distributions of the four images in Figure 4.14:.....	39
Figure 5.1	Schématisation des échelles diadiques	48
Figure 5.2	Exemple de configuration de porosités évaluées à deux échelles L_n et L_{n+1}	49
Figure 5.3	Représentation des diagrammes utilisés par le modèle multi-échelle	51
Figure 5.4	Initialisation de la simulation.....	52
Figure 5.5	Étape 1 : utilisation du diagramme #1	53
Figure 5.6	Étape 2 : utilisation du diagramme #2.....	54
Figure 5.7	Étape 3 : utilisation du diagramme #3.....	54
Figure 5.8	Étape 4 : utilisation du diagramme #2.....	55
Figure 5.9	Porosités simulées à l'échelle L_3	56
Figure 5.10	Simulations du modèle multi-échelle sur des images de damiers.....	58
Figure 5.11	Simulations du modèle multi-échelle sur des images de disques.....	60
Figure 5.12	Image de l'alliage de polymère	61
Figure 5.13	Simulations d'une image de polymère avec le modèle multi-échelle	62

LISTE DES ABRÉVIATIONS

CLD	:	Cord Length Distribution
CPU	:	Central Processing Unit
PCL	:	Polycaprolactone
PS	:	Polystyrene
RAM	:	Random Access Memory
SEM	:	Scanning Electron Mircoscopy
SNE	:	Single Normal Equation
SGGMS	:	Stanford Geostatistical Modeling Software
SGSIM	:	Sequential Gaussian Simulation
SISIM	:	Sequential Indicator Simulation

LISTE DES ANNEXES

ANNEXE I : Code de la méthode : courtepointe stochastique	73
---	----

CHAPITRE 1 INTRODUCTION

Les simulations stochastiques sont utilisées pour modéliser une grande variété de systèmes complexes et aléatoires tels que les réservoirs de pétrole [19], les nappes phréatiques [6], les milieux poreux [1, 2], ou encore les matériaux composites [34]. Nous accordons une importance particulière à la propriété de connectivité de ces systèmes car elle est souvent liée à des propriétés physiques importantes. On peut citer les exemples suivants : la perméabilité d'un milieu poreux est directement liée au degré de connectivité de ses pores; la conductivité électrique d'un matériau composite dépend du degré de connectivité entre les phases conductrices du matériau; les caractéristiques de l'écoulement dans un réservoir de pétrole sont liées au degré de connectivité entre les grands canaux qui le constituent; enfin les propriétés physiques de perméabilité et de conductivité d'un alliage de polymère co-continue [20, 26, 27] sont déterminées principalement par le degré de connectivité de chacune des phases qui le composent. La reconstruction de la microgéométrie complexe des alliages de polymères à phases cocontinues constitue un de nos objectifs.

En effet, les alliages de polymères sont des matériaux performants et peu coûteux à développer, ce qui explique l'importance économique grandissante de ce type de matériaux. Cependant, les alliages de polymères à deux phases cocontinues où la proportion de chaque phase est à peu près la même ont été peu étudiés. Dans ce type d'alliage chaque phase est interconnectée à elle-même à travers un réseau complexe. Les images de microscopie de coupes de ce matériau ont révélé une micro-géométrie complexe et difficile à caractériser. Le contrôle de cette micro-géométrie est crucial car il détermine les propriétés du matériau (perméabilité et conductivité électrique)

Dans ce mémoire, on propose une méthode séquentielle de simulation stochastique afin de reproduire de telles micro-géométries complexes. Notre approche a été conçue afin de reproduire la connectivité locale de l'image de référence. La méthode

proposée repose sur l'extraction de probabilités conditionnelles entre blocs de pixels composant l'image de référence. Ces probabilités sont ensuite utilisées pour générer des simulations statistiquement semblables à l'image de référence. Cette méthode mono-échelle est décrite en détail dans l'article intégré à ce mémoire au chapitre 4.

Nous avons développé également une méthode de simulation multi-échelle qui utilise aussi des probabilités conditionnelles extraites de l'image de référence. L'objectif de la méthode est de produire des simulations ayant les mêmes propriétés statistiques que l'image de référence et ce à toutes les échelles de l'image. Cette seconde approche qui est décrite dans le chapitre 5 donne cependant des résultats qui ne sont pas à la hauteur de sa complexité.

Le chapitre 6 fait une synthèse générale de l'ensemble des résultats obtenus à l'aide de ces deux approches et présente quelques recommandations pour la suite de ce projet.

CHAPITRE 2 REVUE CRITIQUE DE LITTÉRATURE

Dans cette section, on fait un bref historique des techniques de simulations stochastiques séquentielles.

Les simulations séquentielles ont longtemps été basées sur les statistiques à deux points. Leur objectif était de produire des simulations ayant les mêmes variogramme et histogramme que ceux observés dans les images de référence. La simulation gaussienne séquentielle et la simulation séquentielle d'indicatrices [12] sont des exemples de méthodes performantes qui reproduisent les statistiques à deux points. Cependant, ces méthodes ne permettent pas de reproduire correctement des structures géométriques complexes et interconnectées tels que les méandres fluviaux ou encore les milieux poreux. En effet, la caractérisation de géométries aussi complexes requiert l'utilisation de statistiques basées sur plus de deux points.

Pour caractériser la morphologie complexe des milieux poreux en particulier, plusieurs modèles ont été développés. On peut citer les modèles en réseau [8, 15], les modèles de grain [30], les modèles de percolation [11, 15], les modèles diagénétique [5, 23], ou encore les modèles multi-fractales [18, 24, 27, 28]. Cependant, ces modèles se basent sur des hypothèses concernant la géométrie des pores ou encore sur l'existence d'une propriété d'invariance d'échelle. Ils n'offrent pas la possibilité de générer directement des simulations en extrayant l'information d'images de référence quelconque.

Parmi les approches de simulations multipoints, on peut citer l'approche multi-fractale de Saucier et Muller [28, 30]. Cette approche repose sur la reproduction des moments multi-échelle mesurés à partir de l'image de référence. Les images simulées montrent que la caractérisation par moments multi-échelles permet de reconstruire les images contenant des objets à faible densité et de petite taille comparativement à la taille

de l'image. Cependant, lorsque la densité des objets augmente, la qualité des reconstructions se dégrade. De plus, cette méthode ne permet pas de reproduire avec précision les propriétés d'anisotropie et surtout de connectivité de l'image de référence.

Une autre méthode de simulation multipoints est l'approche basée-objet. Cette méthode repose sur la paramétrisation des géométries que l'on désire reproduire. Les objets paramétrés sont générés aléatoirement pour produire des simulations stochastiques [30, 17, 21]. La principale faiblesse de cette approche est la difficulté de paramétriser certaines géométries complexes. De plus, il est difficile de générer des simulations conditionnelles avec cette méthode.

La méthode de recuit simulé ('simulated annealing') est également utilisée dans les simulations multipoints [13, 14]. Cette méthode itérative consiste à définir une fonction objectif qui est optimisée. Dans le cadre des simulations multipoints, la fonction objectif inclut plusieurs termes afin de reproduire les statistiques multipoints désirées. Cependant, dans cette approche itérative la convergence n'est pas toujours assurée et le taux de convergence comme le critère d'arrêt sont souvent inconnus. Aussi, le nombre de termes inclus dans la fonction objectif croît avec le nombre de statistiques multipoints à reproduire, ce qui requiert des ressources CPU et RAM considérables.

L'approche multipoints établie par Guardiano et Srivastava [16] se distingue par son originalité. Leur méthode est en fait une simulation séquentielle d'indicatrices qui utilise des probabilités conditionnelles directement extraites de l'image de référence. En plus de sa simplicité, cette approche a l'avantage de permettre des simulations conditionnelles. De plus, la méthode n'est pas itérative contrairement à l'approche Monte Carlo établie par Caers et Journel [9, 10]. Cependant cette méthode n'était pas utilisée en pratique car la programmation n'était pas suffisamment performante pour générer des simulations dans un délai raisonnable. C'est Strebelle qui permettra la mise en œuvre pratique de la méthode établie par Guardiano et Srivastava avec son algorithme 'single normal equation' (SNE) [32, 33]. Nous avons utilisé le logiciel

SGEMS [22] afin de générer des simulations utilisant l'algorithme SNE. Actuellement, l'algorithme de Strebelle est le plus utilisé pour les simulations multipoints étant donné sa vitesse d'exécution et la qualité des simulations produites. Cependant, les simulations de Strebelle se basent uniquement sur les données observées dans l'image d'entraînement. Lorsque des configurations multipoints jamais observées apparaissent lors d'une simulation, l'algorithme réduit le nombre de points jusqu'à rencontrer une configuration multipoints déjà observée dans l'image d'entraînement. Cette manière de procéder occasionne une perte de qualité des simulations liée à la réduction du nombre de points utilisés.

Une autre approche consiste à utiliser des configurations de points similaires à celles observées dans l'image de référence. Cette approche a été utilisée par Arpat [4] et Breiman [7] dans leurs méthodes de classification par patrons et réseaux de neurones. Bien que les résultats préliminaires obtenus par cette nouvelle méthode soient prometteurs, il reste néanmoins des questions importantes à élucider. Entre autres, la formulation de l'algorithme pour les reconstructions stochastiques et son adaptation aux simulations conditionnelles. Les recherches à venir visent à mieux comprendre comment utiliser cette approche dans le cadre de simulations stochastiques.

CHAPITRE 3 ORGANISATION DU TRAVAIL

Dans ce projet, notre objectif a été de produire des simulations stochastiques qui reproduisent au mieux les propriétés de l'image de référence et en particulier la propriété de connectivité entre objets de l'image. Afin d'atteindre cet objectif, nous avons commencé par travailler sur une approche de simulation multi-échelle. Le chapitre 5 décrit plus en détail cette première méthode et présente quelques-uns des résultats obtenus. Étant donné la qualité des simulations obtenues d'une part, et la complexité de l'approche d'autre part, cette méthode a été finalement écartée. Nous avons par la suite développé une autre méthode de simulation séquentielle qui, malgré sa simplicité, a été plus fructueuse que la précédente. Le chapitre 4 présente l'article intégré décrivant cette seconde approche de simulation. Dans cet article intitulé *A patchwork approach to sequential stochastic simulation : a route towards the analysis of morphology in multiphase systems* on décrit en détail cette méthode et on compare notre approche avec celle de Strebelle sur plusieurs types d'images. Notez que le chapitre 4 est identique du point de vue du contenu à l'article que nous avons soumis pour publication à la revue *Chaos, solitons and fractals*, seule la forme de présentation a changé (numéros de sections et de figures, fonte, etc.). Le chapitre 6 fait une synthèse de l'ensemble des résultats obtenus avec les deux approches et présente quelques recommandations pour la suite du projet.

CHAPITRE 4 Article 1: A patchwork approach to stochastic simulation: a route towards the analysis of morphology in multiphase systems

Ayoub El Ouassini

École Polytechnique de Montréal, C.P. 6079, Station centre-ville, Montréal (Québec), Canada, H3C-3A7. e-mail: ayoub.el-ouassini@polymtl.ca

Antoine Saucier ^b

^b Corresponding author.

École Polytechnique de Montréal, département de mathématiques et de génie industriel, C.P. 6079, Station centre-ville, Montréal (Québec), Canada, H3C-3A7. tel: (1) 514-340-4711 # 4516. Fax: (1) 514-340-4463. e-mail: antoine.saucier@polymtl.ca.

Denis Marcotte

École Polytechnique de Montréal, département de génie civil, géologique et minier, C.P. 6079, Station centre-ville, Montréal (Québec), Canada, H3C-3A7. tel: (1) 514-340-4711 # 4620. Fax: (1) 514-340-4191. e-mail: denis.marcotte@polymtl.ca

Basil D. Favis

École Polytechnique de Montréal, département de génie chimique, C.P. 6079, Station centre-ville, Montréal (Québec), Canada, H3C-3A7. tel: (1) 514-340-4711 # 4527. Fax: (1) 514-340-4159. e-mail: basil.favis@polymtl.ca

Abstract

We propose a new sequential stochastic simulation approach for black and white images in which we focus on the accurate reproduction of the small scale geometry. Our approach aims at reproducing correctly the connectivity properties and the geometry of clusters which are small with respect to a given length scale called block size. Our method is based on the analysis of statistical relationships between adjacent square pieces of image called blocks. We estimate the transition probabilities between adjacent blocks of pixels in a training image. The simulations are constructed by juxtaposing one by one square blocks of pixels, hence the term patchwork simulations. We compare the performance of patchwork simulations with Strebelle's multipoint simulation algorithm on several types of images of increasing complexity. For images composed of clusters which are small with respect to the block size (e.g. squares, discs, sticks), our patchwork approach produces better results than Strebelle's method. The most noticeable improvement is that the cluster geometry is usually reproduced accurately. The accuracy of the patchwork approach is limited primarily by the block size. Clusters which are significantly larger than the block size are usually not reproduced accurately. As an example, we applied this approach to the analysis of a co-continuous polymer blend morphology as derived from an electron microscope micrograph.

Key words: sequential stochastic simulations, transition probabilities, local geometry, connectivity, polymer blends morphology.

PACS:

4.1 Introduction

Stochastic simulations are used to model a wide variety of random systems and processes such as oil reservoirs [15], groundwater flow [5], porous media [1] [2] and composite materials [30]. In this paper, we propose a new stochastic simulation approach in which we focus on the accurate reproduction of the small scale geometry. More precisely, our approach aims at modelling correctly the connectivity properties and the geometry of clusters which are small with respect to a given fixed length scale.

Our focus is motivated by the fact that morphology and connectivity often play a central role in the determination of physical properties which are of interest in various contexts. Here are a few examples: the absolute permeability of a porous medium is directly related to the degree of pore connectivity; the electrical conductivity of a composite material depends on the degree of connectivity of the conductive regions within the material; the large scale fluid flow properties of an oil reservoir can be determined by the degree of connectivity of large scale channels; finally, the physical properties (e.g. permeability to gas or liquids, electrical conductivity) of a co-continuous polymer blend [16] are essentially determined by the morphology of the two intertwined phases composing the blend.

In the context of porous media, some of the approaches which have been used to describe the morphology are network models [6] [11], grain models [27], percolation models [9] [21], diagenetic models [4] [19] and fractal models [14] [20] [25] [24]. These models rely either on specific assumptions about the pore geometry or on the existence of a scale invariance property. They do not provide a data-adaptive framework in which the simulation can be performed by extracting information from arbitrary training images.

Sequential simulation is one of the most widely used techniques in stochastic simulations. Sequential simulation techniques have long been based on two-point

statistics. Efficient algorithms have been developed such as the sequential Gaussian simulation (SGSIM), or the sequential indicator simulations (SISIM) [10]. However, these methods cannot reproduce accurately complex and connected structures such as sinuous channels. Indeed SGSIM and SISIM reproduce only the histogram and the variogram, which is insufficient to characterize complex geometries.

The accurate reproduction of complex random geometries may require more than two-point statistics. In the context of a multipoint statistical characterization based on multiscale moments [26], Saucier and Muller showed that an accurate reproduction of multiscale moments does not necessarily lead to an accurate reproduction of cluster shapes and connectivity. The difficulty is that subtle morphological features are not necessarily captured by a given statistical characterization, e.g. multiscale moments.

Complex images can be simulated with the object-based approach, which is based on the parametrization of shapes corresponding to desired geometries, e.g. sinuous channels. These parameters are randomized to generate random geometries within a specified volume [26][13][17]. The main limitation of this approach is the difficulty to parametrize complex geometries. Moreover, the conditioning of the random parameters to local data can be difficult to achieve.

In 1993, Guardiano and Srivastava [12] proposed a new algorithm based on multiple-point statistics. The method proposed is a sequential indicator simulation exploiting conditional probabilities extracted directly from a training image. The advantage of this approach compared with the object-based approach is the possibility to introduce conditional data in a simulation. Moreover, the approach is not iterative unlike other multiple-point approaches such as the Markov Chain Monte Carlo simulation of Caers and Journel [7, 8]. However, the original code was computationally demanding and was not used in practice. Strebelle [28] [29] recently proposed a practical algorithm based on Guardiano and Srivastava's idea. This algorithm allowed the practical use of

multiplepoint statistics in simulations. Strebelle uses only the data observed in the training image to simulate possible realizations. In the course of a simulation, if one encounters a configuration of points which has not been observed in the training image, then the number of points is reduced iteratively until an observed configuration of points is found. Another approach proposed by Arpat [3] deals with unobserved configurations of points by modeling and interpolating between existing configurations in the training image.

In this paper, we present a new sequential simulation method which is based on the analysis of the statistical relationships between adjacent square pieces of image called blocks. More precisely, we estimate the conditional probability to observe a block knowing a set of adjacent blocks. These conditional probabilities are regarded as transition probabilities between adjacent sets of blocks. The statistical information contained in the training image is stored in the transition probabilities.

Our simulations are performed in two steps. Firstly, we scan the reference image to estimate the transition probabilities. Secondly, we use these probabilities to simulate the image block by block, linking each new square block to the previously simulated blocks. The linking procedure is designed to reproduce small scale structures and local connectivity properties as accurately as possible. This block based process resembles the fabrication of a quilt, hence the term patchwork simulation. This approach can reproduce correctly many morphological features because the blocks used in the patchwork process, which are drawn directly from the training image, are essentially correct.

This paper is structured as follows. In section 2, we describe the patchwork simulation method. In section 3, we apply our patchwork method to several types of training images and we compare the results with Strebelle's multipoint simulation algorithm [28] [29]. We consider synthetic images composed of isolated clusters (squares, disks, sticks, channels) that mimic the morphology of two-phase systems for which one phase is

predominant. Finally, we apply this approach to the specific case of a co-continuous polymer blend morphology.

4.2 Description of the patchwork simulation method

4.2.1 Preliminary definitions

Our method uses black and white images, i.e. each pixel of the reference image is coded on a single bit (1 or 0). The proportion of black pixels in an image will be called the image porosity. A square region in the image is called a block. Each block of size n pixels is composed of n^2 pixels, where $n > 0$ is an integer. A state S_i is a unique configuration of pixels observed in a block or in a set of several adjacent blocks in the image. A state S_i is represented by a sequence of bits such as

$$S_i = \underbrace{1010 \dots 100}_{bxn^2} \underbrace{10101}_1$$

where n is the block size and b is the number of adjacent blocks. In Figure 4.1, the three adjacent blocks represented by circles correspond to a start state S_i coded on $3n^2$ bits, whereas the single block represented by a cross corresponds to an arrival state S_j coded on n^2 bits. We will say that the two states S_i and S_j are complementary because their association is a quadruplet of adjacent blocks which is observed in the reference image. Our method is based on the estimation of transition probabilities P_{ij} between such

complementary states. The transition probabilities $P_{i,j} \geq 0$ satisfy $\sum_{j=1}^{2^{3n^2}} P_{i,j} = 1$ for each i .

These probabilities are estimated by scanning a single or several reference images. For each image, the scanning can be done in a single direction or in the four directions of the image, i.e. directions obtained by rotating the reference image by 0, 90, 180 and 270 degrees. If the reference image is statistically isotropic, then we scan it in the four directions to improve the precision of our estimates of the transition probabilities. If the

reference image is statistically anisotropic, then the scanning is performed in a single direction. In addition to the transition probabilities between adjacent blocks, we also estimate the probability of occurrence P_i of each observed start state. This information is used to correct unobserved states as we will see in section 4.2.3.5.

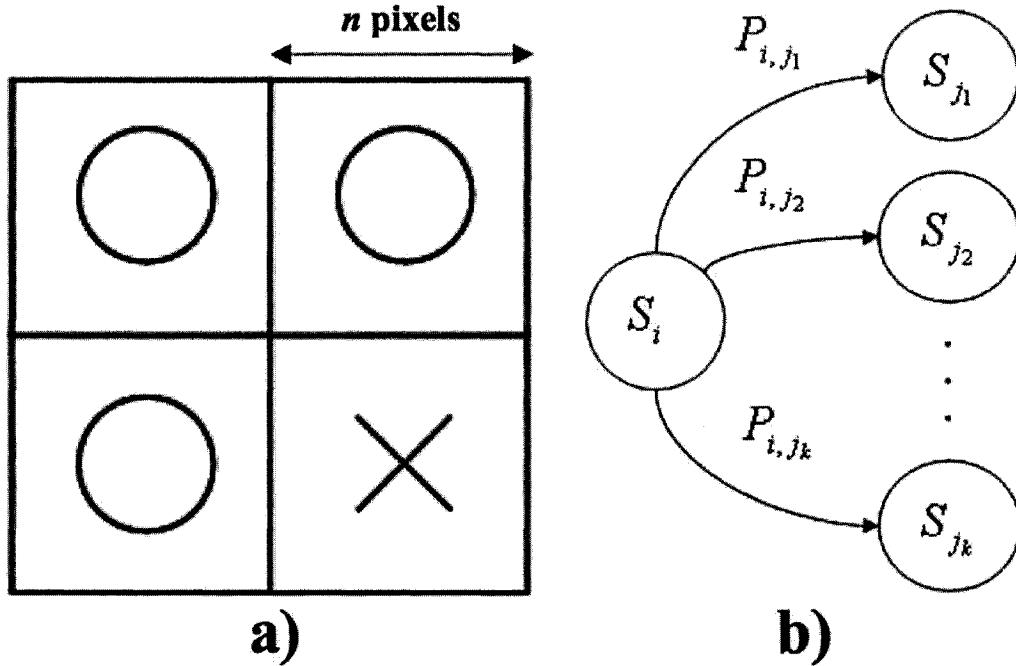


Figure 4.1 Transition probabilities between adjacent blocks of size n pixels: a) set of four adjacent blocks; b) transition probabilities between a start state S_i and several possible arrival states $S_{j_1}, S_{j_2}, \dots, S_{j_k}$. The start state S_i represented by the three circles in a) is coded on $3n^2$ bits. The arrival state represented by a cross in a) is coded on n^2 bits. The transition probability between the complementary states S_i and S_{j_n} is denoted by P_{i,j_n} , where $1 \leq n \leq k$.

4.2.2 Transition probability matrices

We use three types of transition probability matrices which are associated to different configurations of start and arrival states, as illustrated in Figure 4.2.

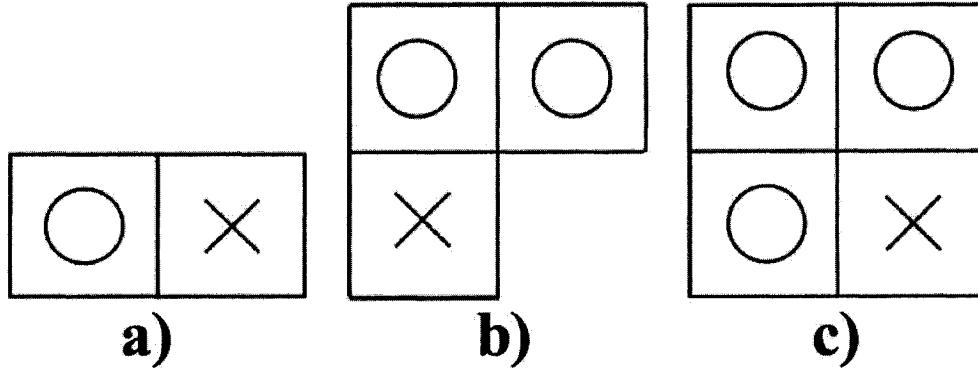


Figure 4.2 The three configurations of start and arrival states used in the patchwork algorithm. Each square represents an image block of length n pixels. Start and arrival states are represented by circles and crosses respectively: a) configuration used to simulate the blocks of the first line of the image; b) configuration used to simulate the blocks of the first column of the image; c) configuration used to simulate the rest of the blocks of the image.

In the following, a specific configuration of start and arrival states will be called a transition type. The first transition probability matrix associated with the transition type represented in Figure 4.2-a is used to simulate the first line of blocks in the image. It contains transition probabilities between start and arrival states both coded on n^2 bits. The second transition matrix associated with the transition type represented in Figure 4.2-b is used to simulate the first column of blocks in the image. This matrix contains transition probabilities between a start state coded on $2n^2$ bits and an arrival state coded on n^2 bits. Finally, the third and main transition probability matrix associated with the third transition type (Figure 4.2-c) contains transition probabilities between a start state coded on $3n^2$ bits and an arrival state coded on n^2 bits. This matrix is used to simulate all the remaining blocks of the image.

4.2.3 Sequential simulation algorithm

The three transition types represented in Figure 4.2 are used sequentially to construct an image, as illustrated in Figure 4.3. The construction is performed block by block. The construction order is left to right, top to bottom. In this section, we describe the

initialization of a simulation and we explain how the three transition types are used in the construction.

4.2.3.1 Initialization

The first step in a simulation consists in selecting randomly a block from the training image. This block, which is positioned in the upper left corner of the image in Figure 4.3-a, is the start state of the simulation.

4.2.3.2 Simulating the first line of blocks with the first transition type

The next step consists in simulating a linear chain of blocks by using sequentially the first transition type (Figure 4.2-a). The corresponding transition probability matrix is used to simulate the next block (cross in Figure 4.3-b)) located at the right side of the start state (black disc in Figure 4.3-b)). Next, we shift to the right and the previous arrival state turns into a start state (black disc in Figure 4.3-c)). We keep shifting to the right until all the blocks of the first line are simulated.

Note that the simulation of the first line of the image is a Markov chain. We emphasize that all pairs of adjacent blocks in the first line are observable configurations of pixels in the image, whereas a group of three or more adjacent blocks is not necessarily an observable configuration.

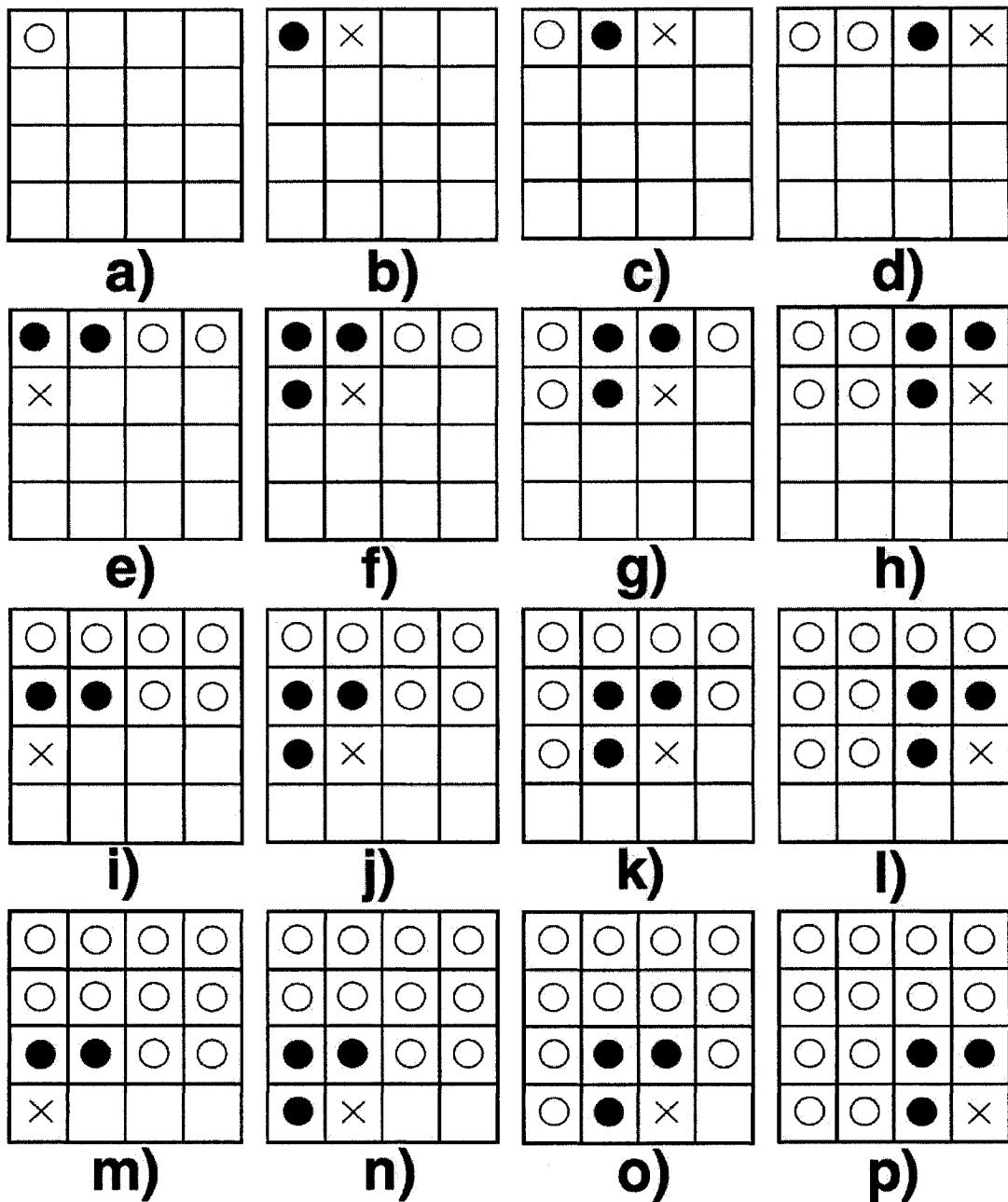


Figure 4.3 Illustration of the use in a simulation of the three transition types. White discs represent blocks which have already been simulated. Black discs represent blocks currently used to simulate a complementary block represented by a cross. The initialization step a) is followed in b) by the use of the first transition matrix (Figure 4.2-a) to simulate the blocks of the first line in b), c) and d). The second transition matrix (Figure 4.2-b) is used to simulate the blocks of the first column in e), i) and m). For the rest of the image, the third matrix (Figure 4.2-c) is used in f), g), h), j), k), l), n), o) and p).

4.2.3.3 Simulating the blocks in the first column with the second transition type

Once all the blocks of the first line are simulated, we shift to the second line to simulate the first block of the second line (the cross in Figure 4.3-e). The second transition type (Figure 4.2-b) is used to simulate the first block of the second line knowing the two first blocks of the first line (the two black discs in Figure 4.3-e). More generally, we use the second transition type for each block located in the first column (Figures 4.3-e, 4.3-i and 4.3-m), except for the initialization block. Note that all the blocks of the second line are simulated via the third transition type before the first block of the third line (Figure 4.3-i) is simulated.

4.2.3.4 Simulating the inner blocks of the image with the third transition type

For all the remaining blocks, i.e. blocks which are neither located in the first line nor in the first column, we use the third transition type (Figure 4.2-c). This transition is used to simulate a single block knowing three adjacent blocks as illustrated in Figures 4.3-f, 4.3-g, 4.3-h, 4.3-j, 4.3-k, 4.3-l, 4.3-n, 4.3-o and 4.3-p.

4.2.3.5 The case of unobserved states

Starting from the second line and third column in the simulation (Figure 4.3-g), we may observe the occurrence of states that do not exist in the training image (the three black discs in Figure 4.3-g). In Figure 4.3-c, the use of the first transition probability matrix ensures that the second pair of horizontal blocks in the top line is an observable configuration. In Figure 4.3-f, the use of the third transition probability matrix ensures that the second pair of vertical blocks is an observable configuration. However, the association of the two pairs of blocks, i.e. the three black discs in Figure 4.3-g, is not necessarily an observable configuration. This triplet of blocks can be either observed, unobserved or impossible to observe for this particular training image. This shows that our patchwork approach can generate new states, unlike Markov chains.

If an unobserved state S_U is encountered, then our strategy is to substitute it for an observed state S_O which is similar to the unobserved state. We quantify the degree of similarity between S_U and S_O in two complementary ways.

Firstly, the state S_O is chosen so that the porosities of S_O and S_U are comparable, i.e.

$$|\phi(S_O) - \phi(S_U)| \leq \varepsilon \phi(S_U), \quad (1)$$

where $\phi(S)$ denotes the porosity of a state S and ε is a tolerance ratio ($\varepsilon = 0.1$ usually gives good results). In the following, the application of the constraint (1) will be called porosity filtering. This filtering helps controlling the global porosity of the simulated image. In some cases, we will see that it can be advantageous to perform simulations without applying the porosity filtering. Secondly, we consider the set of all the observed states which satisfy the porosity constraint (1) and we choose in this set the state S_O which is closest to S_U , i.e. the state S_O which minimizes the distance $d(S_O, S_U)$ between S_O and S_U . We define $d(S_O, S_U)$ by the metric

$$d(S_O, S_U) = \sum_{k=1}^M w_k |S_O(k) - S_U(k)|, \quad (2)$$

where $S_O(k)$ and $S_U(k)$ denote the k^{th} bit of the states S_O and S_U respectively, M is the number of pixels on which the two states are coded, and w_k is the weight assigned to k^{th} bit. If no porosity filtering is used, then the substitution state is simply the state S_O which minimizes $d(S_O, S_U)$. If several observed states are at the same distance from the unobserved state, then we select randomly one of them according to their occurrence probability.

We design the spatial weight distribution to reduce the connectivity discontinuities between the substitute state and the blocks simulated previously. To reach this goal, we give more weight to the bits which are located close to previously simulated blocks. More precisely, the weights w_k decrease linearly with the distance between the k^{th} bit location and the border with previously simulated blocks, as shown in Figure 4.4. The weights w_k satisfy $1 \leq w_k \leq P_{\max}$, where $P_{\max} \geq 1$ is the maximal weight.

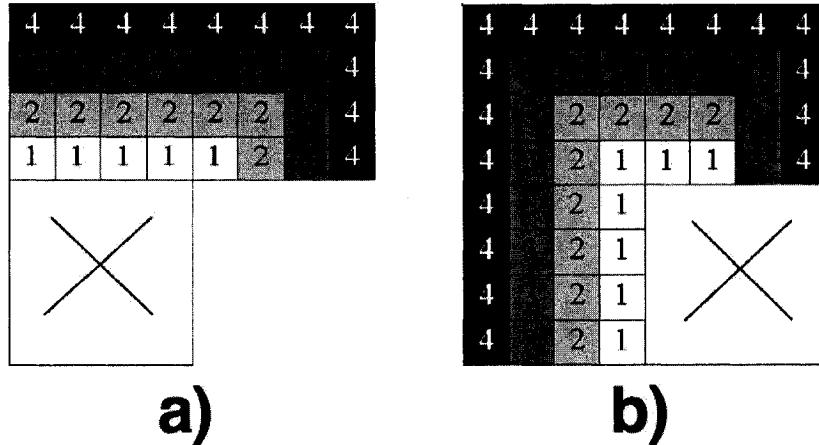


Figure 4.4 Spatial weight distributions used for the metric. The weights range from one to four: a) weight distribution for the transition type displayed in Figure 4.2.b); b) weight distribution for the transition type displayed in Figure 4.2.c). The weights decrease when the distance from the border with previously simulated blocks increases. This border is represented by a black strip.

In Figure 4.4, we illustrate the spatial weight distribution associated to the second and third transition types (Figures 4.2-b and 4.2-c). A maximum weight of four is assigned to the bits located at the outer frontier whereas a minimum weight of one is assigned to the bits located at the inner frontier.

The use of the metric (2) with non uniform weights for the correction of unobserved states has a significant impact on the simulations quality. The major effect of this metric is to reduce the breaking of the clusters which overlap between adjacent blocks. In that sense, using this metric helps respect the local connectivity properties. Figure 4.5 shows the improvement obtained in simulations obtained with a training image composed of discs. The algorithm used to construct the image displayed in Figure 4.5-a is described in section 4.3.2.

We present in Figure 4.5-b a simulation in which we substitute each unobserved state by a randomly chosen observed state. This simulation is clearly of less quality than the simulations of Figure 4.5-c and 4.5-d where our metric is used with a 10% porosity filtering, i.e. with $\varepsilon = 0.1$. The simulation in Figure 4.5-c uses a uniform weight

distribution, i.e. all the weights are equal to one, whereas the simulation in Figure 4.5-d uses a non-uniform weight distribution.

We observe that the number of deformed discs is much smaller with the non-uniform weights metric than with the uniform weights metric (i.e. 27 versus 81 respectively, including isolated pixels). All the simulations presented in the following use the non-uniform weights metric with $P_{max} = 4$.

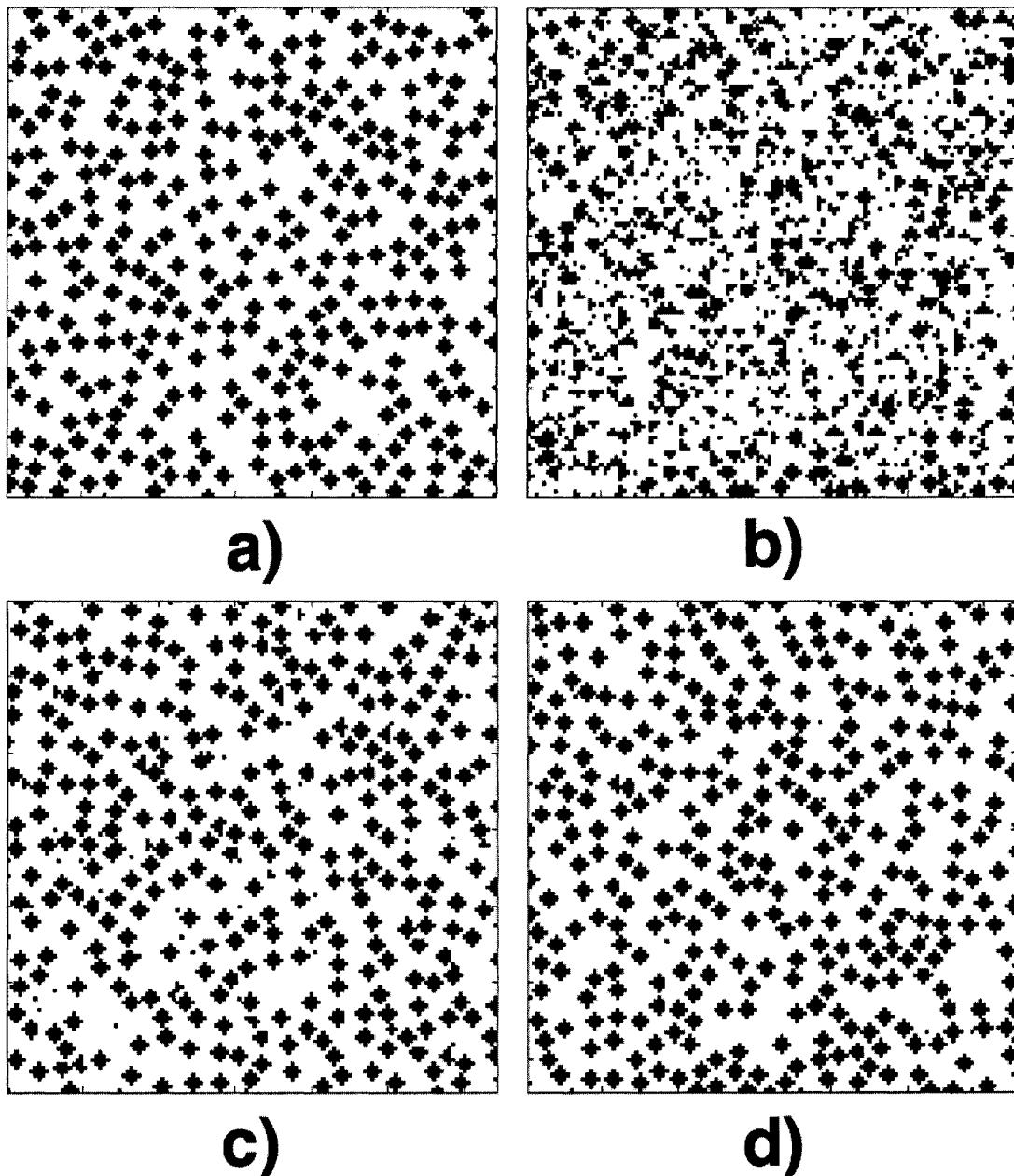


Figure 4.5 Comparison between three simulations using different approaches for the correction of unobserved states, using the training image displayed in a) which contains discs of diameter 4 pixels. In these simulations, we substitute each unobserved state by: b) an observed state chosen randomly; c) an observed state which is close to the unobserved state, using the metric with a uniform weight distribution; d) an observed state which is close to the unobserved state, using the metric with a non-uniform weight distribution such that $P_{max} = 4$. A 10% porosity filtering is used in c) and d).

4.3 Simulation results

In this section, we present simulation results obtained with our patchwork approach and with Strebelle's algorithm. The two approaches are tested and compared on several training images of increasing complexity.

4.3.1 Simulations using checkerboards as training images

For simple training images such as checkerboards, which are composed of blocks laid out regularly on a grid, we expect a priori our block based method to perform well because the number of possible states is relatively small. In Figure 4.6, we show simulations obtained with a checkerboard used as a training image. The size of the images displayed is 32x32 pixels whereas the size of the training images used is 256x256 pixels. Checkerboards are composed of black and white square meshes. In our simulations, the mesh size varies from one pixel to six pixels. We used the simulation model based on a 4x4 pixels grid without porosity filtering. The results displayed in the second column of Figure 4.6 show that the 4x4 model can reproduce perfectly checkerboards with a mesh size smaller than five pixels. We emphasize that our model has the ability to reproduce checkerboards with an offset effect. For checkerboards with a mesh size larger than four pixels (Figures 4.6-e and 4.6-f), the 4x4 model does not reproduce checkerboards perfectly. Indeed, we observe in the second column of Figures 4.6-e and 4.6-f the occurrence of several rectangles. Nevertheless, the black and white rectangles preserve the checkerboard visual appearance.

We tested Strebelle's algorithm on the same checkerboard images. In this paper, all the simulations based on Strebelle's algorithm were produced with the software SGEMS [18]. The simulations presented in the third column of Figure 4.6 were obtained by using a maximum of 60 conditioning points and 6 multigrids (these parameters were found to give the best results). The multigrid concept consists in simulating first the coarsest grid (large scale), then a refined grid conditional to the values already simulated on the coarse grid, and so on until the final desired grid is obtained.

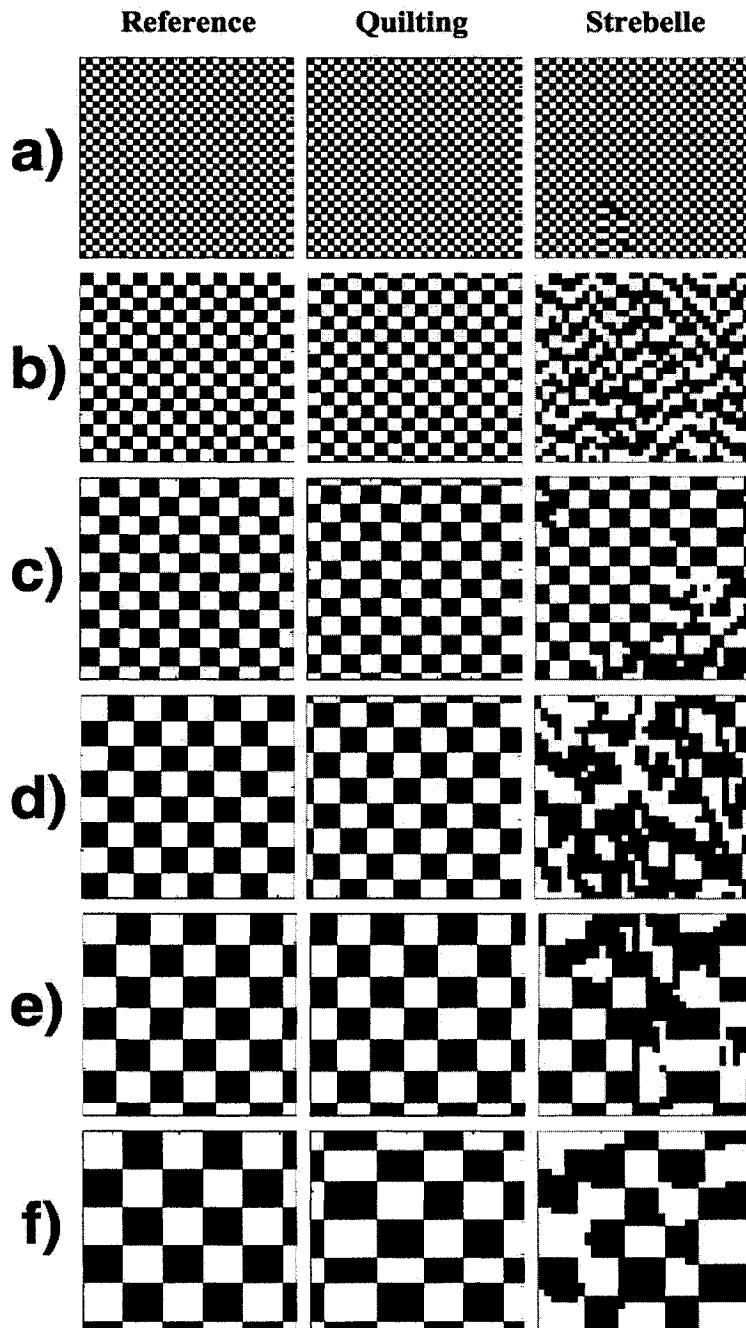


Figure 4.6 Simulations using checkerboards with varying mesh size as training images. For each of the six cases a), b), ... and f), the image in the first column is the reference image, the image in the second column is the patchwork simulation using a model based on a 4×4 pixels grid and the third column contains the simulation using Strebelle's algorithm with a maximum of 60 points for conditioning and six multigrids. The dimensions of the black square (in unit of pixels) are a) 1x1, b) 2x2, c) 3x3, d) 4x4, e) 5x5 and f) 6x6.

All the simulations involving Strebelle's algorithm use a porosity equal to the training image porosity. We can see that Strebelle's approach does not reproduce perfectly any of the checkerboards. Even checkerboards with a mesh size of one pixel (third column in Figure 4.6-a) are not reproduced perfectly. Hence our approach produces much better results than Strebelle's approach for this particular type of training images.

4.3.2 Simulations using training images composed of discs

4.3.2.1 Discs of constant diameters

We tested the two methods on images composed of discs having a fixed diameter. A disc is a simple object in the sense that it can be fully described by only three parameters, i.e. the radius and the two coordinates of the center. It follows from this simplicity that the visual comparison between reference and simulated images is easy, especially if the discs do not intersect each other. The discs locations in the image are chosen according to a uniform probability distribution, but the discs are constrained to be disjoint. The algorithm used to construct these images is the following. Firstly, we draw a random disc location from a uniform distribution. If the resulting disc does not intersect any of the previously simulated discs, then we keep it. Otherwise we draw a new disc location. This process is stopped when the total number of discs in the image reaches a predetermined threshold.

Figure 4.7 displays simulations based on training images composed of discs with diameters ranging from 4 to 16 pixels. The size of the training images is 256x256 pixels, but the size of the images displayed in Figure 4.7 is 128x128 pixels. The patchwork simulations are performed with the 4 x4 model.

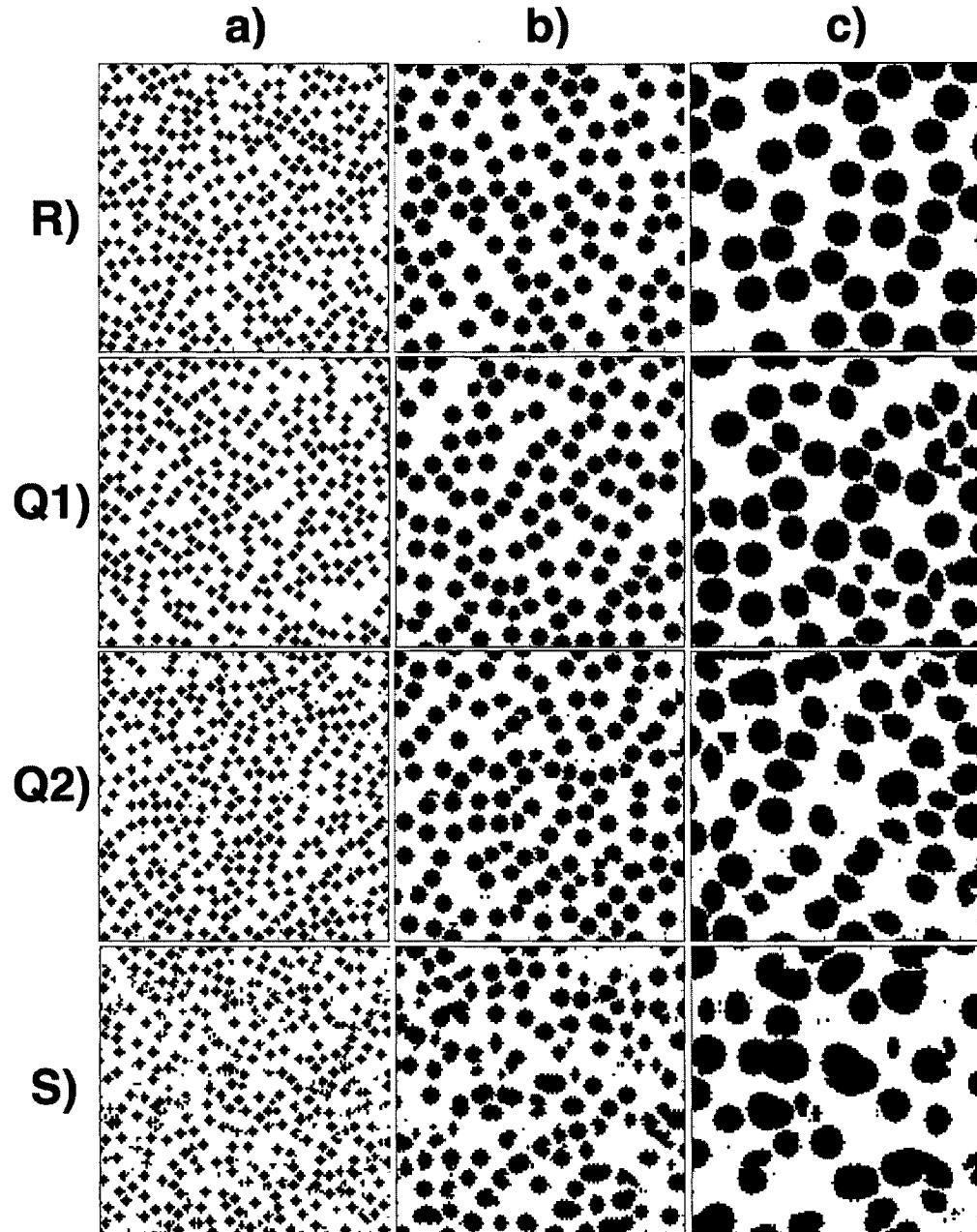


Figure 4.7 Simulations based on training images composed of discs with a fixed diameter D : a) $D = 4$ pixels; b) $D = 8$ pixels; c) $D = 16$ pixels. The patchwork simulations are done with a 4×4 pixels model. The first line (R) contains the reference images. Three types of simulations are presented: Q1) patchwork simulations without porosity filtering; Q2) patchwork simulations with a 10% porosity filtering; S) simulations obtained with Strebelle's algorithm using a maximum of 60 points for conditioning and four multigrids. The size of the images displayed is 128x128 pixels, i.e. a quarter of the training image size.

In the second line of Figure 4.7 (Q1), we present simulations obtained with the patchwork approach without porosity filtering. Apart from a slight porosity loss, the simulations with $D = 4$ pixels and $D = 8$ pixels are excellent. There are few errors in the simulated discs and the similarity between the reference and the simulation is visually striking. Similar results were obtained for $1 \leq D \leq 8$. However, for $D = 16$ pixels, we observe a significant degradation of the simulations quality. Nevertheless, the general appearance of the simulation remains acceptable.

In the third line of Figure 4.7 (Q2), we present the patchwork simulations obtained with a 10% porosity filtering. The results look similar, apart from the occurrence of a few isolated pixels which do not exist in the reference image. This is explained by the smaller choice of substitution states available when correcting unobserved states. For $D = 4$ pixels, we also observe that the simulation porosity is closer to the reference image porosity. This is a positive effect of the porosity filtering.

In the fourth line of Figure 4.7 (S), we present the simulations obtained with Strebelle's algorithm on the same training images. We see that our patchwork approach produces better results than Strebelle's approach, especially for images containing discs with a small diameter. Indeed, we observe that the proportion of deformed discs in these simulations (Figures 4.7-a-S and 4.7-b-S) is clearly higher than for patchwork simulations.

Figure 4.8 displays four simulations obtained with our patchwork approach using a 3×3 pixels model without porosity filtering. The training images contain discs with diameters ranging from 3 to 16 pixels. The simulations obtained with $D = 3$ and $D = 6$ pixels, which are displayed in Figures 4.8-a and 4.8-b respectively, are good. The absence of porosity filtering results in a slight porosity loss. Note that for $D = 3$ pixels the small number of pixels implies that the discs are actually squares. For $D > 6$ pixels (Figures 4.8-c and 4.8-d), we observe that the discs are definitely not reproduced accurately because several large deformed discs are present. These results indicate that the patchwork model produces good quality simulations if the disc diameter is smaller than twice the block size. Above this limit, the simulation quality degrades significantly.

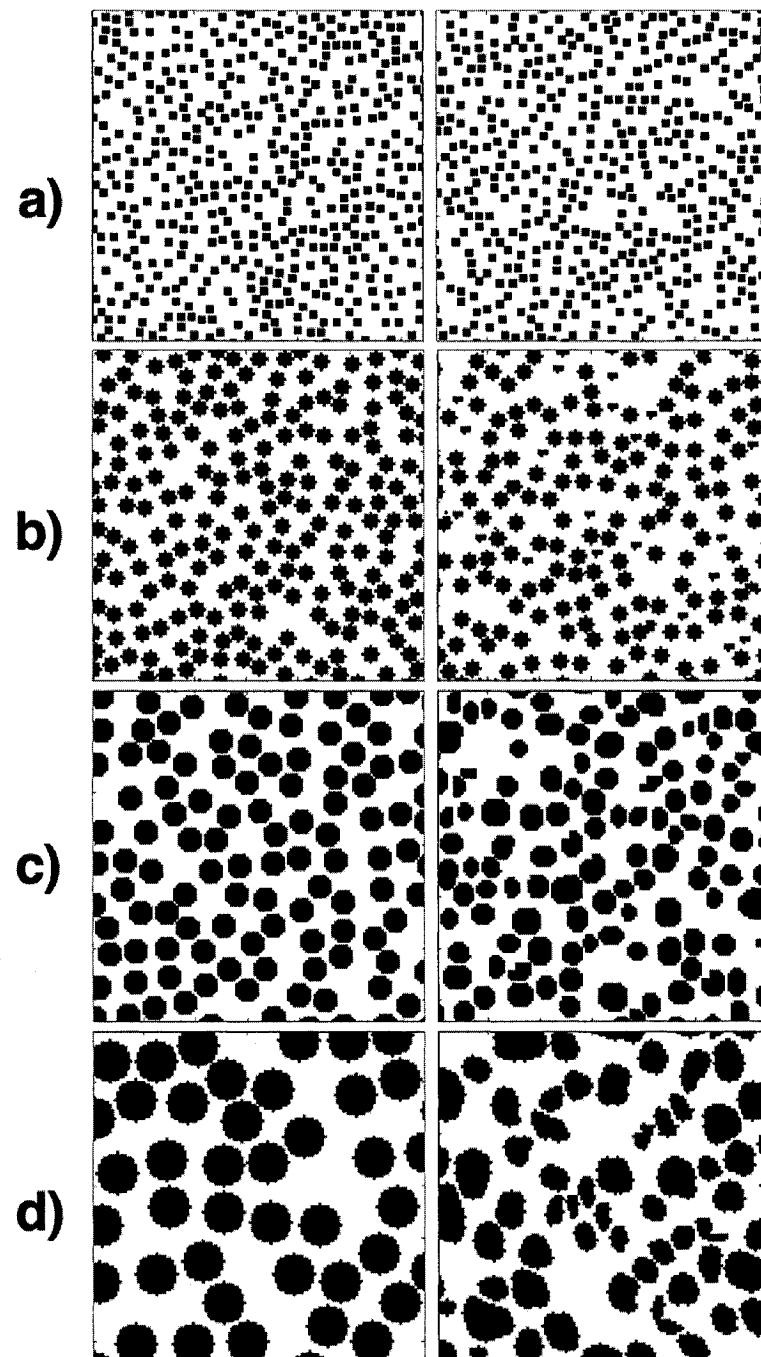


Figure 4.8 Patchwork simulations based on training images composed of discs with a fixed diameter D : a) $D = 3$ pixels; b) $D = 6$ pixels; c) $D = 9$ pixels and d) $D = 16$ pixels. The left column contains the reference images and the right column the corresponding simulations using a 3×3 pixels model without porosity filtering. The size of the images displayed is 128x128 pixels, whereas the training images have a size of 256 x 256 pixels.

4.3.2.2 Discs of variable diameters

We tested the two models on images containing discs with different diameters within the same image, as illustrated in Figure 4.9. These images were constructed as follows. First, we choose randomly a disc location from a distribution which is uniform in the image square. Next, we generate a disc diameter D that takes D_{\max} equiprobable values, where D_{\max} is the maximum diameter and D is an integer such that $1 \leq D \leq D_{\max}$. If the disc generated does not intersect any of the previously simulated discs, then we keep it. Otherwise, a new disc location is chosen and another disc diameter is generated. These steps are repeated until the total number of discs reaches a predetermined threshold. The size of the training images used is 256x256 pixels and the images displayed in Figure 4.9 have a size of 128 x128 pixels.

The second column of Figure 4.9 presents the simulations obtained with the patchwork approach based on a 4x4 pixels model with a 10% porosity filtering. For $D_{\max} \leq 8$ pixels, the simulations are good. However, the simulation quality degrades for $D_{\max} > 8$ pixels. These results show that the simulations quality is primarily limited by the maximum disc diameter. We observe that the proportion of large discs is reduced with respect to the reference images, and a porosity loss occurs.

The results obtained with Strebelle's algorithm on the same training images are presented in the third column of Figure 4.9.

Comparing the second and third columns of Figures 4.9, it is clear that the patchwork approach gives better results. The simulations using Strebelle's algorithm exhibit a larger number of deformed discs and the degree of deformation of these discs is also higher. Moreover, we observe an excess of oversized deformed discs. In Figure 4.9-c, our patchwork approach preserves better the discs shape but Strebelle's algorithm reproduces better the number of large discs and the global porosity of the image.

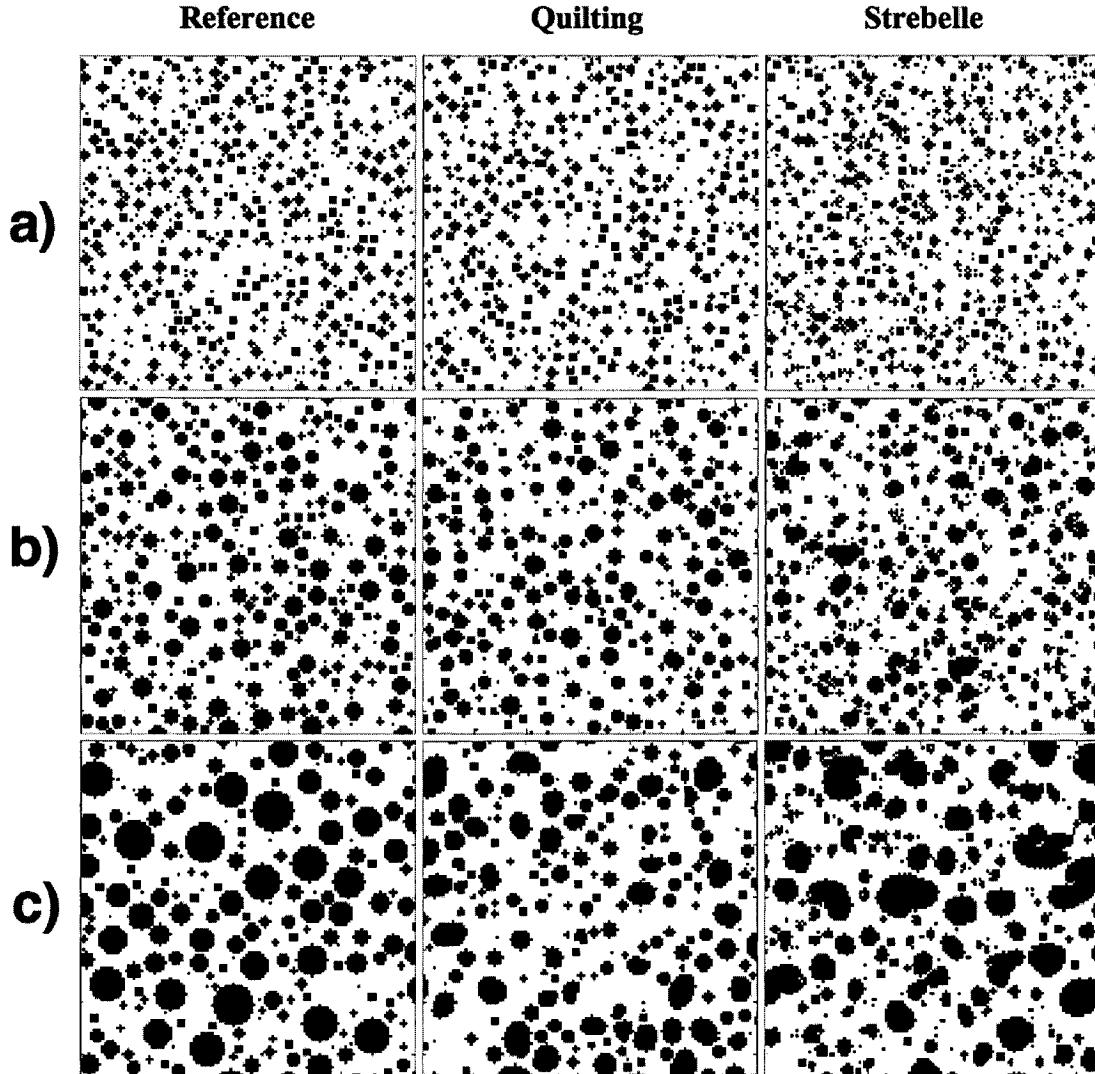


Figure 4.9 Simulations based on training images composed of discs with a varying diameter size D : a) $1 \leq D \leq 4$ pixels; b) $1 \leq D \leq 8$ pixels; c) $1 \leq D \leq 16$ pixels. The first column contains the reference images. The second column contains the patchwork simulations using the 4×4 pixels model with a 10% porosity filtering. The third column contains the simulations using Strebelle's algorithm using a maximum of 60 points for conditioning and six multigrids. The size of the images displayed is 128x128 pixels, whereas the training images have a size of 256x256 pixels.

4.3.3 Simulations using training images composed of sticks

We first tested the two methods on training images composed of sticks of constant length and varying orientation. These images are slightly more complex than images

composed of discs because each stick is characterized by four parameters (length, orientation and two position coordinates). The simple geometry of sticks facilitates the visual comparison between the simulations and the training images, especially if the sticks do not intersect each other.

The sticks locations in the training image are chosen according to a uniform probability distribution, but the sticks are constrained to be disjoint. The stick orientation is chosen randomly with equal probability among the four inclination angles 0, 45, 90 and 135 degrees. Figure 4.10 displays simulations based on training images of 256x256 pixels containing sticks of length ranging from 4 to 16 pixels. The thickness of each stick is one pixel. The second line of Figure 4.10 contains the patchwork simulations based on a 4x4 model with a 10% porosity filtering. The third line of Figure 4.10 contains the patchwork simulations based on a 4 x4 model without porosity filtering. The fourth line of Figure 4.10 displays the simulations obtained with Strebelle's algorithm.

The patchwork simulations obtained without porosity filtering (Figure 4.10-Q2) suffer from a porosity loss. The simulations quality decreases when the stick length is larger than the block size (4 pixels). The stick length distribution is good for $L = 4$, i.e. most sticks have the correct length $L = 4$ pixels, but degrades for $L = 8$ and 16 pixels. Indeed, simulated sticks are often longer or shorter than the sticks in the reference image. The patchwork simulations obtained with a 10% porosity filtering (Figure 4.10-Q1) have a porosity which is comparable to the reference image porosity. In Figures 4.10-c-Q1 and 4.10-c-Q2, the abundance of undersized sticks orientated in the north-east direction is striking.

The simulations obtained with Strebelle's algorithm (Figure 4.10-S) have an appropriate porosity because the algorithm allows a partial control on porosity. However, the simulations suffer from an excess of undersized sticks in all three cases, and an almost complete absence of long sticks. Moreover, we observe that many of the simulated sticks fail to be disjoint.

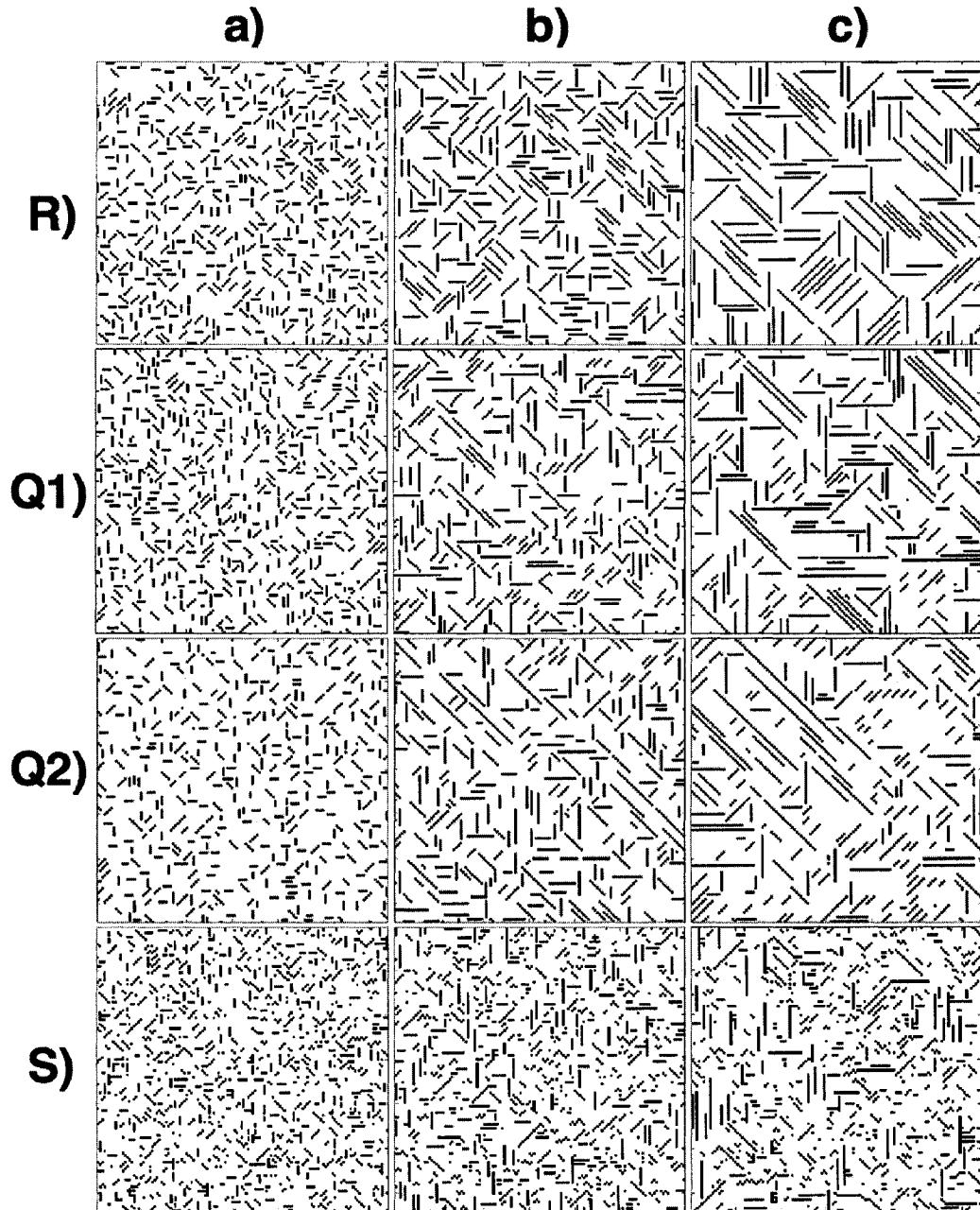


Figure 4.10 Simulations based on training images composed of sticks of a fixed length L : a) $L = 4$ pixels; b) $L = 8$ pixels and c) $L = 16$ pixels. The patchwork simulations are based on the 4×4 pixels model. The first line contains the training images (R). The second line (Q1) contains the patchwork simulations with a 10% porosity filtering. The third line (Q2) contains the patchwork simulations without porosity filtering. The fourth line (S) contains the simulations produced with Strebelle's algorithm using a maximum of 60 points for conditioning and four multigrids. The size of the images displayed is 128×128 , whereas the size of the training images is 256×256 pixels.

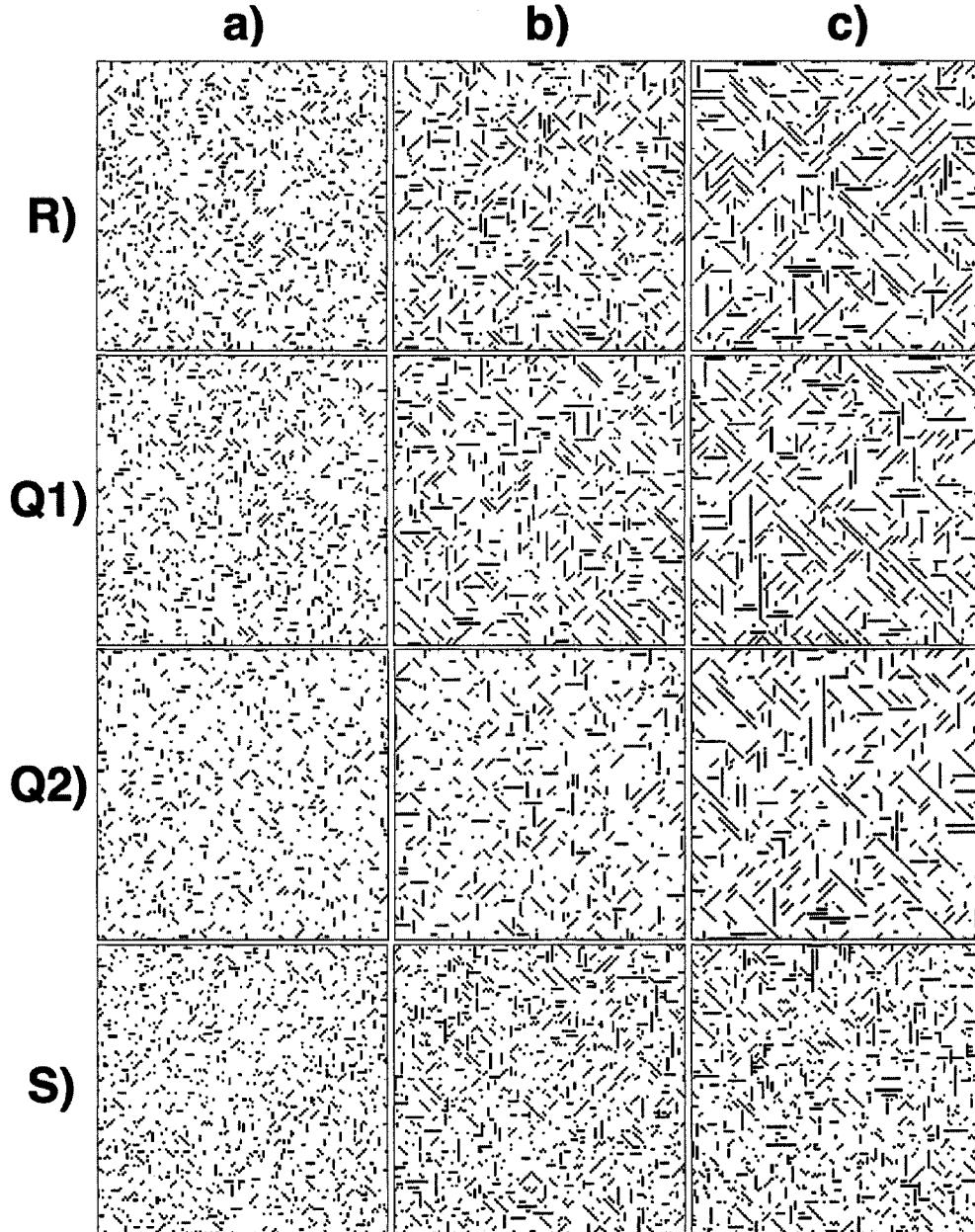


Figure 4.11 Simulations based on training images composed of sticks with varying stick length L : a) $1 \leq L \leq 4$ pixels; b) $1 \leq L \leq 8$ pixels; c) $1 \leq L \leq 16$ pixels. The patchwork simulations are based on the 4×4 pixels model. The first line contains the training images (R). The second line (Q1) contains the patchwork simulations with a 10% porosity filtering. The third line (Q2) contains the patchwork simulations without porosity filtering. The fourth line (S) contains the simulations produced with Strebelle's algorithm using a maximum of 60 points for conditioning and four multigrids. The size of the images displayed is 128×128 , whereas the size of the training images is 256×256 pixels.

We tested the models on images containing sticks with varying lengths and orientations. The stick length L is chosen randomly and takes one of L_{\max} equiprobable values, where L_{\max} is the maximum stick length and L is an integer such that $1 \leq L \leq L_{\max}$. For patchwork simulations, we observe in Figure 4.11 that the simulations quality degrades if L_{\max} is larger than the block size (4 pixels). We also observe a porosity loss for the patchwork simulations obtained without porosity filtering. The simulations obtained with Strebelle's algorithm again suffer from an excess of undersized sticks and intersecting sticks, as well as a deficit of long sticks.

4.3.4 Simulations using a training image containing channels

In this section, we compare our approach and Strebelle's algorithm on an image which contains objects having a size comparable to the training image size. The objects chosen mimic the shape of a channel, as shown in Figure 4.12-R. Figure 4.12 displays simulations obtained with the two approaches. The patchwork simulation uses a 4 x 4 pixels model. Unlike all the previous patchwork simulations, we scanned the training image in a single direction because the image is anisotropic. The simulations obtained with Strebelle's algorithm use 60 conditioning points and five multigrids.

For both methods, the simulated channels look similar to the training image channels. Best results are obtained in Figure 4.12-Q2 where the channels are continuous across the whole image, as in the reference image. In the Figures 4.12-Q1 and 4.12-S, channels appear (or disappear) abruptly within the simulated window.

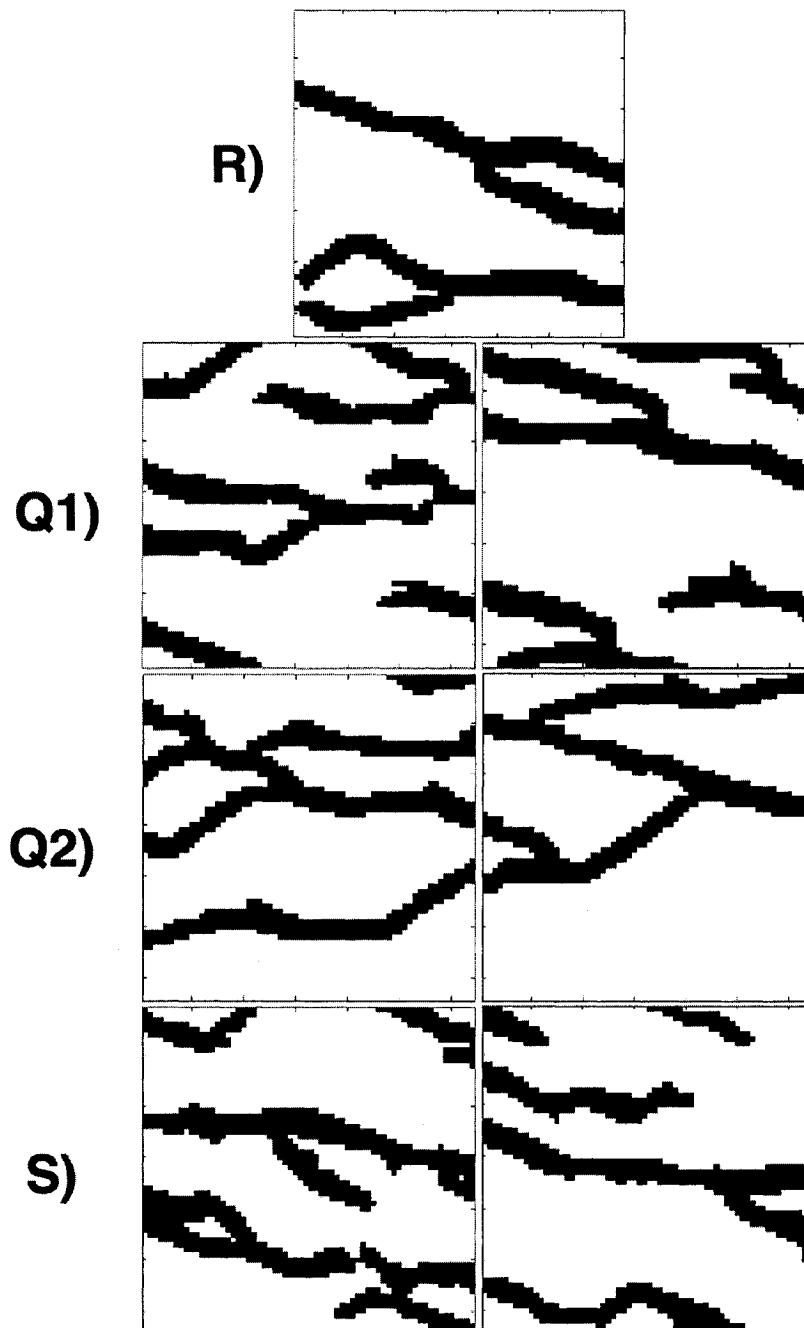


Figure 4.12 Simulations based on the reference image R) which is composed of two channels: Q1) patchwork simulations with a 10% porosity filtering; Q2) patchwork simulations without porosity filtering; S) simulations based on Strebelle's algorithm. The patchwork simulations are based on a 4×4 pixels model. Strebelle's algorithm was used with a maximum of 60 conditioning points and five multigrids. The size of the images displayed is 64×64 pixels.

4.3.5 Simulations based on an electron microscope polymer blend image

A polymer blend is a material obtained by mixing two or more single phase polymers [22, 23]. The vast majority of polymer blends are not miscible in the thermodynamic sense and hence form multiphase systems resembling oil-in-water type emulsions. For immiscible polymers, there are two main morphologies for the blend [16]: dispersed or co-continuous. In the co-continuous morphology, each polymer is fully interconnected through a complex microstructure. The understanding of this morphology is important for the development of new materials based on polymer blends and the potential applications include: carriers for conductive polymers, membranes and scaffolds for tissue engineering and drug delivery.

The plastic industry shows an increasing interest for polymer blends because of their economical importance. They are low cost materials with superior properties compared with monophase polymers.

In this section, we focus on the stochastic simulation of two-dimensional cuts of a co-continuous polymer blend. More precisely, we test our model on an electron microscope image of a co-continuous polymer blend. The image (Figure 4.13) was produced by microtomy and scanning electron microscopy (SEM).

Figure 4.13 displays an electron microscope image of a blend composed of polystyrene and polycaprolactone (PS/PCL) in volumic proportions of 50%-50%. The specimens were microtomed to create a perfect plane face using a Reichert Jung 2050 Supercut Microtome with a glass knife equipped with the Liquid Nitrogen Freezing Device LN20. In order to improve the contrast in the image, the PCL was extracted with acetic acid. After selective solvent extraction and coating with a gold-palladium alloy, the microtomed specimens were observed under a Jeol JSM 840 scanning electron microscope at a voltage of 10 kV.

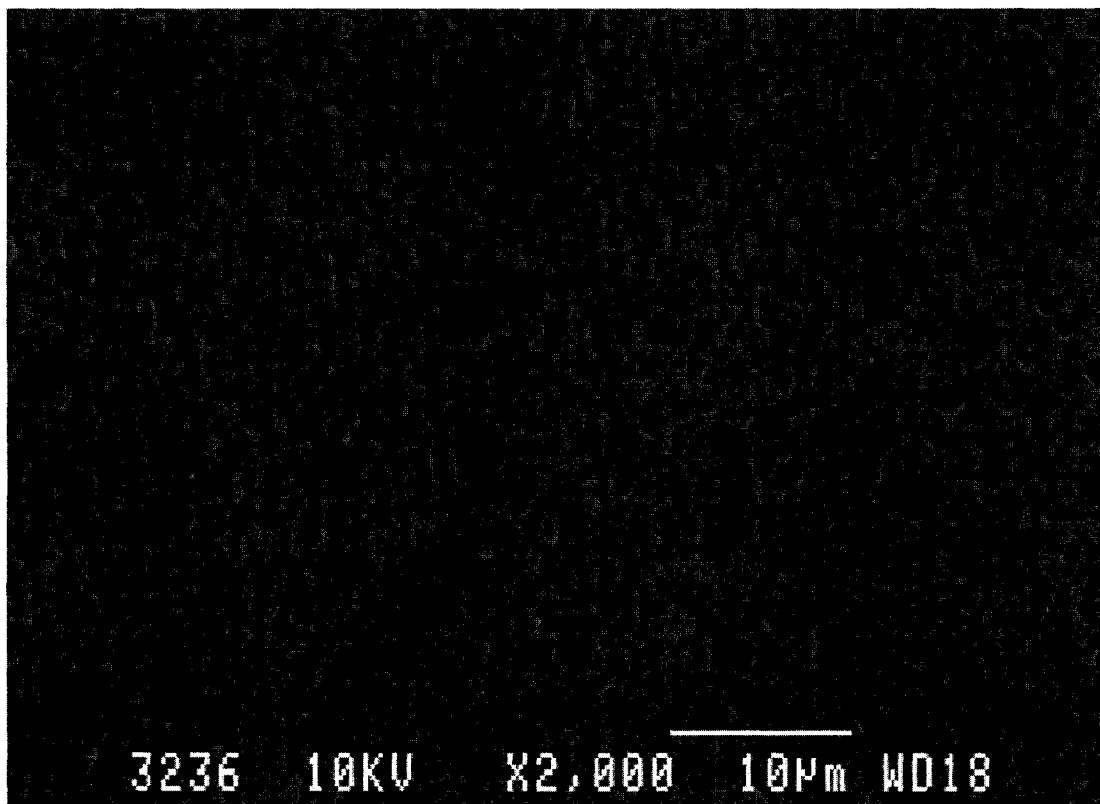


Figure 4.13 Electron microscope image of a co-continuous polymer blend. The image size is 694 x 503 pixels.

The original image displayed in Figure 4.13 was filtered to produce a training image for our model. A threshold was applied to the image after noise removal and contrast enhancement. The size of the training image displayed in Figure 4.14-R is 314 x 434 pixels. The patchwork simulations using a 4 x 4 and a 7 x 7 model with a 10% porosity filtering are displayed in the Figures 4.14-Q1 and 4.14-Q2 respectively. The similitude between the training image and the simulations is visually striking. However, the white channels appear to be broader in the patchwork simulations than in the reference image. In addition, we observe an excess of large black clusters, especially with the 7 x 7 model. The simulation obtained with Strebelle's algorithm using 60 conditional points and 4 multigrids is displayed in Figure 4.14-S. This simulation looks more similar to the training image than the patchwork simulation. In particular, we observe that the number of small black grains found in the reference image is better reproduced.

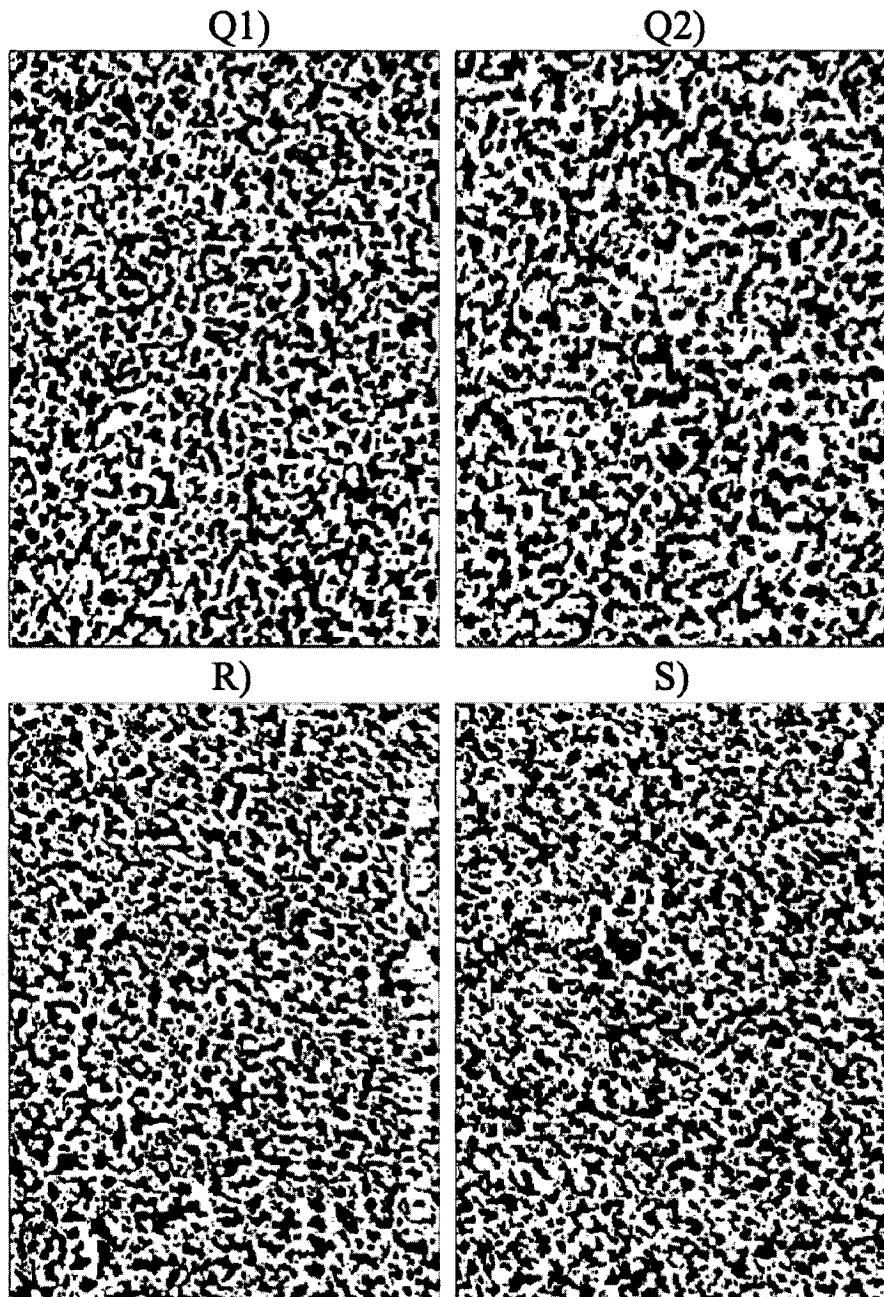


Figure 4.14 Simulations using a polymer blend microscopic image as a training image: R) reference image; Q1) patchwork simulation using a 4×4 model and a 10% porosity filtering; Q2) patchwork simulation using a 7×7 model and a 10% porosity filtering; S) simulation obtained with Strebelle's algorithm using 60 conditioning points and four multigrids. The size of these images is 314 x 434 pixels.

To compare quantitatively the simulations with the polymer blend training image, we measured the porosity and the cord length distribution (CLD) of the images. Cords are black segments of thickness one pixel which are found at the intersection of the image with a straight line. The cord lengths are measured in the horizontal and vertical directions only. The porosities of the patchwork simulations are 42.8% (4 x 4 model) and 41.7% (7 x 7 model), whereas the porosity of the simulation with Strebelle's method is 42.9%. These porosities are close to the training image porosity (42.3%). Figure 4.15 displays the CLDs for the four images in Figure 4.14. Comparing the CLDs displayed in Figure 4.15, we observe that the CLD associated with Strebelle's algorithm is closest to the training image CLD. The CLD associated with the patchwork simulation suffers from a deficit of small cord lengths and a slight excess of long cord lengths.

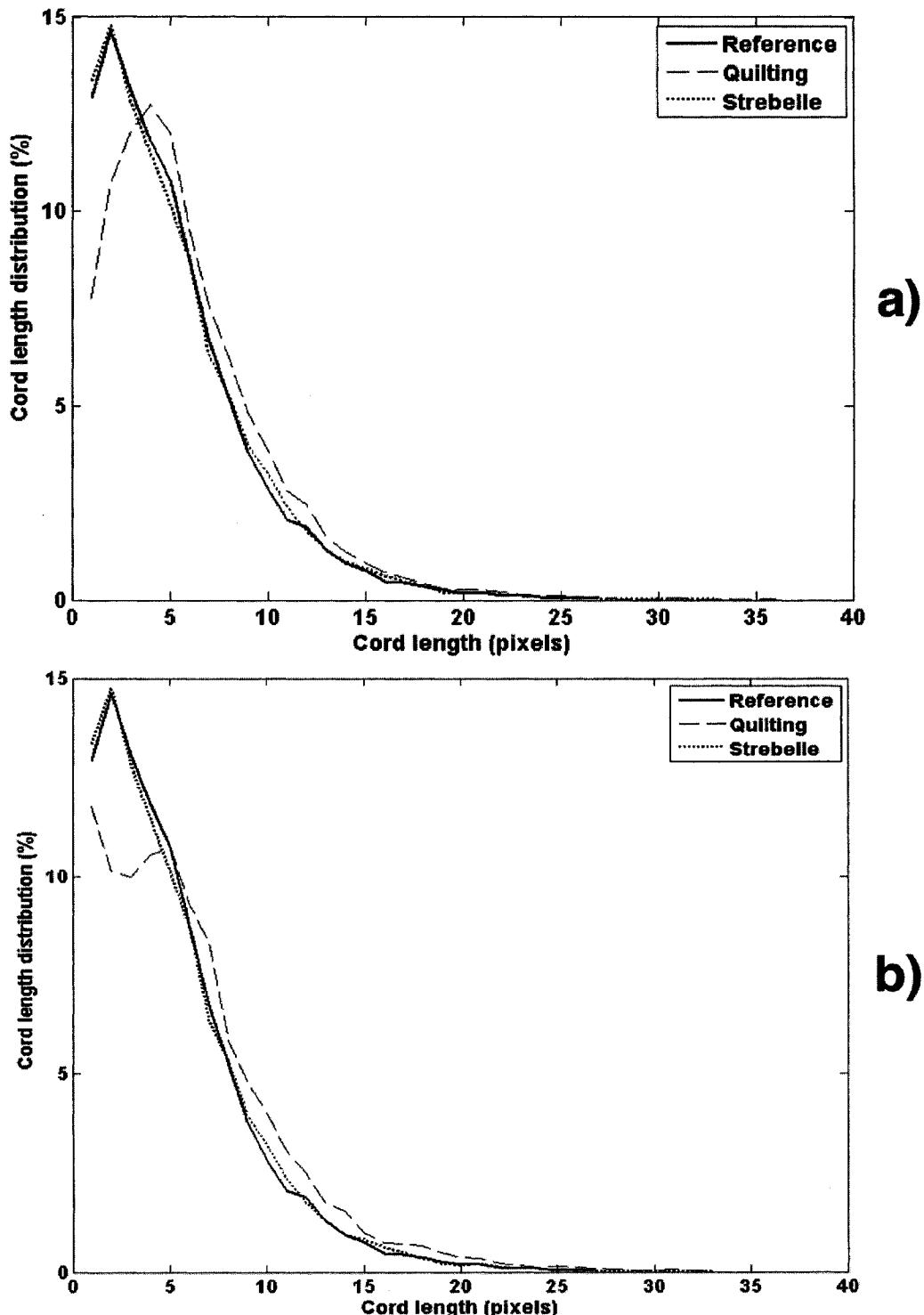


Figure 4.15 Cord length distributions of the four images in Figure 4.14: a) patchwork simulation using the 4 x 4 model; b) patchwork simulation using the 7 x 7 model.

4.4 Discussion

The simulations presented in this paper on several types of training images show the strengths and limitations of our patchwork approach. We showed that the patchwork method performs very well on checkerboards or on images composed of small discs. For such images, patchwork simulations give better results than Strebelle's algorithm. Indeed, we have shown that the checkerboards with a mesh size smaller or equal to the block size are perfectly reproduced. Moreover, images containing discs of a fixed or varying diameter are reproduced with a good quality if the maximum diameter is smaller than twice the block size. Our patchwork approach is primarily limited by the relationship between the size of the objects we want to reproduce and the block size. The quality of the reconstructions decreases when the difference between these two parameters increases.

For images composed of long sticks, Strebelle's algorithm failed completely whereas the patchwork approach failed to generate long sticks in the 45 degrees direction. This directional bias is linked to the inverse "L" shape of the conditioning pattern (Figure 4.2-c). Let us suppose that the upper part of a long stick in the 45 degrees direction is simulated in a block labeled by A. The lower part of this stick should intersect a block B located on the lower left side of block A. However, block B will be simulated using only the knowledge of blocks which are located on the left side of block A, thus without the knowledge that the beginning of a long stick exists in block A. Consequently, the sticks simulated in block B will only match by chance the sticks simulated in block A, explaining why so many small sticks appear in that direction. Images could be rotated to alleviate the directional bias induced by the *L*-shape model.

The simulations of channels were comparable for both methods. This shows that in some cases the patchwork approach can simulate correctly structures which are much longer than the block size.

For the microscopic image of a polymer blend, Strebelle's algorithm produced a slightly better simulation than the patchwork approach. In particular, the cord length distribution

of the patchwork simulations exhibits a moderate excess of long cord lengths with respect to the reference image. In this image, the presence of clusters which are much larger than the block size explains the poorer performance of the patchwork simulations, which are not designed to accurately reproduce large-scale features.

Both approaches require a good estimation of the transition probabilities to produce good results. To reach this goal, the size of the training images has to be large compared with the block size. With small training images, the transition probabilities are poorly estimated and the simulation results have a lower quality.

The block based approach presented in this article uses black and white images only. This can be easily improved by changing the binary coding of the states to a base p coding where p is the number of possible states at each pixel. In the patchwork approach, conditioning to hard data is possible on a local basis only by choosing among the arrival states that fits the hard data within the currently simulated block.

Generalization of the patchwork approach to three-dimensional (3D) images is straightforward when 3D training blocks are available. However, the real interest would rather be to extrapolate to 3D in a reasonable way from a series of 2D training images. For the moment, no satisfying model or technique exists to accomplish this challenging task.

4.5 Conclusion

We presented in this paper a new sequential method of simulation based on the estimation of transition probabilities between adjacent blocks composing the training image. Our approach has the ability to reproduce correctly many morphological features because the blocks used in the patchwork process, which are drawn directly from the training image, are essentially correct.

We compared the performance of patchwork simulations with Strebelle's multipoint simulation algorithm on several types of images of increasing complexity. For images composed of clusters which are small with respect to the block size (e.g. squares, discs,

sticks), our patchwork approach produces better results than Strebelle's method. The most noticeable improvement is that the cluster geometry is usually reproduced accurately. In general, the simulation quality degrades when the maximum cluster size is larger than the block size. The patchwork approach was applied to the training image of a co-continuous polymer blend. The resulting simulations are good visually but the CLD associated with the patchwork simulation suffers from a deficit of small cord lengths and a slight excess of long cord lengths. In the polymer blend image, the presence of clusters which are much larger than the block size explains the poorer performance of patchwork simulations, which are not designed to accurately reproduce large-scale features.

Although the patchwork approach seems promising, particularly for two phase systems where one phase dominates, some questions need further research, such as: 1) how to control the probability distribution of the porosity at the block scale and 2) how to alleviate the interaction of directional features with the shape of the conditioning pattern. Once these questions resolved, the formidable task of generalizing from 2D images to 3D blocks will have to be examined carefully.

4.5.0.0.1 Acknowledgements A. Saucier, D. Marcotte and B. D. Favis acknowledge financial support from individual NSERC grants. The authors thank Dr. P. Sarazin for providing the polymer blend micrograph used in the analysis.

References

- [1] ADLER, P.M. (1992). *Porous Media*. Butterworth-Heinemann, Boston.
- [2] ADLER, P.M., JACQUIN, C. and QUIBLIER, C.J. (1990). Flow in simulated porous media. *International Journal of Multiphase Flow*, 16:691–712.
- [3] ARPAT, B. (2003). *Report No. 16: A pattern recognition approach to multiple-point simulation*. Stanford Center for Reservoir Forecasting, Stanford University.
- [4] BAKKE, S. and ØREN, P. (1997). 3-d pore-scale modeling of sandstones and flow simulations in pore networks. *SPE Journal*, 2:136.
- [5] BEAR, J. and VERRUIJT, A. (1987). *Modeling Groundwater Flow and Pollution*. Kluwer Academic Publishers, Dordrecht.
- [6] BRYANT, S., MELLOR, D. and CADE, C. (1993). Physically representative network models of transport in porous media. *AIChE Journal*, 39:387.
- [7] CAERS, J. (1998). Stochastic simulation using neural networks. In *Stanford Center for Reservoir Forecasting Annual Meeting 11*, pages 1–66. Stanford University, Vol. 2.
- [8] CAERS, J. and JOURNEL, A.G. (1998). Stochastic reservoir simulation using neural networks trained on outcrop data. *SPE paper no. 49026*.
- [9] CHATZIS, I. and DULLIEN, F. (1977). Modelling pore structure by 2-d and 3-d networks with applications to sandstones. *Journal of Canadian Petroleum Technology*, page 97.

- [10] DEUTSCH, C. and JOURNEL, A. (1998). *GSLIB: Geostatistical Software Library*. Oxford University Press, New York. Second Edition.
- [11] FATT., I. (1956). The network models of porous media i. capillary pressure characteristics. *AIME Petroleum Transactions*, 207:144.
- [12] GUARDIANO, F. and SRIVASTAVA, R.M. (1993). Multivariate geostatistics: Beyond bivariate moments. In A. Soares, editor, *Geostatistics-Troia, vol. 1*, pages 133–144, Dordrecht, Netherland. Kluwer Academic.
- [13] HALDORSEN, H. and DAMSLETH, E. (1990). Stochastic modeling. *Journal of Petroleum Technology*, pages 404–412.
- [14] KATZ, A. and THOMPSON, A. (1986). Quantitative prediction of permeability in porous rock. *Phys. Rev. B*, 34:8179.
- [15] LAKE, L. (1989). *Enhanced Oil Recovery*. Prentice Hall, Englewood Cliffs.
- [16] LI, J., MA, P.L. and FAVIS, B.D. (2002). The role of the blend interface type on morphology in cocontinuous polymer blends. *Macromolecules*, 35:2005–2016.
- [17] OMRE, H. (1991). Stochastic models for reservoir characterization. In J. Kleppe and S. M. Skjæveland, editors, *Recent Advances in Improved Oil Recovery Methods for North Sea Oil Sandstone Reservoirs*, Dordrecht, Netherland. Norwegian Petroleum Directorate.

- [18] REMY, N. (2004). *Stanford Geostatistical Modeling Software (S-GeMS), version 1.2, build.* Contributors: Joy Rajiv and Michael Sharps. Available on the Internet at http://prdownloads.sourceforge.net/sgems/sgems_installer-1.2.exe?download.
- [19] ROBERTS, J. and SHWARTZ, L. (1985). Grain consolidation and electrical conductivity in porous media. *Phys. Rev. B*, 31:5990.
- [20] ROY, S. and TRAFALDAR, S. (1997). Archies's law from a fractal model for porous rock. *Phys. Rev. B*, 55:8038.
- [21] SAHIMI, M. (1993). Flow phenomena in rocks: From continuum models to fractal, percolation, cellular automata and simulated annealing. *Rev. Mod. Phys.*, 65:1393.
- [22] SARAZIN, P. and FAVIS, B.D. (2003). Morphology control in co-continuous poly (L-lactide)/polystyrene blends: a route towards highly structured and interconnected porosity in poly (L-lactide) materials. *Biomacromolecules*, 4:1669–1679.
- [23] SARAZIN, P. and FAVIS, B.D. (2005). Influence of temperature-induced coalescence effects on co-continuous morphology in poly (ϵ - caprolactone)/polystyrene blends. *Polymer*, 46:5966–5978.
- [24] SAUCIER, A. and MULLER, J. (1993). Remarks on some properties of geometrical multifractals. *Physica A*, 199:350–362.
- [25] SAUCIER, A. and MULLER, J. (1999). Textural analysis of disordered materials with multifractals. *Physica A*, 268:221–238.

- [26] SAUCIER, A., RICHER, J. and MULLER, J. (2002). Assessing the scope of the multifractal approach to textural characterization with statistical reconstruction of images. *Physica A*, 311/1-2:231–259.
- [27] STOYAN, D., KENDALL, W. and MECKE, J. (1987). *Stochastic Geometry and its Applications*. Wiley, Berlin, Chichester. Second Edition.
- [28] STREBELLE, S. (2002). Conditional simulation of complex geological structures using multiple point statistics. *Mathematical Geology*, 34(1):1–22.
- [29] STREBELLE, S. (2003). New multiple-point statistics simulation implementation to reduce memory and cpu-time demand. Presented at the 2003 conference of the International Association for Mathematical Geology, in Portsmouth, UK, September 7–12. Available on the Internet at <http://www.iamg.org/meetings/Proceedings2003/papers/Strebelle2.pdf>.
- [30] TORQUATO, S. (2000). Modeling of physical properties of composite materials. *International Journal of Solids and Structures*, 37:411–422.

CHAPITRE 5 APPROCHE MULTI-ÉCHELLE

Dans ce chapitre, on décrit la méthode multi-échelle que nous avons développée. On y présente également quelques simulations obtenues avec cette approche. Le modèle présenté ci-dessous a été initié par l'approche multi-échelle développée par Saucier et Muller [28, 30]. En effet, dans leur approche de caractérisation par moments multi-échelle, Saucier et Muller arrivent à reconstruire des images contenant des petits objets à faible densité. Cependant, la propriété de connectivité à laquelle on accorde une grande importance dans ce projet n'était pas reproduite correctement. L'approche multi-échelle que nous avons développée vise justement à reproduire correctement cette propriété. Parmi les approches multi-échelles développées qui produisent de bons résultats on retrouve celles basées sur les ondelettes [3, 35, 36]. Par exemple, l'approche de Vemuri et Radisavljevic [36] permet de générer une grande variété de formes tridimensionnelles à partir des superquadriques déformables [35].

5.1 Description de la méthode

Notre approche de modélisation multi-échelle vise à produire des simulations ayant les même propriétés statistiques que l'image de référence et ce à toutes les échelles de l'image. Plus précisément, notre modèle reproduit les probabilités de transition entre les porosités évaluées à des échelles diadiques successives de l'image. Notre approche peut se résumer en deux étapes principales : l'extraction des probabilités de transition entre porosités évaluées à des échelles successives de l'image de référence, puis la simulation stochastique se basant sur ces probabilités. Avant de décrire les probabilités conditionnelles utilisées par le modèle ainsi que son fonctionnement, des définitions préliminaires s'imposent.

5.1.1 Définitions préliminaires

Notre modèle multi-échelle utilise les porosités de l'image de référence évaluées à des échelles diadiques. On appelle porosité d'une image noir et blanc la proportion de pixels noirs dans celle-ci. On définit l'échelle diadique L_n d'une image carrée de taille L_0

comme suit : $L_n = \frac{L_0}{2^n}$, $n \in \{0, 1, \dots, N\}$

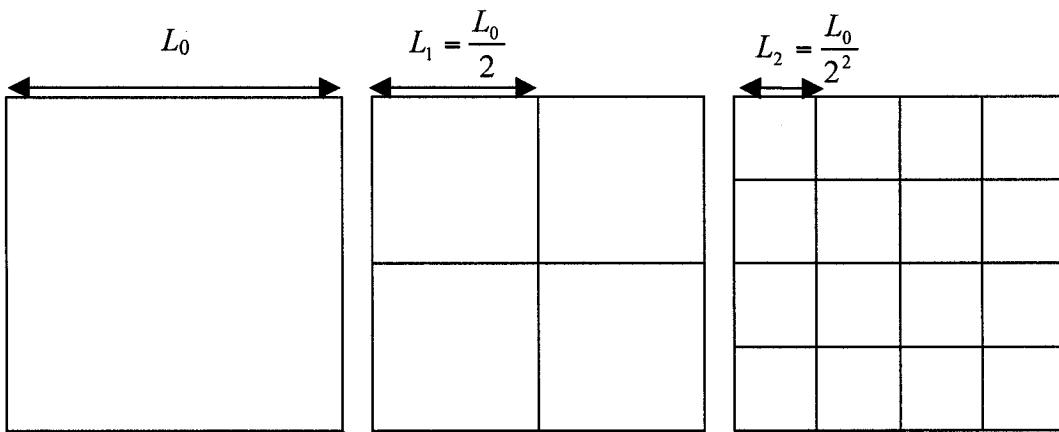


Figure 5.1 Schématisation des échelles diadiques. Représentation pour une image carrée de taille L_0 des échelles diadiques L_1 et L_2 .

Dans la Figure 5.1, on représente les échelles diadiques L_1 et L_2 d'une image de taille L_0 . On dira que la porosité est évaluée à l'échelle L_1 (resp. L_2) lorsqu'elle est calculée pour chacune des 4 (resp. 16) régions de l'échelle L_1 (resp. L_2). L'échelle $L_n=1$ avec $n=\log_2(L_0)$ représente l'échelle la plus précise de l'image soit celle du pixel. Le modèle multi-échelle utilise des images noir et blanc seulement. Les porosités évaluées aux différentes échelles de l'image sont arrondies par rapport à la porosité moyenne, i.e. elles valent 0 lorsque inférieures à la porosité moyenne et 1 autrement. Les porosités sont codées sur un seul bit (0 ou 1). On appelle un état S_i de l'image un nombre binaire $S_i = 100101\dots101$, où chaque bit représente la porosité arrondie évaluée dans une région de l'image. La Figure 5.2 présente une configuration possible des porosités évaluées à deux échelles diadiques qui se suivent L_n et L_{n+1} . Les grands (resp. petits) disques

représentent les porosités évaluées aux échelles L_n (resp. L_{n+1}). L'ensemble des disques blancs représentent les bits d'un état de départ S_i tandis que les disques noirs représentent les bits d'un état d'arrivée S_j . L'état de départ S_i est codé sur 25 bits dont 9 représentent des porosités évaluées à l'échelle L_n et 16 des porosités évaluées à l'échelle L_{n+1} . L'état d'arrivée est codé sur 4 bits représentant des porosités évaluées à l'échelle L_{n+1} .

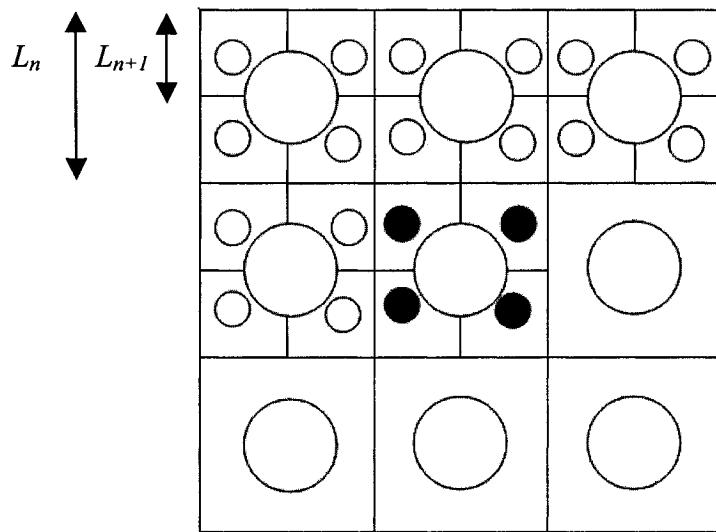


Figure 5.2 Exemple de configuration de porosités évaluées à deux échelles L_n et L_{n+1} . Les grands (resp. petits) cercles représentent les porosités évaluées à l'échelle L_n (resp. L_{n+1}). Les cercles blancs représentent l'état de départ S_i tandis que les cercles noirs représentent l'état d'arrivée S_j .

Les probabilités conditionnelles $P_{i,j}$ d'observer un état S_j sachant qu'un état S_i a été observé sont évaluées à toutes les paires d'échelles diadiques de l'image. Ces probabilités de transition $P_{i,j}$ vérifient :

$$\begin{aligned} P_{i,j} &\geq 0, \quad \forall (i, j), \\ \sum_{j=1}^{2^{25}} P_{i,j} &= 1, \quad \forall i. \end{aligned}$$

La Figure 5.2 présente une des quatre configurations utilisées pour évaluer les probabilités de transition $P_{i,j}$ entre états de l'image.

5.1.2 Description des matrices de probabilités de transition

Le modèle multi-échelle utilise quatre matrices de probabilités de transition pour estimer les porosités de l'image à une échelle L_{n+1} connaissant les porosités à l'échelle précédente L_n . La Figure 5.3 illustre les quatre configurations correspondant aux états de départ et d'arrivée de chacune de ces matrices.

La matrice principale de notre modèle est schématisée par le diagramme #4 dans la Figure 5.3. Les états de départ de cette matrice sont codés sur 25 bits (disques en blanc) vers des états d'arrivée codés sur 4 bits (disques en noir). Cette matrice ne permet pas de simuler les porosités de la première ligne ou de la première colonne à l'échelle L_{n+1} . En effet, ces porosités sont supposées connues pour la matrice principale. Pour générer ces porosités, d'autres matrices ont été définies afin de compléter la matrice principale de ce modèle.

Ainsi, pour amorcer la simulation, on utilise le diagramme #1 de la Figure 5.3. La matrice correspondant à ce diagramme permet de connaître les porosités à l'échelle L_{n+1} connaissant les porosités à l'échelle L_n . Les états de départ sont codés sur 9 bits (grands disques blancs) tandis que les états d'arrivée sont codés sur 20 bits (petits disques noirs). Cette matrice n'est utilisée qu'une seule fois dans une simulation multi-échelle au début de la simulation.

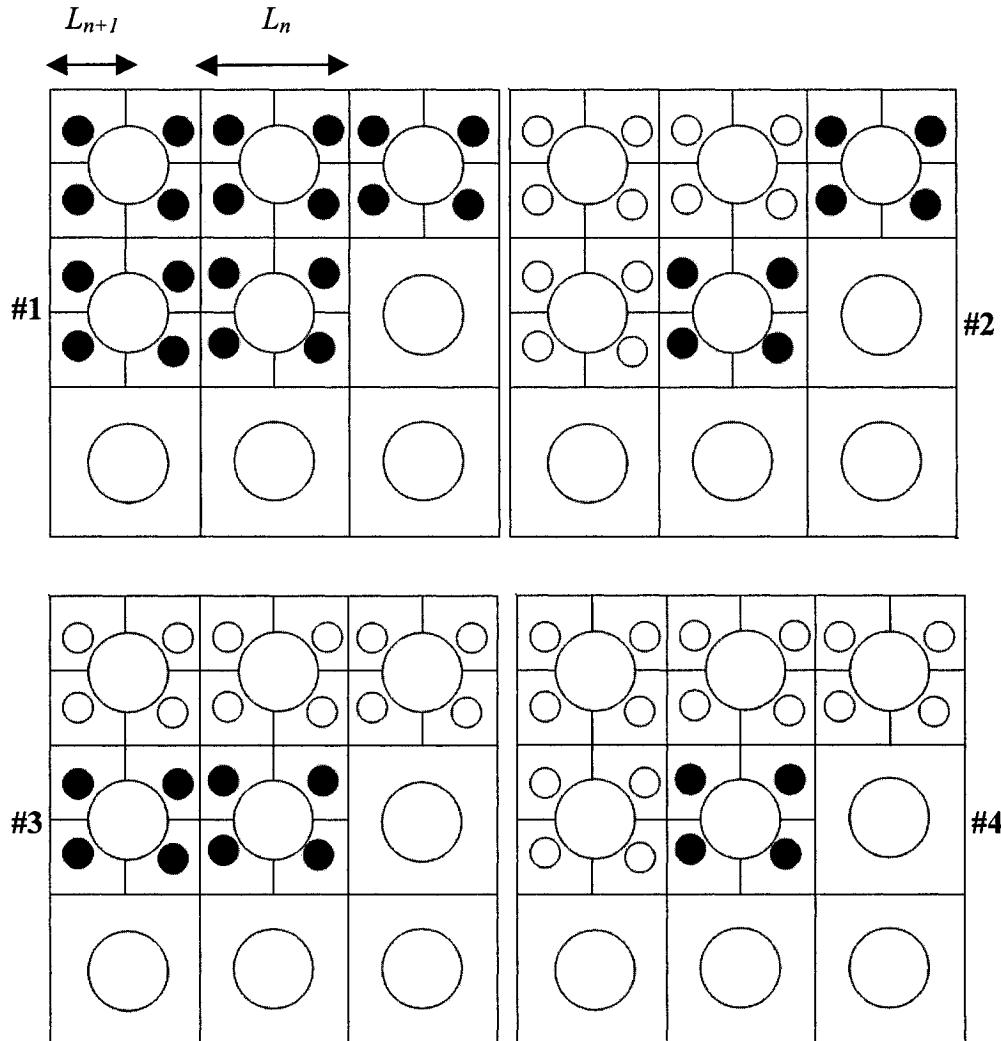


Figure 5.3 Présentation des diagrammes utilisés par le modèle multi-échelle. Chaque diagramme représente les états de départ et d'arrivée d'une des quatre matrices de probabilités de transition. Les bits des états de départ sont schématisés par des disques blancs tandis que les disques noirs schématisent les bits des états d'arrivée.

Pour simuler les porosités de l'échelle L_{n+1} situées à la première ligne de l'image ainsi que dans les fins de lignes, on utilise la matrice schématisée par le diagramme #2. Cette matrice mesure les probabilités de transition entre des états de départ codés sur 21 bits et des états d'arrivée codés sur 8 bits. Enfin, pour simuler les porosités de l'échelle L_{n+1} situées à la première colonne, on utilise la matrice schématisée par le diagramme #3.

Les 16 porosités évaluées à l'échelle L_2 de l'image de référence sont utilisées comme état de départ de notre simulation. Dans le cas où plusieurs images de référence sont utilisées, on sélectionne aléatoirement les porosités évaluées à l'échelle L_2 d'une des images de référence utilisées.

Étape no 1 : Utilisation du diagramme#1 de la Figure 5.3.

Ce diagramme est utilisé afin de simuler les porosités à l'échelle L_3 connaissant les porosités à l'échelle L_2 au début de la simulation. La Figure 5.5 illustre l'utilisation de la matrice de transition #1.

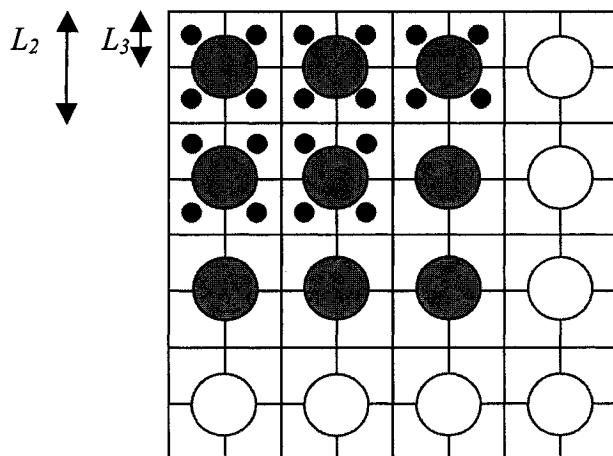


Figure 5.5 Étape1 : utilisation du diagramme #1

Ce diagramme permet d'assurer la transition de 9 porosités connues à l'échelle L_2 (grands disques gris), vers 20 porosités à l'échelle L_3 (petits disques noirs).

Étape no 2 : Utilisation du diagramme#2 de la Figure 5.3.

Ce diagramme est utilisé pour simuler les porosités à l'échelle L_3 de la première ligne ($i=1$) ainsi que celles situées à la fin de chaque ligne ($i>1, j=end$).

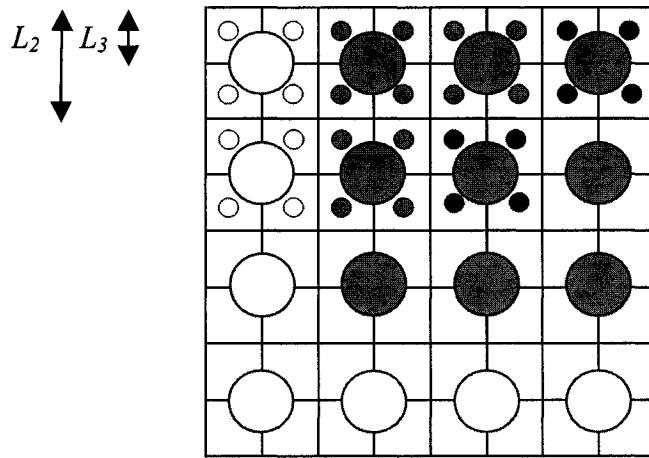


Figure 5.6 Étape 2 : utilisation du diagramme #2.

Comme on peut le voir dans la Figure 5.6, cette matrice permet de trouver une transition entre 21 porosités (9 à l'échelle L₂ et 12 à l'échelle L₃) vers 8 porosités à l'échelle L₃.

Étape no 3 : Utilisation du diagramme#3 de Figure 5.3.

Cette matrice est utilisée pour simuler les porosités de la première colonne ($i>1, j=1$)

Comme on peut le voir dans la Figure 5.7 cette matrice permet de trouver une transition entre 21 porosités (9 à l'échelle L₂ et 12 à l'échelle L₃) vers 8 porosités à l'échelle L₃.

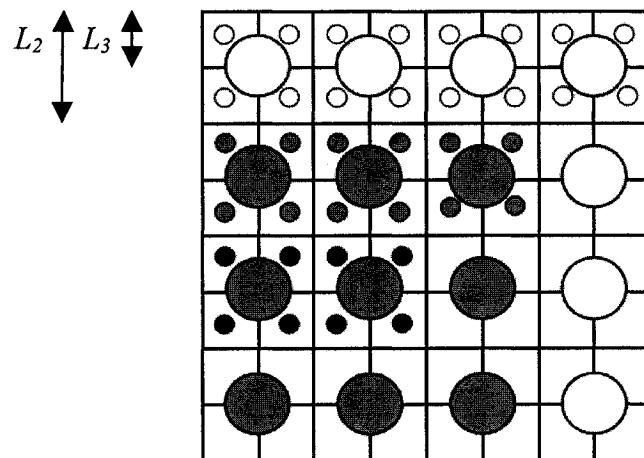


Figure 5.7 Étape 3 : utilisation du diagramme #3.

Étape no 4 : Utilisation du diagramme#3 de Figure 5.3

Cette matrice est utilisée pour simuler également les porosités des fins de lignes ($i>1$, $j=end$).

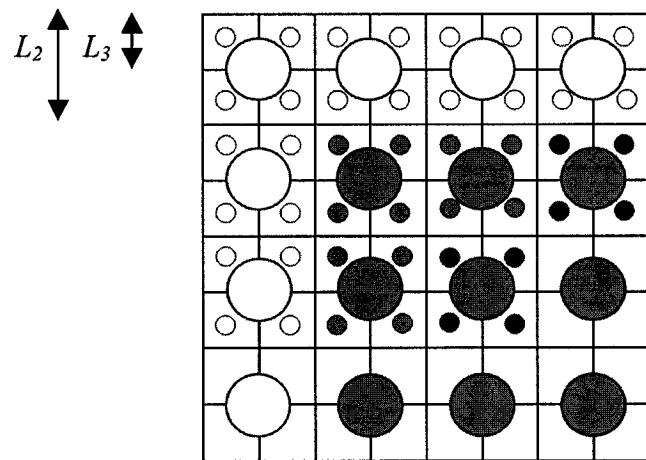


Figure 5.8 Étape 4 : utilisation du diagramme #2.

Comme on peut le voir ci-dessus cette matrice permet de trouver une transition entre 21 porosités (9 à l'échelle L_2 et 12 à l'échelle L_3) vers 8 porosités à l'échelle L_3 .

Ainsi, on voit dans cet exemple comment les porosités à l'échelle L_3 sont partiellement simulées à partir des porosités de l'échelle L_2 . Les porosités à l'échelle L_3 de la dernière ligne et colonne ne sont pas toutes simulées.

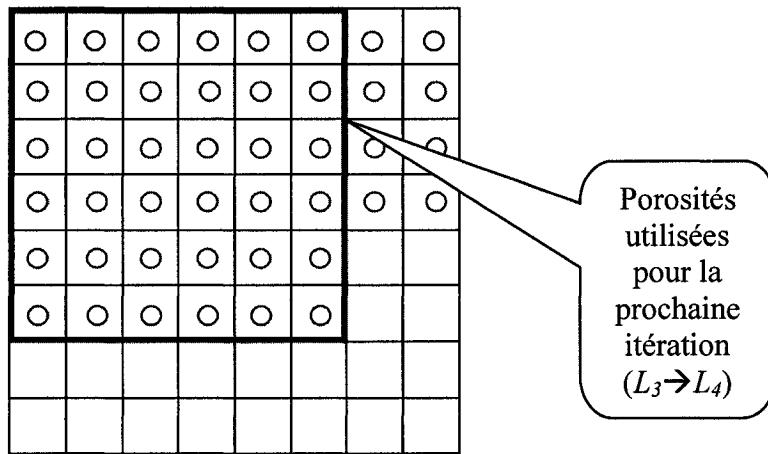


Figure 5.9 Porosités simulées à l'échelle L_3 .

Par la suite, on simulera de la même manière les porosités à l'échelle L_4 , connaissant celles de l'échelle L_3 . On répétera ce procédé jusqu'à obtenir une image à l'échelle du pixel. Tel qu'illustré à la Figure 5.9, on voit qu'à chaque itération les porosités de la dernière ligne et colonne ne sont pas entièrement simulées. C'est pour cette raison que l'image simulée au final est deux fois plus petite que l'image de référence. Cette contrainte sur la taille des images simulées est liée aux schémas de transition utilisés par notre modèle (cf. Figure 5.3). On aurait pu générer des simulations de même taille que l'image de référence en ajoutant d'autres schémas de transition en plus des quatre déjà utilisés. Cependant, les résultats que nous avons obtenus avec le modèle multi-échelle n'étaient pas suffisamment encourageants pour justifier la complication du modèle multi-échelle.

5.2 Résultats

5.2.1 Simulations tests

Les simulations tests présentées dans cette section nous ont permis de valider notre modèle. En effet, les simulations ci-dessous montrent que le modèle permet de reproduire parfaitement des images sous forme de damier jusqu'à une taille limite de quatre pixels. La Figure 5.10 présente les simulations obtenues à partir d'images de

damiers de taille variant de 1 à 4 pixels. Pour chacun des quatre cas, l'image de gauche représente l'image de référence tandis que l'image de droite est la simulation. La taille des images affichées dans cette figure est de 32x32 pixels.

On peut voir dans la Figure 5.10 que les damiers de taille inférieure à 4 pixels sont parfaitement reproduits et ce dans 100 % des cas. De plus, comme les simulations obtenues montrent un décalage comparativement à l'image de référence, cela montre la capacité de généralisation du modèle multi-échelle. Par contre, nous avons constaté que les damiers de 4 pixels n'étaient pas toujours correctement reproduits. Dans la simulation affichée dans Figure 5.10.d), on peut voir l'apparition d'une anomalie dans le coin bas à gauche de l'image. Celle-ci s'explique par l'apparition d'états non observés lors de la transition entre l'avant-dernière échelle et l'échelle du pixel de la simulation. Le problème des états non observés se pose également pour le modèle mono-échelle que nous avons développé. On peut dire que généralement dans les simulations multi-échelle, les états non observés sont rencontrés à des petites échelles. Cependant, lorsque des états non observés sont rencontrés aux grandes échelles, ces derniers se propagent d'une échelle à la suivante en s'amplifiant. Pour les simulations multi-échelle, nous avons également essayé de réduire le problème des états impossibles en appliquant une métrique qui corrige l'état non observé par un état observé ayant le plus grand nombre de bits en commun. Il faut préciser que la métrique utilise une pondération quatre fois plus importante pour les porosités évaluées aux grandes échelles (L_n) comparativement à celles évaluées aux plus petites échelles (L_{n+1}).

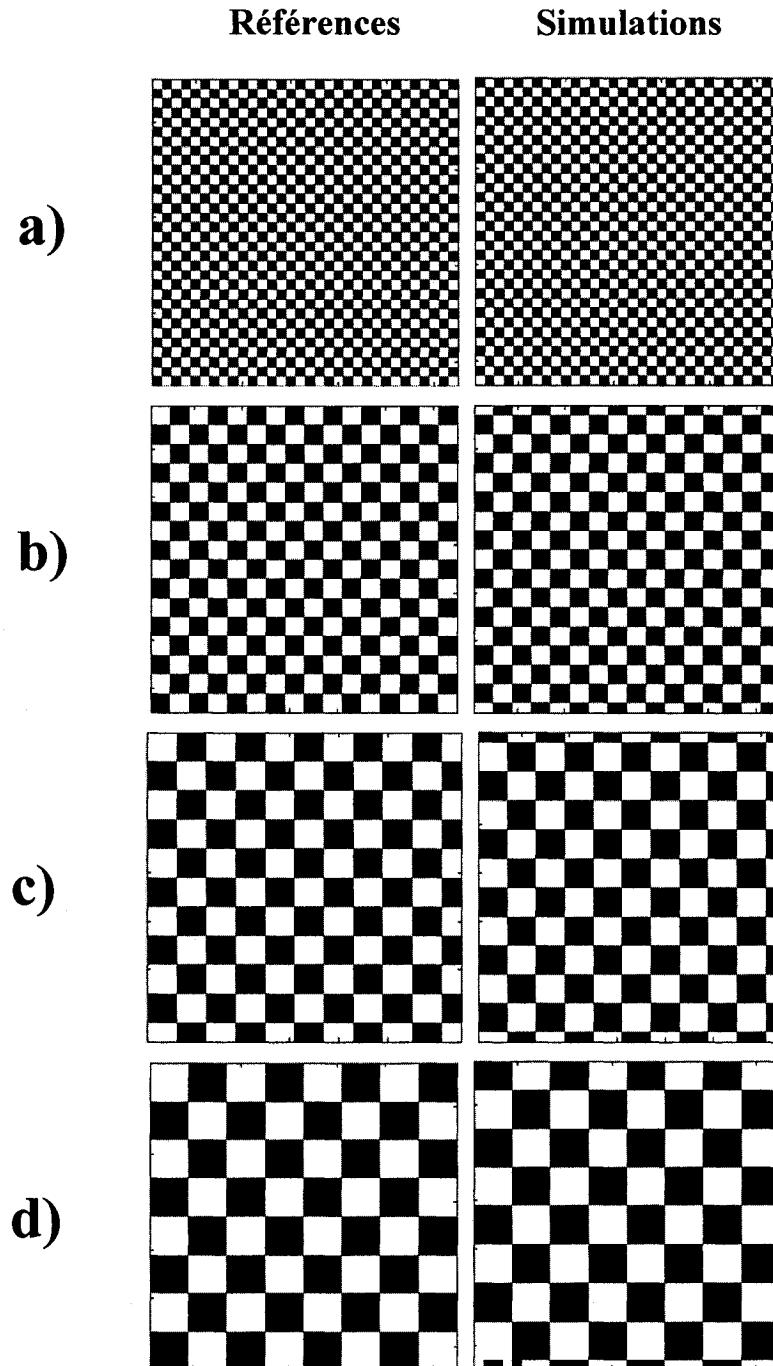


Figure 5.10 Simulations du modèle multi-échelle sur des images de damiers. Le modèle est testé sur des damiers de taille variant de a) 1 à d) 4 pixels. Pour chacun des quatre cas a), b), c) et d) on présente à gauche l'image de référence et à droite on présente la simulation. La taille des images affichées est 32x32 pixels tandis que celle des images de référence utilisées est de 64x64 pixels.

5.2.2 Simulations de disques

Dans cette section, on teste le comportement du modèle multi-échelle sur des images contenant des disques de diamètre fixe disposés aléatoirement dans l'image. Les disques dans l'image de référence ne se superposent pas. La Figure 5.11 présente quelques simulations obtenues à partir d'images contenant des disques dont le diamètre varie de 1 à 4 pixels. La taille des images référence utilisées est de 64x64 pixels tandis que celle des images affichées est de 32x32 pixels.

La Figure 5.11 montre que les disques de 1 pixel sont plutôt bien reproduits à l'exception de quelques pixels qui se touchent. Pour les disques de diamètre supérieur à 1 pixel, la qualité des simulations est moins bonne. En effet, de nombreuses anomalies peuvent être observées : disques coupés, pixels isolés, etc. Les problèmes des états non observés et de perte d'information liée au seuillage des porosités expliquent en grande partie l'apparition de telles anomalies. La qualité de ces simulations est nettement inférieure à celles des simulations obtenues avec l'autre modèle (*cf.* section 4.3.2.1).

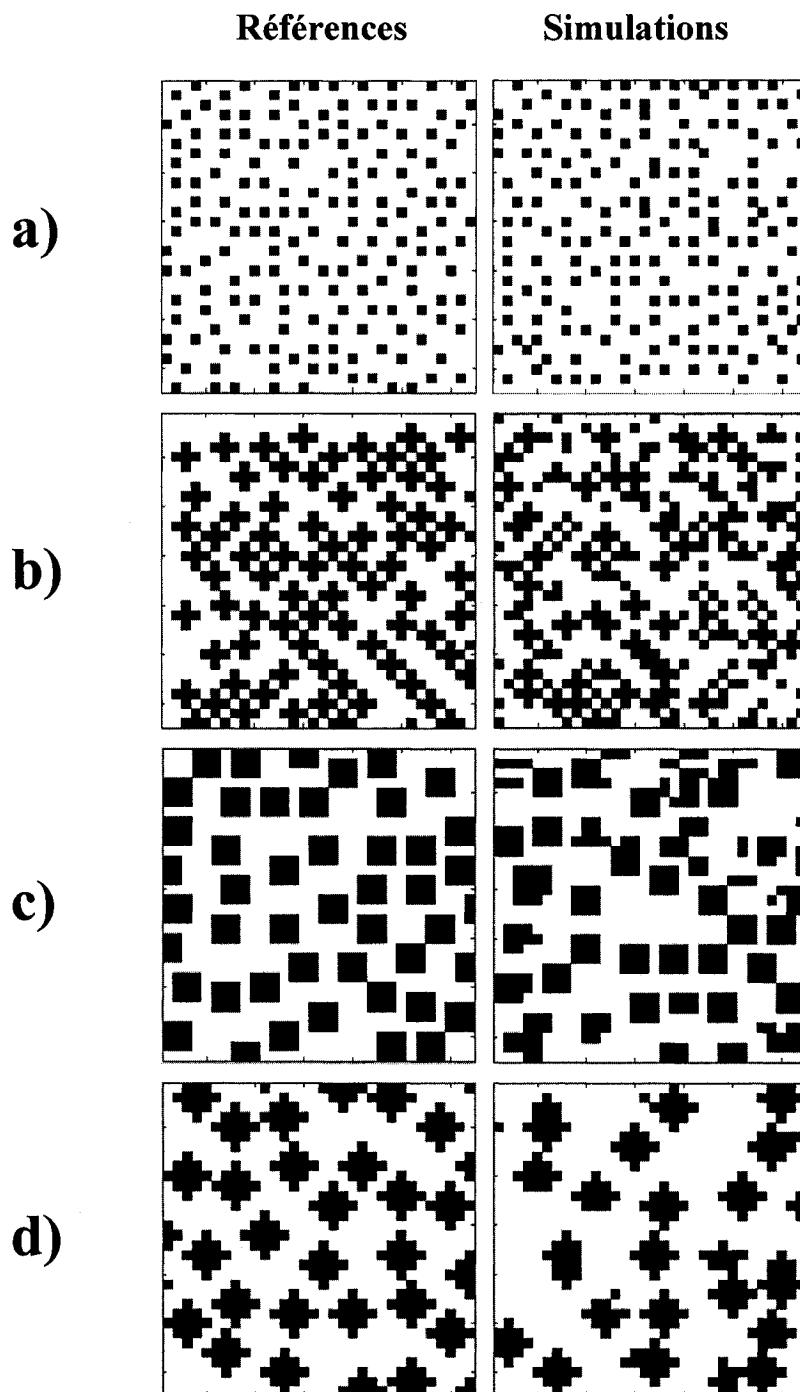


Figure 5.11 Simulations du modèle multi-échelle sur des images de disques. La taille des disques varie de a) 1 à d) 4 pixels. Pour chacun des quatre cas a), b), c) et d) on présente à gauche l'image de référence et à droite la simulation. La taille des images affichées est 32x32 pixels tandis que celle des images de référence utilisées est 64x64 pixels.

5.2.3 Simulations avec une image de polymère

Ci-dessous, on présente l'image de référence utilisée pour la mesure des matrices de transitions. Les images fournies par le laboratoire de recherche sur les polymères sont de tailles 694x503 pixels. Cependant, comme le modèle multi-échelle utilise des échelles diadiques, on a dû travailler sur une partie seulement de l'image de référence de taille 256x256 pixels.

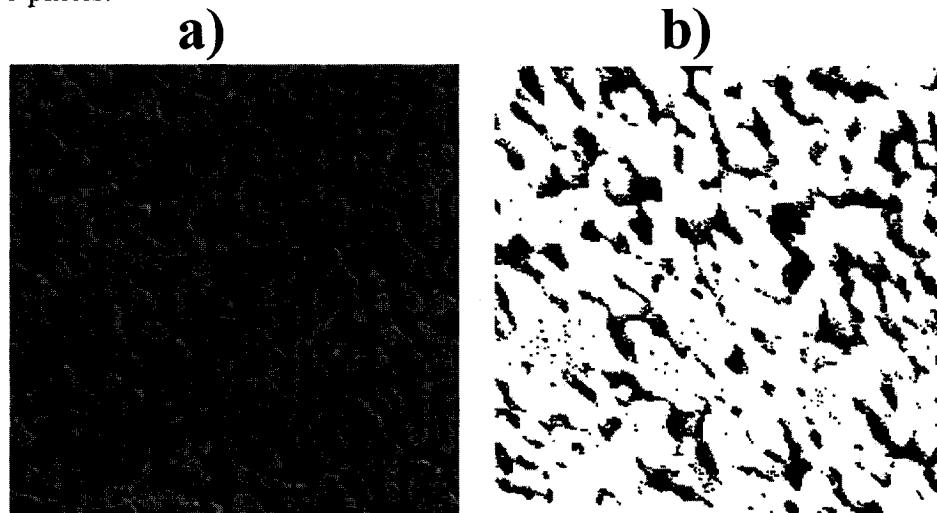


Figure 5.12 Image de l'alliage de polymère avant a) et après seuillage b).

Comme le modèle multi-échelle utilise des images noir et blanc seulement, l'image représentée à la Figure 5.12.a) a été seuillé à l'aide de la médiane, i.e. les pixels en dessous de la médiane ont été arrondis à 0 tandis que le reste des pixels a été arrondi à 1. Le résultat du seuillage est présenté à la Figure 5.12.b). Cette image en noir et blanc a été utilisée pour la mesure des matrices de probabilité de transitions. Ci-dessous, on présente quelques exemples de simulations obtenues par le modèle multi-échelle. Les images simulées sont deux fois plus petites que l'image de référence. Afin de faciliter la comparaison avec la référence, on présente à la Figure 5.13 les simulations de 128x128 pixels ainsi que des régions de 128x128 pixels de l'image de référence affichée à la Figure 5.12.b). On peut voir à la Figure 5.13 que la qualité des simulations obtenues est plutôt limitée. En effet, on peut voir une perte de porosité évidente dans ces simulations. Ces résultats ont en partie expliqué l'abandon de l'approche multi-échelle.

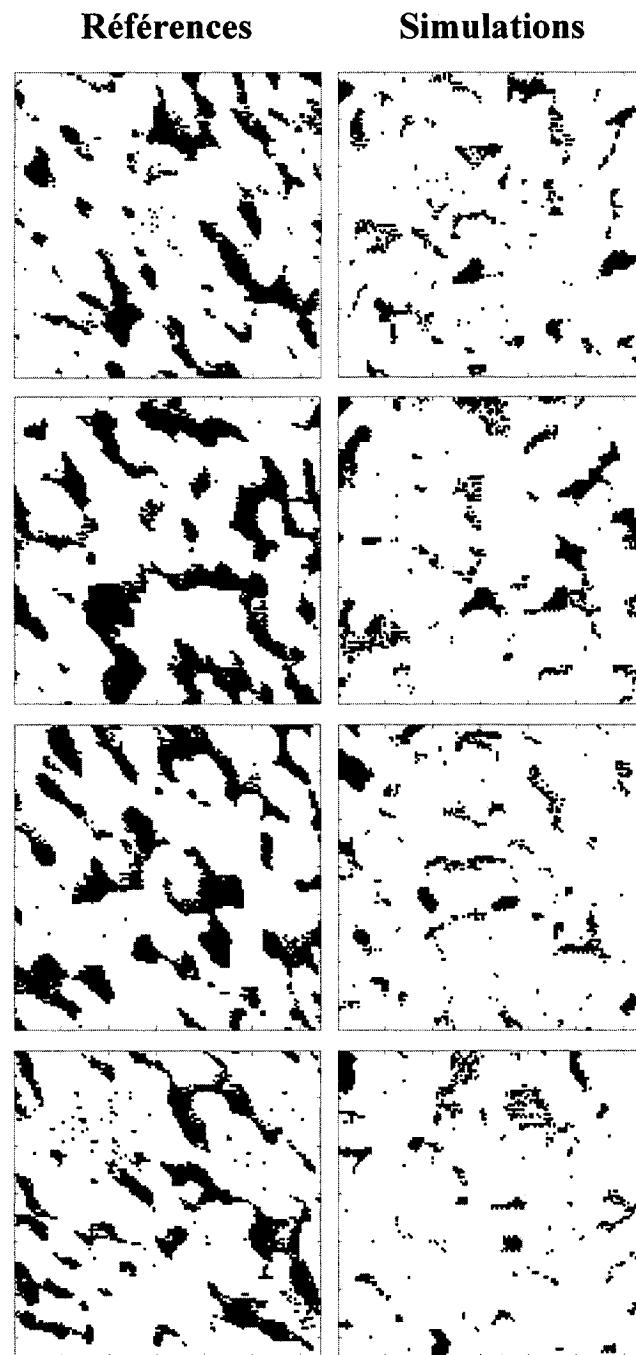


Figure 5.13 Simulations d'une image de polymère avec le modèle multi-échelle. Les images de gauche présentent des régions distinctes de l'image de référence affichée dans Figure 5.12.b). Les images de droites présentent des exemples de simulations obtenues. Les images affichées sont de tailles 128x128 pixels tandis que l'image de référence utilisée (Figure 5.12.b)) est de taille 256x256 pixels.

5.3 Synthèse

Le modèle multi-échelle utilise les probabilités de transition entre porosités arrondies évaluées à des échelles différentes de l'image de référence. C'est un modèle complexe et coûteux en termes de quantité de calculs. Par exemple, pour une image de 64x64 pixels seulement, il faut mesurer en tout $4 \times 4 = 16$ matrices de probabilités de transitions. Ce nombre croît avec la taille de l'image de référence utilisée. De plus, le modèle multi-échelle occasionne une perte d'information à cause du seuillage des porosités. Un autre désavantage du modèle multi-échelle est son incapacité à générer des images de même taille que l'image de référence contrairement au modèle mono-échelle. Enfin, la qualité des simulations obtenues nous a conduits à écarter ce modèle. Cependant, l'approche multi-échelle n'a pas fourni tout son potentiel et l'amélioration de certains points pourrait avoir un effet majeur sur la qualité des simulations produites. En particulier, en réglant les problèmes de perte d'informations liée au seuillage des porosités et de perte de porosité, on pourrait améliorer significativement la qualité des simulations.

CHAPITRE 6 SYNTHÈSE et CONCLUSION

6.1 Discussion

Les résultats de simulation du modèle multi-échelle étaient dans l'ensemble insatisfaisants. En effet, le modèle arrive à reproduire correctement les damiers de taille inférieure à quatre pixels mais réussit moins bien à reproduire correctement les disques de diamètre supérieurs à un pixel. De plus, les simulations de l'alliage de polymère étaient de faible qualité. Ces résultats combinés avec la complexité du modèle nous ont conduits à écarter cette première approche au profit d'une approche mono-échelle moins complexe.

Cette seconde approche, appelée *courtepointe stochastique*, nous a permis de générer des simulations de meilleure qualité que le modèle multi-échelle. Ce modèle se base sur l'extraction de l'image de référence de probabilités conditionnelles entre blocs de pixels adjacents. Les simulations présentées dans l'article montrent la capacité du modèle à reproduire parfaitement les damiers dont le motif de base a une taille inférieure à celle du bloc. Les disques disjoints de diamètre inférieur au double de la taille du bloc sont reproduits avec un faible taux de disques déformés. Pour ces deux types d'images, notre modèle a généré de meilleures simulations que l'algorithme de Strebelle. Cependant, il faut noter que la qualité des simulations se dégrade lorsque le diamètre des disques dépasse la taille du bloc. Les simulations de bâtonnets montrent aussi une dégradation de la qualité des simulations lorsque la longueur des bâtonnets dépasse la taille du bloc. Lorsque la longueur des bâtonnets n'est pas supérieure à la taille du modèle, les simulations obtenues sont de meilleure qualité que celles utilisant l'algorithme de Strebelle.

Enfin, la simulation d'une image de polymère a mis en évidence la capacité du modèle à générer des simulations visuellement comparables à l'image de référence. La

mesure de la distribution des longueurs de cordes dans ces simulations a montré cependant que celle-ci n'était pas reproduite parfaitement. Par contre, l'algorithme de Strebelle réussit à reproduire plus fidèlement cette distribution.

Un avantage majeur de l'approche que nous proposons est sa grande simplicité en plus de la qualité des simulations obtenues pour certains types d'image. Il faut préciser également que la qualité des simulations produites est tributaire d'une bonne estimation des matrices de probabilités de transition. Pour atteindre cet objectif, il faut que les images de référence utilisées soient suffisamment grandes pour permettre une bonne estimation des probabilités de transition entre blocs de l'image.

Une des principales faiblesses de notre modèle est le manque de contrôle sur la porosité des simulations obtenues. Nous avons observé que pour les images de référence de porosité inférieure (resp. supérieure) à 50%, les simulations produites avaient systématiquement une porosité inférieure (resp. supérieure) à celle de l'image de référence. Ce biais de la porosité s'explique par notre approche de correction des états non observés qui consiste à remplacer systématiquement tout état non observé par l'état observé qui se raccorde le mieux aux blocs de pixels déjà simulés. En procédant de la sorte, on favorise la reproduction des états les plus fréquemment observés dans l'image de référence. Nous avons partiellement remédié à ce problème en ajoutant une contrainte de filtrage de porosité lors de la correction des états non observés. L'état observé est ainsi choisi de manière à avoir la bonne porosité tout en se raccordant au mieux aux blocs de pixels déjà simulés. L'algorithme de Strebelle offre aussi la possibilité d'exercer un certain contrôle sur la porosité des simulations générées.

Une autre faiblesse de notre approche de simulation est son incapacité à générer des simulations conditionnelles. Il est toutefois possible de faire des simulations conditionnelles en choisissant les blocs simulés qui respectent les conditions spécifiées. En effet, les blocs simulés ne seront plus choisis en fonction de leurs propriétés probabilistes seulement mais aussi en fonction de leur respect des points conditionnant.

Actuellement, notre modèle traite uniquement des images en noir et blanc. Cette limitation est liée à la programmation actuelle du modèle qui utilise un codage binaire pour chaque bloc. Le codage de chaque bloc sur une base plus élevée nous permettrait de traiter des images avec plusieurs tons de gris. Une autre approche consisterait à identifier chaque bloc, non pas par sa configuration de pixels mais par sa position dans l'image, éliminant ainsi le problème du codage de bloc.

Par ailleurs, le modèle que nous proposons se limite aux images stationnaires. En effet, on fait l'hypothèse que les propriétés statistiques de l'image sont les mêmes partout dans l'image.

Enfin, le problème fondamental que nous avons en commun avec l'algorithme de Strebelle est celui des états non observés qui apparaissent lors d'une simulation. La métrique que nous avons proposée pour corriger les états non observés améliore considérablement la qualité des simulations produites. Cependant, en corrigéant les états non observés on modifie les propriétés statistiques de l'image, notamment sa porosité.

6.2 Conclusion et Recommandations

Dans ce projet, nous avons travaillé sur différentes approches de simulations stochastiques d'images bidimensionnelles. Les méthodes multi-échelles développées n'ont pas été concluantes. Par contre, la méthode mono-échelle que nous avons développée semble prometteuse. Il s'agit d'une méthode simple reposant sur l'extraction de probabilités conditionnelles entre blocs de pixels adjacents de l'image de référence. Notre approche produit de meilleures simulations que l'algorithme de Strebelle pour certains types d'images comme les damiers ou encore les agrégats de taille relativement petite par rapport à celle du modèle. Par contre, lorsque la taille des objets augmente comparativement à celle du modèle, on observe une dégradation de la qualité des simulations.

Il reste des points à améliorer dans notre approche, le plus important étant le contrôle de la porosité des simulations produites. La programmation du modèle de manière à pouvoir traiter des images avec plusieurs tons de gris constitue également un de nos objectifs à court terme.

Enfin, la généralisation de notre modèle à des images tridimensionnelles est possible. En effet, il suffira de calculer les probabilités de transition entre des blocs tridimensionnels de pixels au lieu de blocs bidimensionnels. Actuellement, il n'existe pas de technique fiable pour reconstruire des images tridimensionnelles à partir d'images bidimensionnelles. Une fois que nous aurons amélioré les points faibles de notre modèle, nous pourrons concentrer nos efforts sur une technique qui permette la transition de blocs bidimensionnels à des blocs tridimensionnels de pixels.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] ADLER, P.M. (1992). *Porous Media*. Butterworth-Heinemann, Boston.
- [2] ADLER, P.M., JACQUIN, C. and QUIBLIER, C.J. (1990). Flow in simulated porous media. *International Journal of Multiphase Flow*, 16:691–712.
- [3] ARNÉODO, A., BACRY, E., and MUZY, J.F. (1995). The thermodynamics of fractals revisited with wavlets. *Physica A*, Vol. 213, pages. 232-275.
- [4] ARPAT, B. (2003). *Report No. 16: A pattern recognition approach to multiple-point simulation*. Stanford Center for Reservoir Forecasting, Stanford University.
- [5] BAKKE, S. and ØREN, P. (1997). 3-d pore-scale modeling of sandstones and flow simulations in pore networks. *SPE Journal*, 2:136.
- [6] BEAR, J. and VERRUIJT, A. (1987). *Modeling Groundwater Flow and Pollution*. Kluwer Academic Publishers, Dordrecht.
- [7] BREIMAN, L., FRIEDMAN, L., OLSHEN, R. and STONE, C. (1984). *Classification and regression trees*, publ. Wadsworth, Monterrey.
- [8] BRYANT, S., MELLOR, D. and CADE, C. (1993). Physically representative network models of transport in porous media. *AIChe Journal*, 39:387.
- [9] CAERS, J. (1998). Stochastic simulation using neural networks. In *Stanford Center for Reservoir Forecasting Annual Meeting 11*, pages 1–66. Stanford University, Vol. 2.
- [10] CAERS, J. and JOURNEL, A.G. (1998). Stochastic reservoir simulation using neural networks trained on outcrop data. *SPE paper no. 49026*.

- [11] CHATZIS, I. and DULLIEN, F. (1977). Modelling pore structure by 2-d and 3-d networks with applications to sandstones. *Journal of Canadian Petroleum Technology*, page 97.
- [12] DEUTSCH, C. and JOURNEL, A. (1998). *GSLIB: Geostatistical Software Library*. Oxford University Press, New York. Second Edition.
- [13] DEUTSCH, C.V. (1992). Annealing techniques applied to reservoir modeling and the interpretation of geological and engineering (well test) data. Unpublished doctoral dissertation, page 306, Stanford University.
- [14] FARMER, C.L. (1998). The generation of stochastic fields of reservoir parameters with specified geostatistical distributions. In Edwards, S., and King, P.R, eds., *Mathematics in oil production*, pages 235-252, Clarendon Press, Oxford.
- [15] FATT., I. (1956). The network models of porous media i. capillary pressure characteristics. *AIME Petroleum Transactions*, 207:144.
- [16] GUARDIANO, F. and SRIVASTAVA, R.M. (1993). Multivariate geostatistics: Beyond bivariate moments. In A. Soares, editor, *Geostatistics-Troia, vol. 1*, pages 133–144, Dordrecht, Netherland. Kluwer Academic.
- [17] HALDORSEN, H. and DAMSLETH, E. (1990). Stochastic modeling. *Journal of Petroleum Technology*, pages 404–412.
- [18] KATZ, A. and THOMPSON, A. (1986). Quantitative prediction of permeability in porous rock. *Phys. Rev. B*, 34:8179.

- [19] LAKE, L. (1989). *Enhanced Oil Recovery*. Prentice Hall, Englewood Cliffs.
- [20] LI, J., MA, P.L. and FAVIS, B.D. (2002). The role of the blend interface type on morphology in cocontinuous polymer blends. *Macromolecules*, 35:2005–2016.
- [21] OMRE, H. (1991). Stochastic models for reservoir characterization. In J. Kleppe and S. M. Skjæveland, editors, *Recent Advances in Improved Oil Recovery Methods for North Sea Oil Sandstone Reservoirs*, Dordrecht, Netherland. Norwegian Petroleum Directorate.
- [22] REMY, N. (2004). *Stanford Geostatistical Modeling Software (S-GeMS), version 1.2, build*. Contributors: Joy Rajiv and Michael Sharps. Available on the Internet at http://prdownloads.sourceforge.net/sgems/sgems_installer-1.2.exe?download
- [23] ROBERTS, J. and SHWARTZ, L. (1985). Grain consolidation and electrical conductivity in porous media. *Phys. Rev. B*, 31:5990.
- [24] ROY, S. and TRAFALDAR, S. (1997). Archies's law from a fractal model for porous rock. *Phys. Rev. B*, 55:8038.
- [25] SAHIMI, M. (1993). Flow phenomena in rocks: From continuum models to fractal, percolation, cellular automata and simulated annealing. *Rev. Mod. Phys.*, 65:1393.
- [26] SARAZIN, P. and FAVIS, B.D. (2003). Morphology control in co-continuous poly (L-lactide)/polystyrene blends: a route towards highly structured and interconnected porosity in poly (L-lactide) materials. *Biomacromolecules*, 4:1669–1679.

- [27] SARAZIN, P. and FAVIS, B.D. (2005). Influence of temperature-induced coalescence effects on co-continuous morphology in poly (ε - caprolactone)/polystyrene blends. *Polymer*, 46:5966–5978.
- [28] SAUCIER, A. and MULLER, J. (1993). Remarks on some properties of geometrical multifractals. *Physica A*, 199:350–362.
- [29] SAUCIER, A. and MULLER, J. (1999). Textural analysis of disordered materials with multifractals. *Physica A*, 268:221–238.
- [30] SAUCIER, A., RICHER, J. and MULLER, J. (2002). Assessing the scope of the multifractal approach to textural characterization with statistical reconstruction of images. *Physica A*, 311/1-2:231–259.
- [31] STOYAN, D., KENDALL, W. and MECKE, J. (1987). *Stochastic Geometry and its Applications*. Wiley, Berlin, Chichester. Second Edition.
- [32] STREBELLE, S. (2002). Conditional simulation of complex geological structures using multiple point statistics. *Mathematical Geology*, 34(1):1–22.
- [33] STREBELLE, S. (2003). New multiple-point statistics simulation implementation to reduce memory and cpu-time demand. Presented at the 2003 conference of the International Association for Mathematical Geology, in Portsmouth, UK, September 7–12. Available on the Internet at <http://wwwIAMG.org/meetings/Proceedings2003/papers/Strebelle2.pdf>.
- [34] TORQUATO, S. (2000). Modeling of physical properties of composite materials. *International Journal of Solids and Structures*, 37:411–422.

- [35] TERZOPOULOS, D. and METAXAS, D. (1991). Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.13, no.7, pages 703-714.
- [36] VEMURI, B.C. and RADISAVLJEVIC, A. (1994). Multiresolution Stochastic Hybrid Shape Models with Fractal Priors. *ACM Transactions on Graphics*, vol. 13, no. 2, pages 177–207.

ANNEXE I

Code de la méthode séquentielle de simulation :

Courtepointe Stochastique

Dans cette annexe, on présente le code MATLAB utilisé pour générer les simulations affichées dans le chapitre 5 du mémoire. Les simulations ont été générées principalement à l'aide des fonctions *Scanner()* et *Simuler(..)*. La première fonction est utilisée comme son nom l'indique pour scanner une ou des images de référence afin d'en extraire les matrices de probabilités de transition. La seconde fonction quant à elle est utilisée pour générer des simulations à partir des matrices de probabilités de transition fournies par la fonction *Scanner()*. Les fonctions auxiliaires, *poids()* et *le_plus_proche()* utilisées par la fonction *Simuler()* ont été placés également dans cette annexe.

```

function [mat]=Scanner(I,np,iso);
% DESCRIPTION: Fonction qui détermine pour une ou des images de référence les 3 % matrices des probabilités de transition du modèle mono échelle.
% PARAMETRE : I --> l'image de référence ou les images de référence
%           np --> taille du bloc utilisé
%           iso --> type de l'image
%           iso=1 si image isotrope
%           iso=0 si image anisotrope
% RETOUR :      mat --> Structure de données contenant les matrices de
%                  probabilités de transition ainsi que les matrices de
%                  probabilité d'occurrence des états rencontrés

nl=size(J,1); % Nombre de ligne de l'image de référence
nc=size(J,2); % Nombre de colonne de l'image de référence
nz=size(J,3); % Nombre d'images de référence

% déclaration des matrices de probabilité de transition (vides)
m=struct('trs',-1,'eta',-1);
mat(1:3)=m;
x=0;
% déclaration de compteurs
cmpt=[2 1 1];
% type de scann :
iso=iso*3+1;

% Pour toutes les images de référence
for f=1:nz
    I=J(:,:,f);

```

```

% Type de scan (isotropique/anisotropique)
for k=1:iso
    k
    I=rot90(I,k);
    nl=size(I,1);
    nc=size(I,2);
    for i=1:nl-2*np+1
        for j=1:nc-2*np+1
            % les états correspondant aux modèle séquentiel
            e1=I(i:i+np-1,j:j+np-1);
            e2=I(i:i+np-1,j+np:j+2*np-1);
            e3=I(i+np:i+2*np-1,j:j+np-1);
            e4=I(i+np:i+2*np-1,j+np:j+2*np-1);
            % Affectation des états de départ et d'arrivée dans la variable etat
            etat(1,1)=bi2de(e1(:))+1;
            etat(1,2)=bi2de(e2(:))+1;
            etat(2,1)=bi2de(e3(:))+1;
            etat(2,2)=bi2de(e4(:))+1;
            etat(3:4,1)=[etat(1,1:2)';
            etat(3:4,2)=etat(2,1);
            etat(5:7,1)=[etat(1,1);etat(1,2);etat(2,1)];
            etat(5:7,2)=etat(2,2);
            z=1;
            for t=1:3
                y=1;
                while y<=cmpt(t)
                    x(1:t,1:2)=etat(z:z+t-1,:);
                    z=z+t;
                % Construction des matrices de transition
                if mat(t).eta==-1
                    mat(t).trs(1:t,1:3)=[x(1:t,:); cat(1,1,zeros(t-1,1))];
                    mat(t).eta(1:t,1:2)=[x(1:t,1); cat(1,1,zeros(t-1,1))];
                else
                    b=0;
                    a=reshape(mat(t).eta(:,1),t,[]);
                    if t==1
                        b=find(mat(t).eta(:,1)==x(t,1));
                    else
                        b=find(all(a==x(:,1)*ones(1,size(a,2))));
```

end

```

                    if isempty(b)
                        mat(t).trs(length(mat(t).trs(:,1))+1:length(mat(t).trs(:,1))+t,1:3)=[x(1:t,:); cat(1,1,zeros(t-1,1))];
```

```

mat(t).eta(length(mat(t).eta(:,1))+1:length(mat(t).et
a(:,1))+t,1:2)=[x(1:t,1) cat(1,1,zeros(t-1,1))];

else
    b=(b-1)*t+1;
    mat(t).eta(b,2)=mat(t).eta(b,2)+1;
    b=0;
    a=[reshape(mat(t).trs(:,1),t,[]);reshape(mat(t).trs(:,2
),t,[])];
    if t==1
        b=find(mat(t).trs(:,1)==x(t,1) &
        mat(t).trs(:,2)==x(t,2));
    else
        b=find(all(a==x(:)*ones(1,size(a,2))));
    end
    if isempty(b)
        mat(t).trs(length(mat(t).trs(:,1))+1:length(m
at(t).trs(:,1))+t,1:3)=[x cat(1,1,zeros(t-1,1))];
    else
        b=(b-1)*t+1;
        mat(t).trs(b,3)=mat(t).trs(b,3)+1;
    end
end
end
x=0;
y=y+1;
end
end
end
end
end
end
end
end % Fin Boucle nz

```

```

% Boucle qui calcule les probabilités d'occurrences et de
% transition pour chacune des trois matrices.
for i=1:3
    % probabilité d'occurences des états
    mat(i).eta(:,2)=mat(i).eta(:,2)/sum(mat(i).eta(:,2));
    a=[reshape(mat(i).trs(:,1),i,[])];
    % détermination des probabilités de transition
    for j=1:length(mat(i).eta(:,1))/i
        if i==1
            b=find((a==mat(i).eta(i*(j-1)+1:i*j,1)*ones(1,size(a,2))));
        else

```

```

b=find(all(a==mat(i).eta(i*(j-1)+1:i*j,1)*ones(1,size(a,2))));
end
b=(b-1)*i+1;
mat(i).trs(b,3)=cumsum(mat(i).trs(b,3))/sum(mat(i).trs(b,3));
end

end

```

```

function [phi,stat]=Simuler(I,mat,np,pmax,cp,param_a,filt);
% DESCRIPTION: Fonction qui reconstruit une image en utilisant les matrices de
% probabilité de transition fournies par la fonction précédente
% Scanner(). Dans cette fonction on utilise la métrique qui corrige %
les états impossibles par des états observés à l'aide d'une
distribution linéaire des poids (Cf fonction les_poids()).

% PARAMETRE : I --> Image de référence.
% mat --> structure contenant les 3 matrices de transition et
% d'occurrence du modèle séquentiel.
% np --> taille du modèle séquentiel
% cp --> pourcentage de coupure pour la porosité
% param_a --> paramètre d'ajustement pour le filtrage
% filtr --> booléenne indiquant le type de simulation effectuée.
% simulation avec filtrage si filtr=1
% simulation sans filtrage si filtr=0

% RETOUR : phi --> Image reconstruite par la méthode de courtepoinette stochastique.
% stat--> Résultats statistiques sur l'image simulé

% REMARQUE: Mise en œuvre de métrique qui utilise une variation linéaire des
% poids avec en plus le respect de la porosité de l'état non observé.

```

```
I=logical(I);
N=log2(size(I,1)); % échelle max de I
nl=floor(size(I,1)/np);
nc=floor(size(I,2)/np);
```

```
% Variables globales pour statistiques  
global stat;  
stat=zeros(1,7);
```

% Variables locales
Ptr=0; % matrice des probabilités de transition

```

Peta=0; % matrice des probabilités des états
q=0;
t=0;
phi=logical(ones(nl*np,nc*np));

% Selection aléatoire d'un bloc dans l'image
x=ceil(rand(1,2).*[(nl)*(np-1)-1 (nc)*(np-1)-1]);
phi(1:np,1:np)=I(x(1)+1:x(1)+np,x(2)+1:x(2)+np);

% Determination des poids pour la métrique #1 (frontière exterieure)
[p1,p2]=les_poids(np,pmax);
p1=reshape(p1',1,3*np^2);

% ensemble des états possible a l'echelle n
e2=logical(de2bi(mat(2).eta(:,1)-1,np^2+1)); e2=e2(:,1:end-1);
e2=logical(reshape(e2',2*np^2,[])');
% porosité des états possible associé la seconde matrice
nb2=2*np^2-sum(e2,2);
nb2(:,2)=mat(2).eta(1:2:end,2); % fréquence d'occurrence des états

% ensemble des états possible a l'echelle n
e3=de2bi(mat(3).eta(:,1)-1,np^2+1); e3=e3(:,1:end-1);
e3=logical(reshape(e3',3*np^2,[])');
% porosité des états possible associé la troisième matrice
nb3=3*np^2-sum(e3,2);
nb3(:,2)=mat(3).eta(1:3:end,2); % fréquence d'occurrence des états

% taille des matrices
t2=length(mat(2).trs)/2;
t3=length(mat(3).trs)/3;

% Taille de l'image à simuler
nl=ceil(size(I,1)/np);
nc=ceil(size(I,2)/np);
% Pour tous les blocs à simuler
for i=1:nl
    for j=1:nc
        % Determination de la matrice de probabilité de transition approprié
        if (i==1)
            t=1;
            Ptr=mat(1).trs;
            Peta=mat(1).eta;
            x=phi(1:np,1+(j-1)*np:np*j);
            x=bi2de(x(:)')+1;

```

```

% Determination de l'état d'arrivée
y=Ptr(min(find(Ptr(:,1)==x & (Ptr(:,3)-rand >=0))),2);
if isempty(y)
    fprintf('vide');
    %0X
    y=Ptr(min(find(Ptr(:,2)==x & (Ptr(:,3)-rand >=0))),1)
    if ~isempty(y)
        fprintf('plus vide');
    else
        y=Peta(ceil(rand*length(Peta)),1);
    end
end
elseif ((i>1 & j==1))
    t=2;
    e=e2;
    nb=nb2;
    Ptr=mat(2).trs;
    Peta=mat(2).eta;
    x1=phi(1+np*(i-2): np*(i-1),1:np);
    x2=phi(1+np*(i-2): np*(i-1),np+1:np*2);
    x=[bi2de(x1(:)')+1;bi2de(x2(:)')+1];
    b=0;
    c=1;
    a=reshape(Ptr(:,1),2,[]);
    b=2*(find(all(a==x(:,1)*ones(1,size(a,2))))-1)+1;
    c=min(find(Ptr(b,3)-rand>=0));
    y=Ptr(b(c),2);
elseif(i>1 & j>1)
    t=3;
    x1=phi(1+np*(i-2):np*(i-1),1+np*(j-2):np*(j-1));
    x1=bi2de(x1(:)')+1;
    x2=phi(1+np*(i-2):np*(i-1),1+np*(j-1):np*(j));
    x2=bi2de(x2(:)')+1;
    x3=phi(1+np*(i-1):np*i,1+np*(j-2):np*(j-1));
    x3=bi2de(x3(:)')+1;
    x= [x1;x2;x3];
    e=e3;
    nb=nb3;
    Ptr=mat(3).trs;
    Peta=mat(3).eta;
    b=0;
    c=1;
    a=reshape(Ptr(:,1),3,[]);
    b=3*(find(all(a==x(:,1)*ones(1,size(a,2))))-1)+1;

```

```

c=min(find(Ptr(b,3)-rand>=0));
y=Ptr(b(c),2);
end
% Cas ou l'état x est impossible (non observé)
if ( isempty(y))
    q=q+1;
    % Détermination de la liste d'états les plus proches
    m=le_plus_proche(e,x,p1,p2,cp,nb,param_a,filtre);
    % Choix d'un des états les plus proches
    f=min(find((cumsum(Peta(m,2))/sum(Peta(m,2)))-rand >0));
    % Choix d'un état selon la loi de probabilité
    m=Peta(m(f):m(f)+t-1,1);
    b=0;
    c=1;
    a=reshape(Ptr(:,1),t,[]);
    b=t*(find(all(a==m*ones(1,size(a,2))))-1)+1;
    c=min(find(Ptr(b,3)-rand>=0));
    y=Ptr(b(c),2);
    m=logical(de2bi(m-1,np^2+1));% Conversion en binaire
    m=m(:,1:end-1);

    % Correction de l'état non observé par un état observé
    if t==3
        phi(1+np*(i-2):np*(i-1),1+np*(j-2):j*np)=[reshape(m(1,:),np,[])]
        reshape(m(2,:),np,[])];
        phi(1+np*(i-1):np*i,1+np*(j-2):(j-1)*np)=[reshape(m(3,:),np,[])];
    else
        phi(1+np*(i-2):np*(i-1),1:2*np)=[reshape(m(1,:),np,[])]
        reshape(m(2,:),np,[])];
    end
    m=bi2de(m);
end
y=y-1;
y=logical(de2bi(y,np^2+1));
y=y(1:end-1);
% Ecriture de l'état simulé
if (t==1)
    phi(1+np*(i-1):np*i,1+(j)*np:np*(j+1))=reshape(y,np,[]);
else
    phi(1+np*(i-1):i*np,1+np*(j-1):j*np)=reshape(y,np,[]);
end
end
end

```

```
% Affichage du nombre d'états impossibles rencontrés
q
% Calculs statistiques
stat(1)=stat(1)/max(1,stat(2));
stat(3:4)=stat(3:4)/q;
phi=phi(1:size(I,1),1:size(I,2));
stat(5)=count(phi)/(size(phi,1)*size(phi,2));
stat(6)=stat(2)/max(1,q);
stat(7)=q/(nl*nc);
%----- stat : variable globale -----%
% stat(1) = porosité moyenne des états filtrés %
% stat(2) = nombre de filtrage %
% stat(3) = porosité moyenne des états impossibles %
% stat(4) = porosité moyenne des états impossibles %
% stat(5) = porosité de la simulation %
% stat(6) = proportion de filtrage avec succès %
% stat(7) = proportion des états impossible %
%----- %
% Affichage des résultats statistiques de la simulation
stat
% Affichage de l'image simulée
figure;image(rot90(phi,1)*255);colormap('gray');
```

```
function [ind]=le_plus_proche(etat_po,etat_impo,p1,p2,cp,nb,a,filtr);
% DESCRIPTION: Fonction qui retourne les indices des états les plus proches d'un
% état impossible selon la métrique spécifié, de plus l'état est choisi %
de manière à avoir une porosité proche de celle de l'état
impossible.
% PARAMETRES: etat_po : matrice contenant la liste des états possibles sous
% forme binaire.
% etat_impo: vecteur de 1x2 ou 1x3 contenant l'état sous forme
décimale.
% p1 : poids associé à la troisième matrice de probabilité
% transition (3 vers 1).
% p2 : poids associé à la deuxième matrice de probabilité
% de transition (2 vers 1).
% cp : pourcentage de coupure pour reproduire la porosité %
de l'état impossible.
% nb : matrice contenant la porosité et la fréquence
% d'apparition des états possibles.
% a : paramètre d'ajustement pour le filtrage
% RETOUR: ind: vecteur contenant la liste des indices des états les plus proches
% selon la métrique définie.
```

```

% déclaration de variable globale
global stat;
% Nombre d'état possible
nb_l=size(etat_po,1);
% Nombre de bits sur lesquels ils sont codés
nb_col=size(etat_po,2);
% type d'état impossible
t=size(etat_impo,1);
% Initialisation de variables locales
d=-1;
% Choix des poids à appliquer
if t==2
    p=p2;
else
    p=p1;
end

% Conversion de l'état impossible en binaire
etat_impo=de2bi(etat_impo-1,nb_col/t+1);
etat_impo=etat_impo(:,1:end-1);
etat_impo=reshape(etat_impo',1,nb_col);
etat_impo=logical(ones(nb_l,1)*etat_impo);
d=(etat_impo==etat_po);

% variable distance
dist=zeros(nb_l,3);
% indices correspondants a chacun des états
dist(:,3)=[1:nb_l]';

% Calcul de la distance selon la métrique
dist(:,1)=d*p';

%-----FILTRAGE-----
% On retient les états dont la différence de porosité relative à
% celle de l'état impossible est inférieure à un certain seuil.
%-----%
if filtr==1
    % nombre d'éléments nul dans l'état impossible
    nbi=count(etat_impo(1,:));

```

```

% porosité de l'état impossible : stat
stat(3)=stat(3)+ (nbi/nb_col);
% différence de porosités entre les états impossibles et possibles
dist(:,2)=abs(nb(:,1)-a*nbi);

% indices des états possible ayant une différence de porosité relative <= cp %
ind=find(dist(:,2)<=a*cp*nbi);
if ~isempty(ind)
    dist=dist(ind,:);
    % nombre de filtrage
    stat(2)=stat(2)+1;
    % porosité moyenne des états filtrés
    stat(1)=stat(1)+ sum(nb(dist(:,3),1))/(length(dist(:,1))*nb_col);
end
end

% Détermination des indices des états les plus proche selon la métrique
ind=dist(find(dist(:,1)==max(dist(:,1))),3);

% porosité moyenne des états observés sélectionnées après application de la métrique
stat(4)=stat(4)+ sum(nb(ind,1))/(length(ind)*nb_col);

ind=(ind-1)*t+1;

function [x,x4,poids]=les_poids(np,pmax)
% AUTEUR : EL OUASSINI AYOUB
% DATE : 13 aout 2005
% DESCRIPTION: Fonction qui détermine les poids à utiliser pour calculer la
%               distance entre un état impossible et une série d'états possible. Ces %
% poids sont inversement proportionnels à la distance qui les sépare
%               des frontières d'un état donné.
% PARAMETRE : np --> Taille des blocs
%               pmax --> Poids maximal (>=1)
% RETOUR :      x,x4 --> Poids associés aux différents bits d'un état.
% NB : Les pods sont calculés pour deux types d'états.

poids=zeros(3*np,2*np);
c=pmax;
if np==1

```

```
a=0;
else
    a=(pmax-1)/(np-1);
end
for i=1:np
    poids(i,i:np)=c;
    poids(i:2*np,i)=c;
    poids(2*np+i,1:2*np-i)=c;
    poids(2*np+i:3*np,2*np-i+1)=c;
    c=c-a;
end
poids(1:np,np+1:2*np)=fliplr(poids(1:np,1:np));
x1=poids(1:np,1:np);
x2=poids(1:np,np+1:2*np);
x3=poids(1+np:2*np,1:np);
x4=poids(2*np+1:3*np,1:end);
x1=x1(:)';
x2=x2(:)';
x3=x3(:)';
x4=x4(:)';
x=cat(1,x1,x2,x3);
```