

UNIVERSITÉ PARIS 8 - VINCENNES À SAINT-DENIS

**Doctorat  
en Informatique**

**Farès BELHADJ**

Modélisation automatique de géo-environnements  
naturels et multi-urbains

Soutenue le 11 décembre 2007

COMPOSITION DU JURY

Didier ARQUÈS	Rapporteur	Université de Marne la Vallée
Eric GALIN	Rapporteur	Université Claude Bernard
Stéphane MÉRILLOU	Examineur	Université de Limoges
Vincent BOYER	Examineur (invité)	Université Paris 8
Patrick GREUSSAY	Directeur	Université Paris 8
Pierre AUDIBERT	Co-directeur (invité)	Université Paris 8

directeurs : Pierre AUDIBERT  
Patrick GREUSSAY



# RÉSUMÉ

Nous présentons de nouveaux modèles de génération de terrains naturels et urbains que nous nommons géo-environnements. Ces modèles ont tous été développés, testés et optimisés sur les ordinateurs du département d'Informatique de l'Université de Paris 8 à Saint-Denis. Nos modèles statistiques ont permis d'obtenir des gains de temps allant jusqu'à 17000% en comparaison avec la meilleure méthode connue dans un problème de reconstruction de données satellites à partir d'échantillons. Au delà de l'efficacité, nos algorithmes donnent les résultats suivants : un bon degré de variabilité des données initiales traitées et la capacité de générer intégralement des modèles réalistes de paysages érodés ; à partir d'une méthode inédite, la *méthode des écoulements*, nous engendrons des modèles-squelettes décrivant des lignes de crêtes et des bassins versants sous la forme de contraintes locales.

Pour les environnements urbains, nous proposons un processus générant un plan d'occupation des sols à partir d'un modèle de terrain. Nous générons successivement la carte de densité de population, les emplacements et les dimensions des sites d'agglomération et enfin le réseau routier les reliant en considérant les données du relief.

Nous proposons enfin des extensions essentielles pour les champs d'application des domaines connexes : notre algorithme fractal peut s'appliquer à la reconstruction d'un modèle de terrain à partir de données partielles, ces données sont soit sous-échantillonnées, soit squelettisées puis amplifiées via notre modèle ; nous proposons également des applications dans le domaine de la reconstruction de photographies et en rendu non-photoréaliste.

Le chapitre 1 décrit les méthodes de génération de relief, en indique les limites et introduit une première méthode fractale prenant en compte l'érosion des terrains générés.

Au chapitre 2, nous proposons notre méthode des écoulements en réponse à un problème de contraintes : comment se passer des modèles complexes de simulation d'érosion et obtenir des reliefs plausibles et contraignables ?

Le chapitre 3 présente le *déplacement des milieux morphologiquement contraint* : notre réponse à la généralisation du problème de contraintes.

Le chapitre 4 entame la problématique de l'urbanisation entièrement automatique du terrain : nous y proposons des solutions aux différents stades de la planification.

Et en conclusion, nous présentons des applications, actuelles et futures, de nos méthodes de génération.



# Remerciements

Je remercie Monsieur Didier Arquès d'avoir accepté de rapporter sur mon travail et de me faire l'honneur de présider mon jury de thèse.

Je remercie Monsieur Eric Galin d'avoir bien voulu être rapporteur de mon travail.

Je remercie Monsieur Stéphane Mérillou de sa participation au jury.

Monsieur Pierre Audibert a soutenu depuis mes débuts mon travail de recherche et a dirigé ce travail en m'orientant dans le meilleur domaine à étudier.

Monsieur Vincent Boyer a été un soutien constant, je le remercie de son amitié et de ses suggestions très fructueuses.

Je remercie Monsieur Patrick Greussay de son soutien, de sa confiance et d'avoir accepté de co-diriger mon travail.

Je remercie tous mes collègues, amis et amies du laboratoire d'intelligence artificielle de Paris 8.

Je remercie ma grand-mère, mes parents, ma soeur et ma famille pour leur soutien constant.

A ma femme, Isma, mon fils, Neil, ainsi que le bout de chou à venir.





2.2.1.2	Le calcul rapide des bassins versants . . . . .	46
2.2.2	Hydrologie de surface . . . . .	47
2.2.2.1	Notre approximation du modèle physique . . . . .	51
2.2.2.2	Retour sur trace et validation . . . . .	57
2.2.3	Modèle de terrain à crêtes et réseau fluvial : l'érosion . . . . .	58
<b>3</b>	<b>Le modèle fractal fortement contraint : vers la continuité</b>	<b>65</b>
3.1	Introduction à l'approche bas-haut-bas . . . . .	65
3.1.1	Notre approche globale . . . . .	66
3.1.1.1	En une dimension . . . . .	66
3.1.1.2	En deux dimensions . . . . .	68
3.1.1.3	Notre algorithme bas-haut-bas . . . . .	71
3.1.2	Second modèle de terrain à crêtes et réseau fluvial . . . . .	73
3.2	La généralisation de l'approche : vers le contrôle de la forme . . . . .	77
3.2.1	La modification de la courbe d'interpolation . . . . .	79
3.2.2	La modification du calcul d'extrapolation . . . . .	80
3.2.3	Notre algorithme généralisé et optimisé . . . . .	82
<b>4</b>	<b>Du terrain naturel au terrain urbanisé</b>	<b>87</b>
4.1	Nos motivations : un problème d'intégration . . . . .	88
4.2	Densité de population . . . . .	89
4.2.1	Les quatre caractéristiques des unités de terrain urbanisé : les sommets du graphe . . . . .	89
4.2.2	Les arcs d'un graphe d'urbanisation : liaisons fluviales et hydrométrie . . . . .	92
4.2.2.1	Notre implémentation du modèle hydrométrique . . . . .	92
4.2.2.2	Application de l'hydrométrie dans le calcul du MNDPH . . . . .	93
4.3	Le réseau d'agglomérations . . . . .	98
4.3.1	Position et configuration des sites d'agglomérations . . . . .	98
4.3.1.1	Distribution utilisée : une approximation de Zipf . . . . .	98
4.3.1.2	Positionnement des sites . . . . .	102
4.3.2	Connecter les sites d'urbanisation . . . . .	106
4.4	Finalisation du réseau : le calcul des routes . . . . .	108
4.4.1	Quelles solutions disponibles ? . . . . .	110
4.4.2	Notre approche de la problématique des routes . . . . .	111
4.4.3	Une première solution au problème des routes . . . . .	112
4.4.4	Résultats et discussion . . . . .	115
4.5	État de l'art . . . . .	117

<b>5 Conclusion et projets</b>	<b>121</b>
5.1 Le générateur de modèles . . . . .	122
5.1.1 Le générateur de terrains . . . . .	122
5.1.2 Le générateur de matériels . . . . .	126
5.2 La reconstruction des surfaces . . . . .	129
5.2.1 La reconstruction à partir d'un sous échantillonnage aléatoire . . . . .	129
5.2.2 La reconstruction à partir des contours . . . . .	134
5.3 De la peinture impressionniste aux planètes artificielles : l'uni- vers dans une graine . . . . .	137
5.4 Projets . . . . .	145
<b>Bibliographie</b>	<b>149</b>



# Introduction

Pourquoi automatiser la conception de terrains, de reliefs, de rivières ? Ce ne sont pas les réponses qui manquent dans le cas des applications : simulateurs de vols, de climat, d'hydrologie, d'agriculture, d'urbanisme, de jeux, de guerre. L'horizon de telles études est évident, c'est le temps réel : le paysage, encore inconnu, que déroule le simulateur est réellement conçu au cours du déplacement, puis modifié à mesure que le temps virtuel passe. Nous n'en sommes pas encore là : le temps réel est possible, nous l'avons abordé dans notre travail de Maîtrise puis de DEA qui précédait cette thèse, et appliqué à des simulateurs de vol. Les cas simples de paysages sont spectaculaires mais restent suffisamment incomplets pour exiger une recherche à double objectif visant la fidélité à la Nature et l'efficacité vis à vis des modèles de simulation physique : des paysages réalistes doivent être faisables sans la complexité, la lourdeur et le pas fin imposé par la discrétisation des équations de la mécanique des fluides.

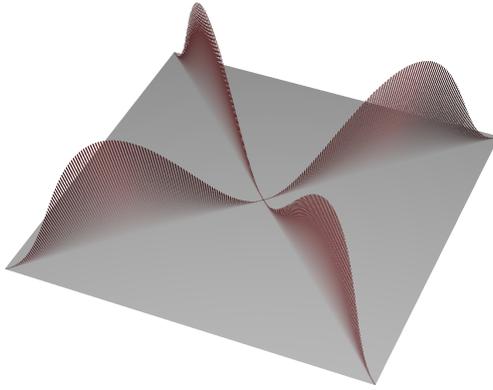
Commençons<sup>1</sup> par présenter un aperçu des résultats obtenus<sup>2</sup> au cours de notre travail de thèse.

La figure 1 illustre une partie de la problématique concernant la génération, sous contraintes, de modèles fractals de terrain. Les contraintes d'altitudes sont posées au premier chapitre (cf. figure 1(a)) : deux diagonales de contraintes sont posées et la deuxième diagonale est décalée d'une unité à

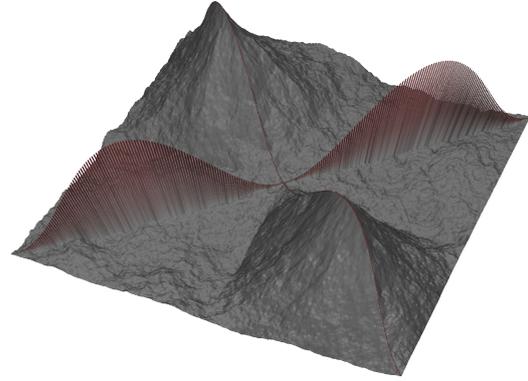
---

1. Cette introduction est rédigée conformément à la méthode suivie dans toutes les thèses et mémoires du Laboratoire d'Intelligence Artificielle de Paris 8 : se présentant comme un résumé très amplifié, elle se compose des éléments les plus saillants des chapitres qui suivront, pour donner à un lecteur un aperçu précis des apports de notre travail, sans devoir parcourir nécessairement l'ensemble de la thèse. Pour cette raison, nos résultats scientifiques sont présentés généralement en premier, et les états de l'art suivront. Notre introduction est donc une instance de cette méthode de présentation propre au laboratoire, elle-même le fruit de nombreuses années de réflexion, d'essais et erreurs.

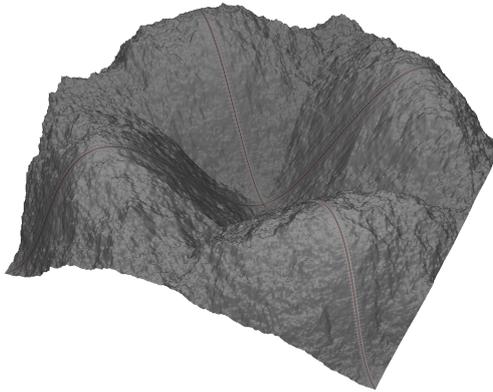
2. L'ensemble de nos méthodes a été développé, testé et optimisé en C ANSI sur *Intel(R) Pentium(R) IV*, la totalité des programmes se chiffre à environ 40000 lignes de code C.



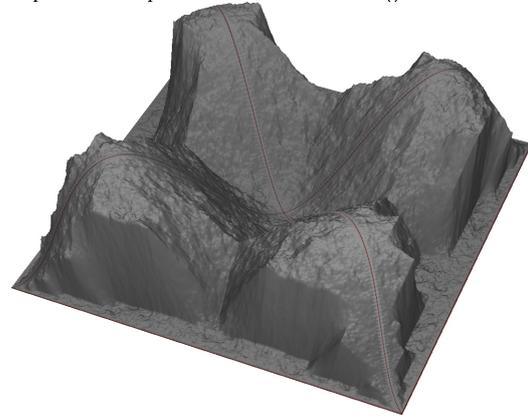
(a) Un modèle-squelette de contraintes: en rouge, les altitudes imposées. La contre-diagonale (bas-gauche à haut-droit) est décalée d'une unité à droite.



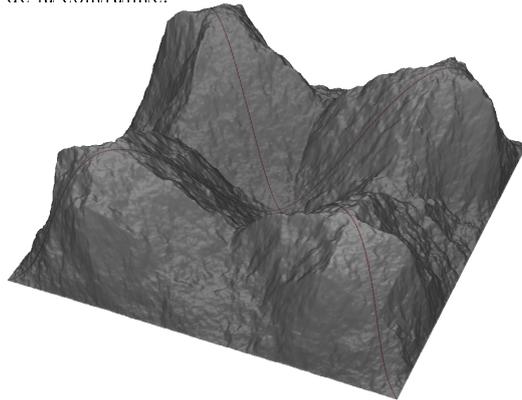
(b) La surface obtenue après l'application de notre implémentation d'un déplacement des milieux standard: la contre-diagonale génère des ruptures abruptes en raison du décalage.



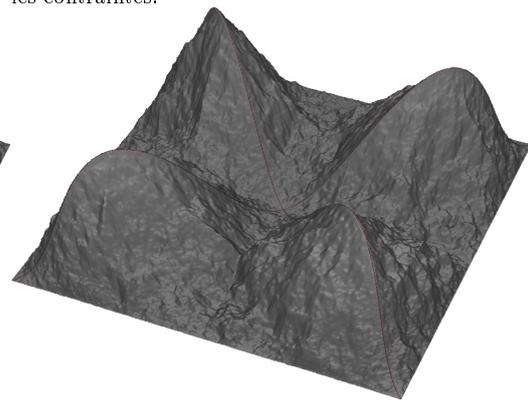
(c) Première solution au problème d'altitudes imposées: notre méthode par fenêtrage croissant construit la surface, de proche en proche, à partir de la contrainte.



(d) Durcissement de la contrainte, des altitudes nulles sont imposées aux bords: la méthode par fenêtrage croissant ne prévoit pas la jonction entre les contraintes.



(e) Notre approche non-locale précalcule, pour des positions clés de la méthode de déplacement des milieux, des altitudes permettant de réaliser la jonction: les problèmes de ruptures à la jonction des contraintes disparaissent.



(f) Notre *déplacement des milieux morphologiquement contraint* est la généralisation de l'approche non-locale: la méthode permet, en plus des contraintes d'altitudes imposées, de modifier l'allure de la surface obtenu.

FIG. 1 – *Progression des solutions apportées à la problématique: satisfaire les contraintes d'altitudes imposées sous la forme d'un modèle-squelette et en faire un terrain fractal.*

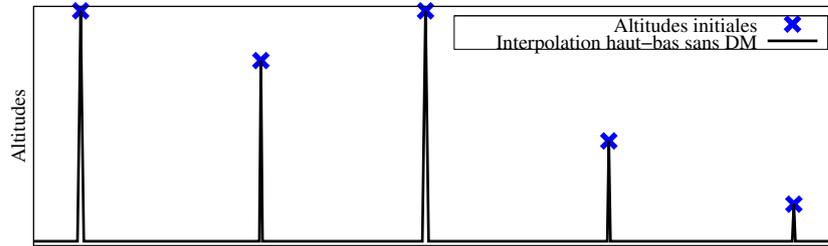
droite, elles représentent quatre crêtes. Nous montrons que nos implémentations des déplacements des milieux produisent des ruptures sur la contre-diagonale (cf. figure 1(b)) car cette dernière est décalée, i.e. les altitudes sous-jacentes ne représentent plus des milieux consécutifs du processus de subdivision. Nous proposons une première solution au chapitre 2 : notre modèle fractal utilise une approche en *fenêtrage croissant*, il construit progressivement le terrain à partir de la contrainte (cf. figure 1(c)). Ce modèle produit des ruptures aux jonctions de contraintes concurrentes : des contraintes aux bords sont ajoutées à partir de la figure 1(d), i.e. quand les altitudes imposées sont numériquement opposées mais localement proches<sup>3</sup>. C'est au chapitre 3 que nous proposons un modèle fractal qui, à lui seul, résout la problématique : une première implémentation règle, en figure 1(e), le problème de ruptures produites ; nous généralisons la méthode et la nommons le *déplacement des milieux morphologiquement contraint* (DMMC), cette dernière permet, en plus, de modifier l'allure de la surface générée (cf. figure 1(f)).

La figure 2 illustre l'utilisation de notre approche non-locale, présentée au chapitre 3, dans le cas de la génération d'une courbe fractale où des positions quelconques sont contraintes à des altitudes imposées. La figure 2(a) montre le résultat obtenu en calculant les moyennes des altitudes des extrema des sous-intervalles, ces derniers sont donnés par les subdivisions successives, par deux, de l'intervalle initial. En figure 2(b) le même processus est réalisé mais avec déplacement des milieux (DM). Dans les deux derniers cas, des pics abrupts apparaissent aux positions contraintes, i.e. la contrainte n'est pas prise en compte par son voisinage. Notre nouvelle méthode réalise un prétraitement sur l'intervalle donné, ajoutant des contraintes à celles initialement imposées : ce processus remonte, dans l'arbre des subdivisions, l'information liée à la contrainte et extrapole les altitudes des extrema des sous-intervalles. Cette diffusion de l'information vers le haut permet, lors du processus de subdivision, de prendre en compte les altitudes initialement imposées (cf. les figures 2(c) et (d)).

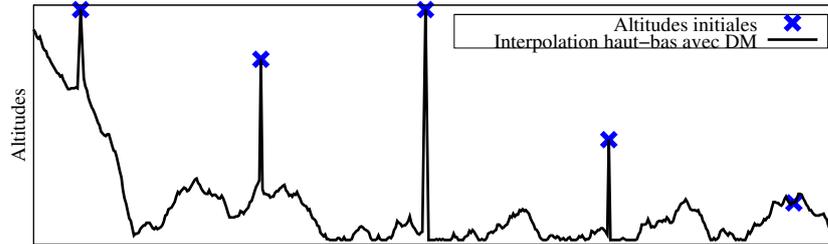
La figure 3 introduit un modèle de terrain entièrement généré. Nous utilisons la méthode des écoulements, présentée en section 2.2, pour engendrer un modèle-squelette décrivant des lignes de crête et un réseau fluvial (cf. figure 3(a)). La méthode DMMC est appliquée au modèle-squelette pour générer l'ensemble de la surface (cf. figure 3(b)).

---

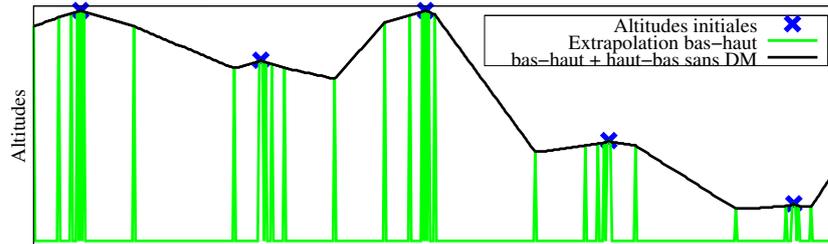
3. C'est ici que nous avons eu l'idée d'introduire notre méthode des écoulements, elle fait la jonction entre ces contraintes et résout, dans le cas de modèles de terrain érodé, le problème de ruptures.



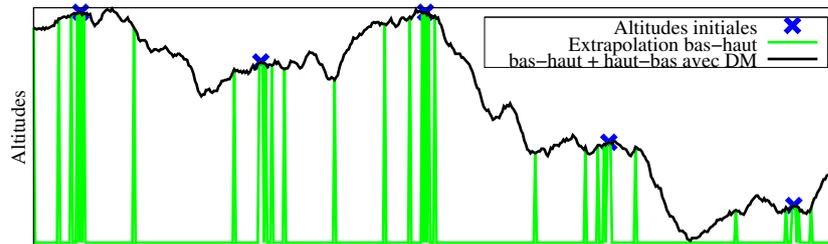
(a) Subdivision standard sans déplacement des milieux : l'intervalle initial est coupé au milieu en deux sous-intervalles, le milieu prend l'altitude moyenne des extrema, et l'opération est répétée récursivement sur les sous-intervalles.



(b) Subdivision standard avec déplacement des milieux : l'intervalle initial est coupé au milieu en deux sous-intervalles, le milieu prend l'altitude moyenne des extrema plus un déplacement vertical proportionnel à la taille de l'intervalle, et l'opération est répétée récursivement sur les sous-intervalles.

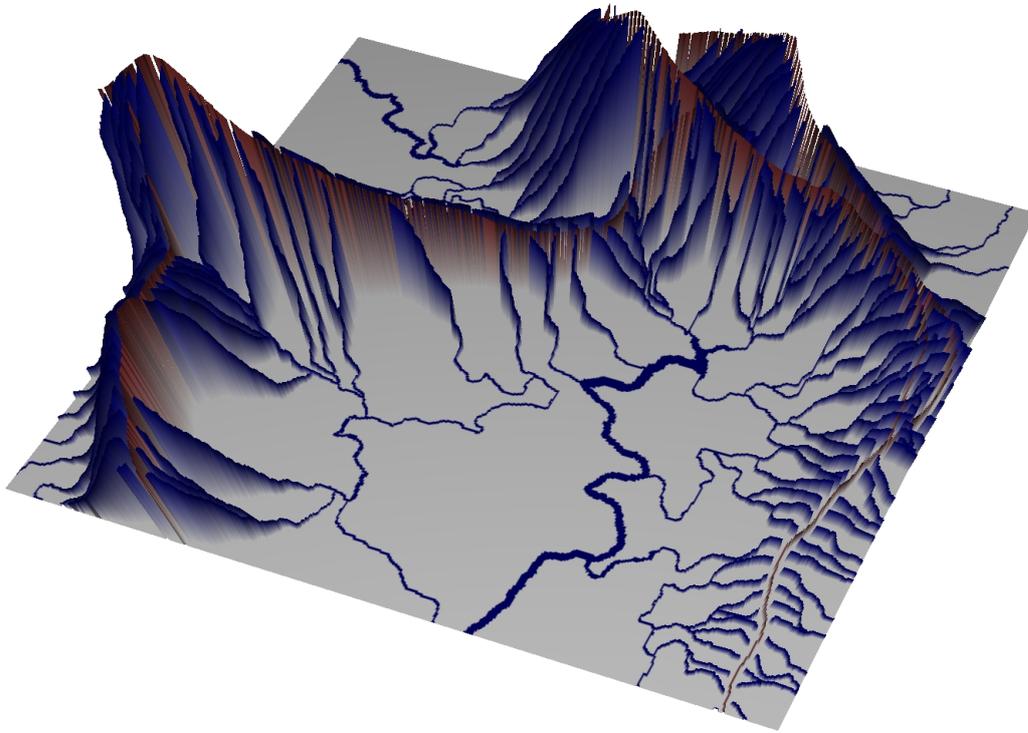


(c) Prétraitement bas-haut des contraintes initiales : des altitudes (les barres verticales vertes) sont ajoutées (extrapolées) à des positions clés de l'intervalle; ces positions sont obtenues en parcourant l'arbre de subdivision en sens inverse, i.e. bas-haut. Enfin, une subdivision standard sans déplacement des milieux est réalisée pour l'intervalle ainsi enrichi en altitudes imposées.

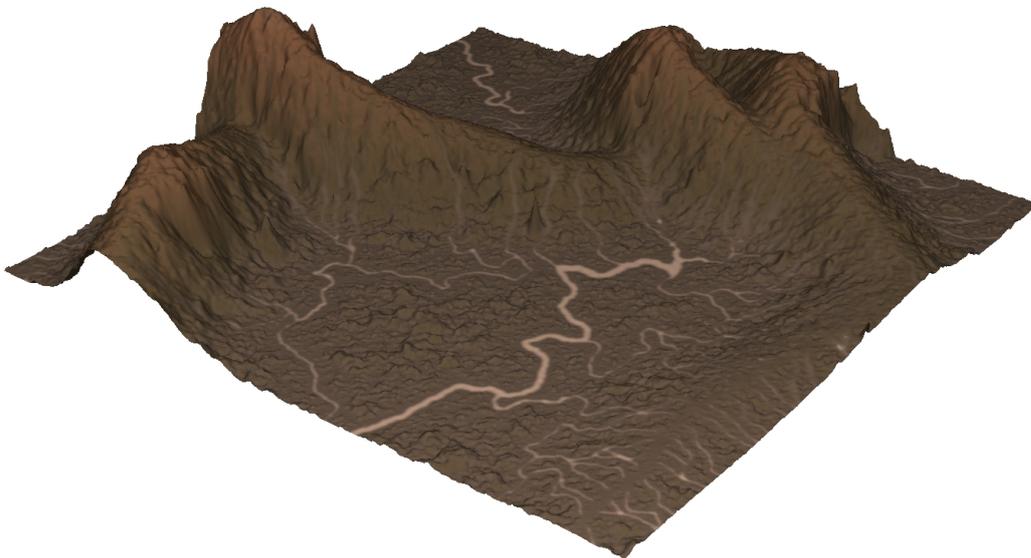


(d) Prétraitement bas-haut des contraintes initiales : des altitudes (les barres verticales vertes) sont ajoutées à des positions clés de l'intervalle; ces positions sont obtenues en parcourant l'arbre de subdivision en sens inverse, i.e. bas-haut. Enfin, une subdivision standard avec déplacement des milieux est réalisée pour l'intervalle ainsi enrichi en altitudes imposées.

FIG. 2 – Comment amener un modèle standard de subdivision à satisfaire les contraintes d'altitudes imposées (voir les croix libellées altitudes initiales) ? Les graphiques, répétés ici, illustrent l'application de notre approche bas-haut-bas présentée au chapitre 3.

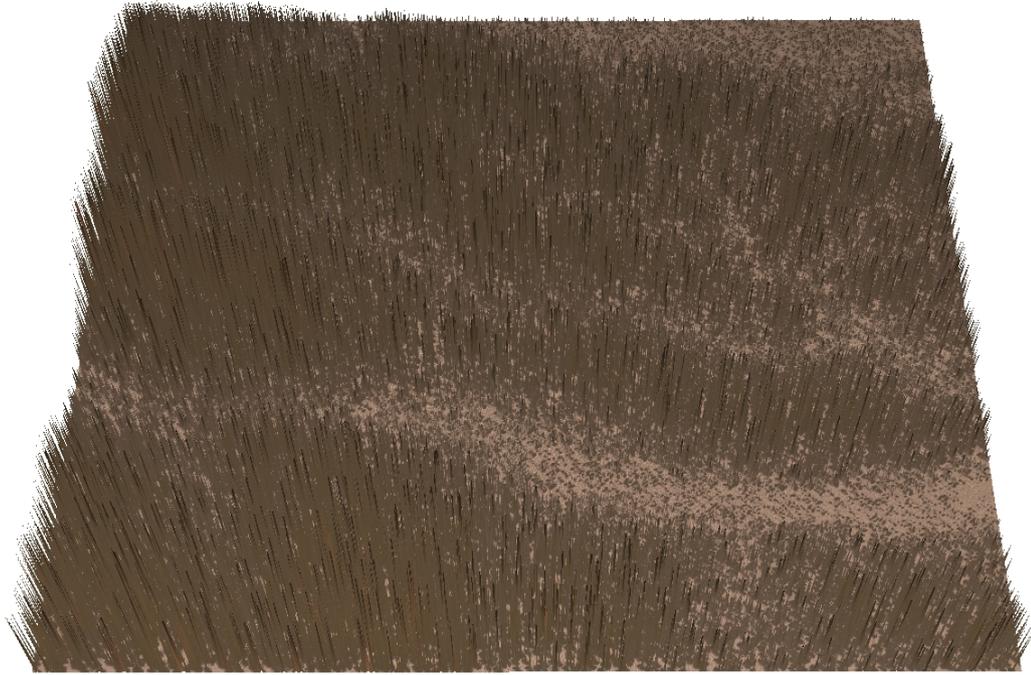


(a) Modèle-squelette obtenu, en 487 millisecondes (sur notre architecture de référence présentée en section 1.1.1.2), par notre *méthode des écoulements*.

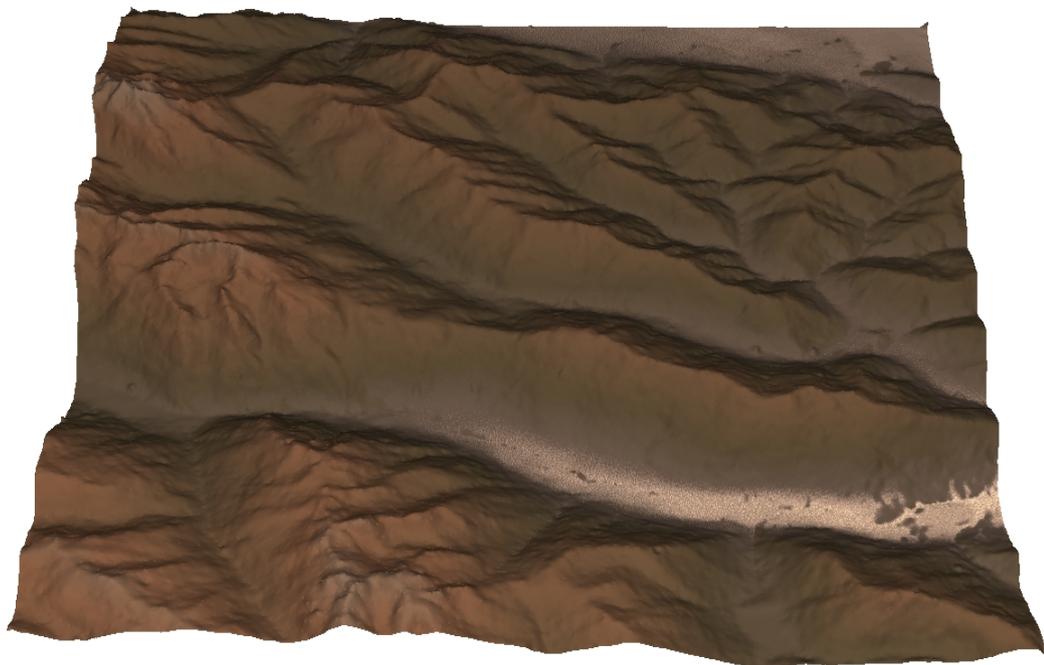


(b) Le terrain est finalisé, 107 millisecondes (sur notre architecture de référence présentée en section 1.1.1.2), par notre *déplacement des milieux morphologiquement contraint*.

FIG. 3 – *Modèle de terrain entièrement généré.*



(a) Modèle numérique de terrain aléatoirement sous-échantillonné à 3.3% de la donnée initiale; cette donnée satellite provient de l'USGS [129] et représente le Mont Washington (état du New Hampshire, USA).



(b) Notre *déplacement des milieux morphologiquement contraint* reconstruit la donnée sous-échantillonnée: l'écart type, par rapport à la donnée originale (i.e. non sous-échantillonnée) est de 5.92 mètres. Nous réalisons, en section 5.2.1, pour ce même modèle et dans des conditions analogues, une comparaison avec la méthode présentée dans [95].

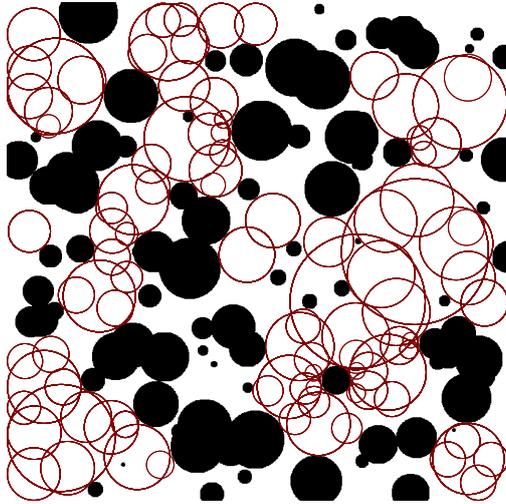
FIG. 4 – *Reconstruction de données satellites partielles : ici la donnée originale est délibérément sous-échantillonnée en vue de réaliser une comparaison avec la donnée reconstruite.*

Nous illustrons une des principales applications de notre méthode DMMC en figure 4 : un modèle de terrain représentant le Mont Washington (état du New Hampshire, USA) est aléatoirement sous-échantillonné à 3.3% (le modèle initial est une donnée satellite provenant de l'USGS [129]) puis reconstruit à l'aide de notre algorithme fractal DMMC présenté en section 3.2.3. Notre méthode est comparée, en section 5.2.1, aux résultats obtenus par J. Pouderoux et al. dans [95] où une approche par réseaux de neurones est utilisée pour approximer la même surface sous-échantillonnée dans les mêmes conditions : notre méthode est 170 fois plus rapide.

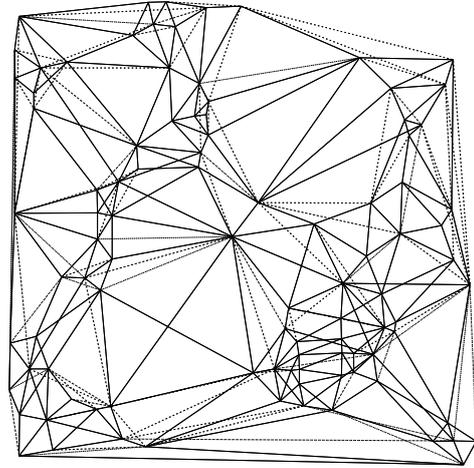
Nous terminons par la figure 5 illustrant le processus, présenté au chapitre 4, d'urbanisation automatique d'un modèle de relief : les sites d'agglomération (i.e. les unités urbaines représentées par des cercles) sont placés en fonction d'une carte de densité de population<sup>4</sup> et leurs dimensions sont fixés en fonction d'un quota déterminé selon une approximation de données statistiques de l'INSEE relevant du recensement de la population française de mars 1999 [55] (cf. section 4.3.1) ; ces sites décrivent une relation de hiérarchie liée à la relation d'inclusion au sens géométrique, nous calculons, en fonction de cette hiérarchisation, plusieurs réseaux planaires décrivant les connexions entre les sites puis nous simplifions le réseau en fonction de l'apport de chacune des connexions (cf. section 4.3.2) ; le réseau est finalisé par le calcul des trajectoires représentant les routes, nous proposons un nouvel algorithme rapide, basé sur une recherche stochastique construisant incrémentalement des courbes de Bézier reliant les sites et prenant en compte les contraintes du relief, ces dernières sont données sous la forme de coût d'aménagement des routes (ou liaisons).

---

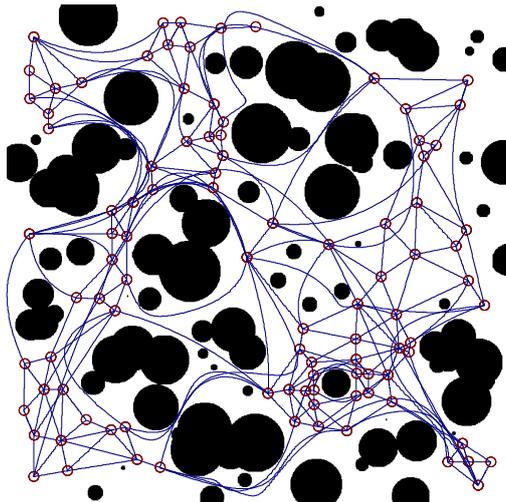
4. Le calcul de la carte de densité de population est réalisé par un processus automatique prenant en compte quatre caractéristiques liées au relief et à l'hydrométrie elle-même calculée (cf. section 4.2.1).



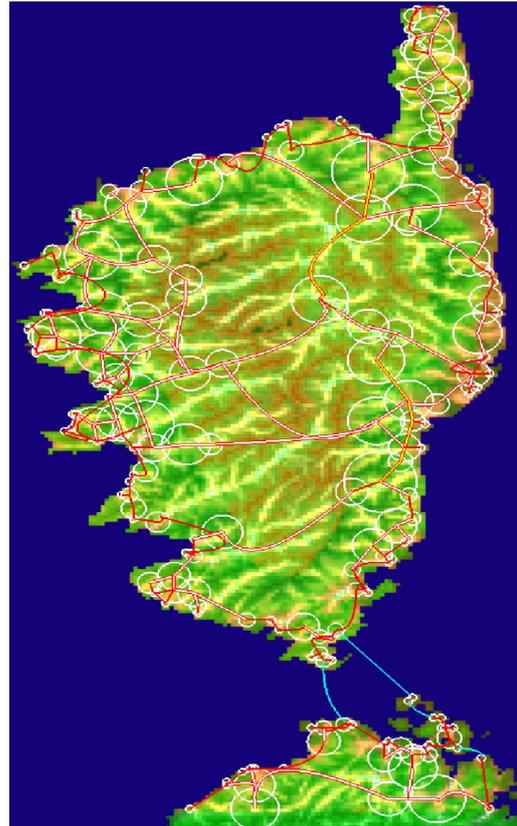
(a) Distribution automatique des sites d'agglomération.



(b) Calcul du réseau final : simplification (suppression des liaisons hachurées) d'une superposition de plusieurs réseaux planaires obtenus en fonction des relations de hiérarchie entre les sites.



(c) Calcul des trajectoires selon des contraintes liées aux coûts de l'aménagement des routes.



(d) Application de l'ensemble des méthodes proposées au chapitre 4 sur un modèle numérique terrain décrivant la Corse et le nord de la Sardaigne.

FIG. 5 – *Urbanisation, vue du dessus : (a), (b) puis (c) d'un relief spéculatif composé de cylindres infiniment hauts et représentant des zones inhabitables ; (d) de l'île de la Corse.*

## Organisation du document

Le *chapitre 1* est une description de nos expérimentations d'implémentation des méthodes existantes de déplacement des milieux et des conclusions que nous en avons tirées. Nous mettons en évidence le problème principal des méthodes étudiées : une limitation extrême des contraintes imposables, en particulier sur les altitudes. Nous proposons donc un nouveau modèle de terrain fractal avec une seule rivière : ce modèle introduit une première forme de contraintes. L'ensemble des expérimentations menées, nous permet, au chapitre 2, de proposer des méthodes tenant compte des contraintes arbitraires.

Le *chapitre 2* entame l'exposé de travaux complètement personnels. Tous d'abord, nous présentons une nouvelle méthode fractale de génération de terrain résolvant les difficultés rencontrées au chapitre 1. Notre solution reste toutefois améliorable : les ruptures d'altitudes sont trop abruptes, la génération des jointures entre les altitudes contraintes reste trop primitive. C'est à cette occasion que nous avons introduit une analogie formelle avec l'écoulement des rivières, et nous avons décidé de *faire couler des rivières entre les contraintes*. Nous obtenons alors une jointure satisfaisante résolvant complètement le problème posé au premier chapitre.

Le *chapitre 3* s'attaque ensuite à la génération non-locale de surfaces : des extrema abrupts dans des terrains naturels sont possibles mais demeurent une exception (due à des localisations inhabituellement denses de matière très dure). Comment obtenir des terrains reproduisant la situation de relative continuité obtenue par les effets progressifs de l'érosion fluviale et météorologique ? Nous proposons un nouvel algorithme fondé sur la caractéristique de faire remonter les contraintes dans l'arbre des sous surfaces et de générer ainsi les points intermédiaires permettant le rapprochement avec l'action progressive de l'érosion sur un terrain réel.

Au *chapitre 4*, nous proposons une approche que nous pensons inédite à la problématique de l'urbanisation d'un terrain, urbanisation qui sera planifiée de manière complètement automatique. Nos solutions s'appliquent aux problèmes posés à l'échelle régionale et aux échelles supérieures. La disposition et les dimensions des composantes des unités urbaines, appelées sites d'agglomération, sont calculées selon un modèle statistique et pour une carte de densité générée en fonction des caractéristiques du relief. Le réseau interurbain est établi et prend en considération la hiérarchisation des sites d'agglomération. Nous générons les trajectoires des routes en utilisant une méthode

stochastique basée sur les courbes de Bézier : des contraintes sont imposées de manière à minimiser le coût de l'aménagement des routes produites.

*En conclusion*, nous développons le champ d'applications de nos méthodes. Nous présentons des rendus réalistes des modèles de terrain générés sans aucune donnée initiale, la méthode des écoulement ou un L-Système génèrent le modèle-squelette, ou alors contraint par un croquis dessiné par l'utilisateur. Nous proposons alors des extensions au champ connexe de la synthèse de textures et nous réalisons un coloriage du squelette du précédent L-Système puis un effet de relief de type crépi à partir du précédent croquis. Nous présentons également nos résultats dans le domaine de la reconstruction de surface à partir de données partielles. Notre algorithme fractal est comparé à une méthode rapide utilisée dans le domaine de l'approximation de surface à partir de données satellites sous-échantillonnées : dans les mêmes conditions, notre méthode est 17000% plus rapide. Nous réalisons également une nouvelle expérimentation à partir d'une squelettisation de la donnée ; une détection de creux et de crêtes est préalable à la reconstruction de la surface. La même opération est appliquée à des photographies couleurs afin d'obtenir un rendu non-photoréaliste. Nous terminons par un modèle d'urbanisation automatique calculé pour un terrain généré.

# Chapitre 1

## La modélisation de terrain

Ce premier chapitre présente le domaine de la modélisation automatique de terrain. Les méthodes de *déplacement des milieux* utilisées comme point de départ de nos travaux y sont décrites en détail. Nous réalisons leurs implémentations et définissons une structure de données ainsi qu'une problématique commune. Celle-ci prend la forme de contraintes arbitraires d'altitudes imposées au modèle de terrain généré ; ce modèle doit satisfaire la contrainte sans produire d'artefacts : des ruptures sont générées par les méthodes standard de déplacement des milieux (cf. section 1.1.2). Par ailleurs, une même problématique nous permet de disposer d'une vue d'ensemble décrivant l'évolution de nos travaux présentés dans les chapitres 2 et 3. De plus, en section 1.2, nous présentons une nouvelle méthode permettant de gérer une contrainte non arbitraire prenant la forme d'une rivière fractale. Ici, la contrainte est générée en même temps que le modèle fractal de terrain. Enfin, nous posons le contexte général en présentant la discipline, ses applications et classifiant les diverses approches existantes.

### 1.1 Nos méthodes de déplacement des milieux

Les méthodes de déplacement des milieux ont été introduites, en 1982, par Fournier *et al.* dans [43] puis développées dans [73, 88, 89, 115, 5]. Ces méthodes agissent par subdivisions récursives d'un intervalle (d'une surface ou d'un volume) de manière à tendre vers une courbe fractale. Chaque récursion ajoute du détail à une courbe en subdivisant les morceaux qui la composent. Ainsi, dans un espace unidimensionnel, une approximation de mouvement brownien fractionnaire  $fBm$  [69] peut être obtenue en effectuant un déplacement des milieux proportionnel à la taille de l'intervalle traité.

Dans le cas d'une surface, il existe plusieurs approximations du  $fBm$  utili-

sant les méthodes de déplacement des milieux. Ces approximations prennent souvent la forme d'interpolations composées auxquelles sont ajoutés des déplacements aléatoires relatifs au niveau de récursion. Deux approximations, la subdivision en *Triangle-Edge* [73] et la subdivision en *Diamond-Square* [73], sont proposées par Fournier *et al.* dans [43]. Elles sont nommées et leur comportement est détaillé par G. Miller dans [73] (cf. les schémas (a) et (b) de la figure 1.1). Dans ce dernier, une troisième approche, la subdivision en *Square-Square* (cf. le schéma figure 1.1(c)), est proposée. Cette méthode utilise une interpolation non-linéaire et produit des surfaces de classe  $\mathcal{C}^1$ .

### 1.1.1 Implémentation des déplacements des milieux

Choisissons d'uniformiser la structure de données sous-jacente à un *Modèle Numérique de Terrain* (MNT) ainsi que le paramétrage des différentes méthodes de génération de MNT (celles citées précédemment et celles à venir). Ces dernières pourront être comparées selon les mêmes contraintes et paramétrages en commun.

Nous représentons un MNT comme un tableau bidimensionnel comprenant  $L \times H$  cellules, chacune est composée de deux champs : l'état et l'altitude. Nous utiliserons la notation  $\mathcal{T}[y][x]$  pour une cellule en coordonnées  $(x, y)$  où  $x$  et  $y$  sont des entiers tels que  $0 \leq x < L$  et  $0 \leq y < H$ . L'altitude, donnée par  $\mathcal{T}[y][x].a$ , est stockée dans un entier court signé<sup>1</sup> et l'état, donnée par  $\mathcal{T}[y][x].e$ , stocke des informations supplémentaires concernant la nature de la cellule. À ce stade nous n'avons besoin que d'un seul couple de valeurs pour le champ d'état : *connu* ou *inconnu* ; un MNT vide (i.e. où aucune altitude n'est encore spécifiée ou connue) est la donnée ayant toutes ses cellules étiquetées avec un état *inconnu*.

Plus tard, nous utiliserons l'information d'état afin de fixer les contraintes initiales aux modèles de terrain à générer. Hormis cet usage standard, connaître l'état d'une cellule est aussi nécessaire au processus de subdivision en *Triangle-Edge* ou même dans le cas de la subdivision en *Diamond-Square* lorsqu'un MNT n'est pas de dimensions s'exprimant sous la forme  $2^n + 1$  où  $n \in \mathbb{N}$  ou simplement quand  $L \neq H$ . Ainsi dans le cas de la subdivision en *Triangle-Edge*, pour un rectangle initial, nous effectuons des déplacements de milieux sur les bords et au centre puis nous répétons l'opération pour chaque sous-rectangle. Les quatre sous-rectangles étant adjacents et afin d'éviter la production de ruptures, il est important de ne pas calculer différents déplacements pour une même cellule se trouvant le long de ces axes d'adja-

---

1. Les données satellites sont disponible sous différents formats de cartes d'élévations, par exemple: les fichiers `dem` de l'USGS ou les fichiers `hgt` de la SRTM [112]. La plupart utilisent un entier signé de 16 bits pour la représentation de l'altitude.

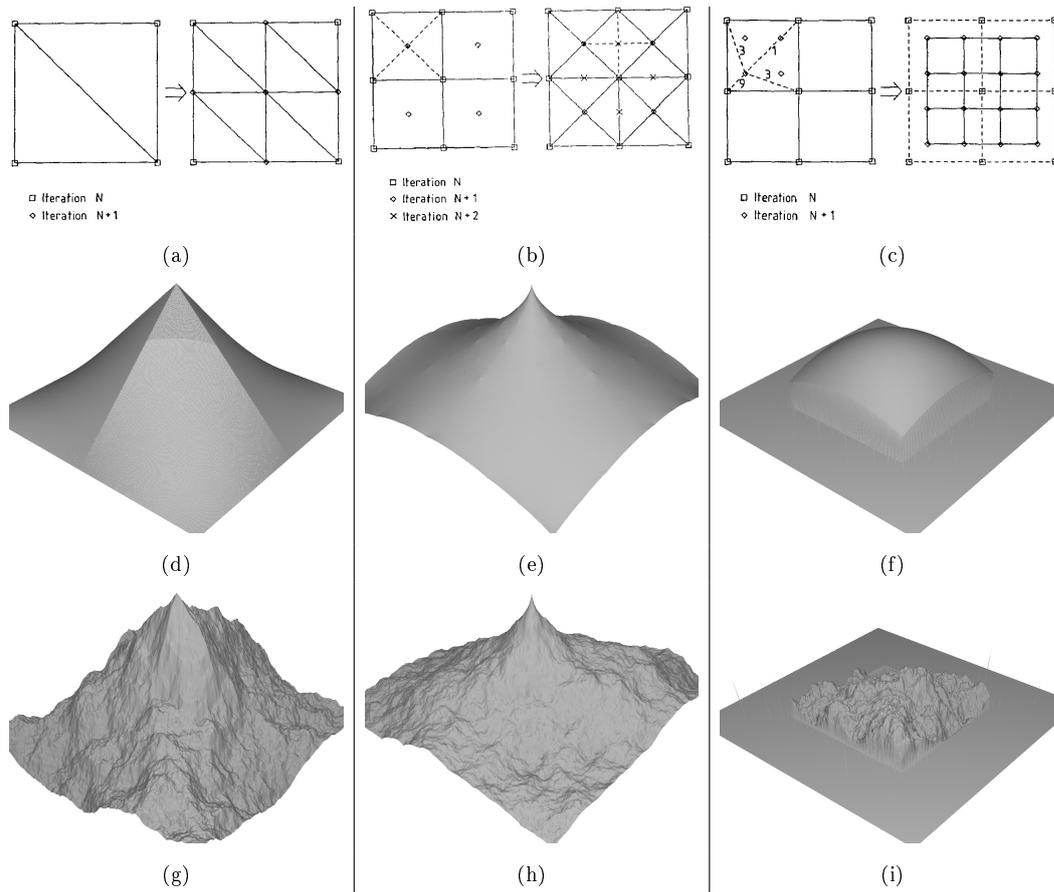


FIG. 1.1 – Méthodes d'interpolation par subdivision successives: (a) La subdivision en Triangle-Edge [73]; (d) l'interpolation obtenue sans déplacement des milieux et (g) le modèle obtenu avec déplacement des milieux; (b) La subdivision en Diamond-Square [73]; (e) l'interpolation obtenue sans déplacement des milieux et (h) le modèle obtenu avec déplacement des milieux; (c) La subdivision en Square-Square [73]; (f) l'interpolation obtenue sans déplacement des milieux et (i) le modèle obtenu avec déplacement des milieux.

cence. Ainsi, nous choisissons<sup>2</sup> de contourner ce problème par l'étiquetage des altitudes déjà calculées en cellules à état *connu* ; le calcul de la valeur intermédiaire (la moyenne) puis le déplacement de l'altitude de la cellule est effectué seulement si cette dernière est étiquetée avec un état *inconnu*. Une particularité de la subdivision en *Square-Square* proposée par G. Miller est de ne pas préserver les altitudes calculées lors des récursions précédentes. Du point de vue de l'interpolation c'est cette même propriété qui garantit la continuité (à l'infini) de la courbe à l'ordre 1. En contre partie elle rend la méthode difficilement contraignable<sup>3</sup>. Enfin, la surface utile générée par cette méthode représente un quart de la surface initiale ce qui en fait, en terme de ressources de calcul, la moins optimale des trois.

Nous fixons des contraintes en posant une altitude et un état *connu* dans certaines cellules d'un MNT vide. Nous posons des contraintes aux coins et au centre d'un MNT vide afin de souligner les différences entre les trois méthodes d'interpolation. Les cellules se trouvant aux coins sont fixées à une basse altitude ( $-2^{14}$ ), la cellule se trouvant au centre du MNT est fixée à l'opposé ( $+2^{14}$ ). La figure figure 1.1 montre, de haut en bas, le schéma récursif de chacune des méthodes de subdivision, l'interpolation sans déplacement des milieux et enfin l'interpolation avec déplacement des milieux. Ces déplacements, notés  $\delta$ , sont pseudo-aléatoires et décroissent, en terme de quantité de mouvement, en fonction du niveau de récursion. Nous les exprimons dans l'ensemble  $[-1,1]^n$ , où  $D$  est la dimension de l'espace topologique, de la manière suivante :

$$\delta = (X + t) \times S \times 2^{-rD\mathcal{H}} \quad (1.1)$$

où  $X$  est un nombre pseudo-aléatoire compris dans l'intervalle  $[-1,1]$ ,  $S = \dim(T[y][x].a) \times s$  et  $s$  est un facteur d'échelle,  $t$  translate l'intervalle de définition du nombre  $X$ ,  $r$  est le niveau de récursion et  $\mathcal{H}$ , une approximation de l'exposant de Hurst<sup>4</sup>, contrôle la dimension fractale de la courbe obtenue.

---

2. Une implémentation itérative du *Triangle-Edge* permettrait d'éviter la multiplicité des déplacements pour une même cellule mais rendrait difficile la parallélisation de la méthode. Sinon, afin de préserver une implémentation récursive et parallélisable, nous pouvons effectuer un même déplacement  $\delta_{xy}$  pour une même cellule  $\mathcal{T}[y][x]$  en utilisant une méthode procédurale. Par exemple une table de hachage retournerait un unique nombre pseudo-aléatoire  $X$  lié à une position  $(x,y)$ .

3. La subdivision en *Square-Square* est une méthode récursive approchant une surface de Bézier. Ainsi nous pouvons poser (à certaines positions clefs) des contraintes qui seront interprétées à la manière de points de contrôle d'un patch de Bézier.

4. Une approximation de la dimension de la surface fractale peut être calculée pour un exposant  $\mathcal{H}$  donné dans l'intervalle  $[0,1]$  et  $s > 0$ . Ainsi cette approximation, notée  $D$ , est donnée par :  $D = 3 - \mathcal{H}$ .

### 1.1.1.1 Temps de calcul

Dans l'ensemble de ce document, toutes nos méthodes de génération de modèles de terrain sont basées ou utilisent, conjointement à une autre méthode, les algorithmes de subdivision présentés plus haut, i.e. la subdivision en *Triangle-Edge* ou la subdivision en *Diamond-Square*, afin d'approximer des surfaces fractales par déplacement des milieux. Ces deux algorithmes ont une complexité d'ordre  $\mathcal{O}(n)$  où  $n = L \times H$  soit les dimensions du MNT ; ils sont très faiblement sensibles aux contraintes initiales (cf. section 1.1.2). Enfin, dans la table 1.1, nous donnons, pour quelques résolutions<sup>5</sup> de MNTs générés, les moyennes des temps de calcul obtenues sur notre architecture de référence détaillée ci-après en section 1.1.1.2 :

Dimensions du MNT	129 × 129	257 × 257	513 × 513	1025 × 1025
Temps de génération en millisecondes	4	13	53	212

TAB. 1.1 – Moyennes des temps de génération de MNTs sans contraintes par notre implémentation des méthodes de déplacement des milieux.

### 1.1.1.2 Architecture de référence

Toutes les méthodes présentées ici et dans les chapitres suivants sont implémentées en C ANSI. L'architecture utilisée pour effectuer toutes les mesures de temps de calcul données dans ce document est équipée d'un processeur *Intel(R) Pentium(R) D CPU 2.80GHz*, de 1024Mo de mémoire vive, d'une carte graphique *ATI Technologies Inc RV530 [RadeonX1600]* et du système d'exploitation *Linux*.

## 1.1.2 Pose de contraintes

Nous étudions les surfaces produites en figure 1.1 (a), (b) et (c) où cinq contraintes sont posées : une altitude haute est imposée au centre du MNT initial et une altitude moyenne est imposée aux quatre coins de ce même MNT. Ces surfaces obtenues sans déplacement de milieux montrent les différences de comportement entre les méthodes de subdivision. Notons que les

---

5. Les résolutions utilisées sont sous la forme  $2^n + 1$  où  $n \in \mathbb{N}$  afin d'unifier les temps de calcul entre les deux méthodes ; cette forme de dimension aide à simplifier les calculs et rend aussi plus performante l'implémentation itérative des méthodes de subdivision. Dans notre cas, la subdivision en *Triangle-Edge* est aussi bien implémentable sous forme itérative que récursive tandis que la subdivision en *Diamond-Square* est difficilement implémentable sous forme récursive à cause de l'interdépendance entre les sous-rectangles pendant l'étape en diamant.

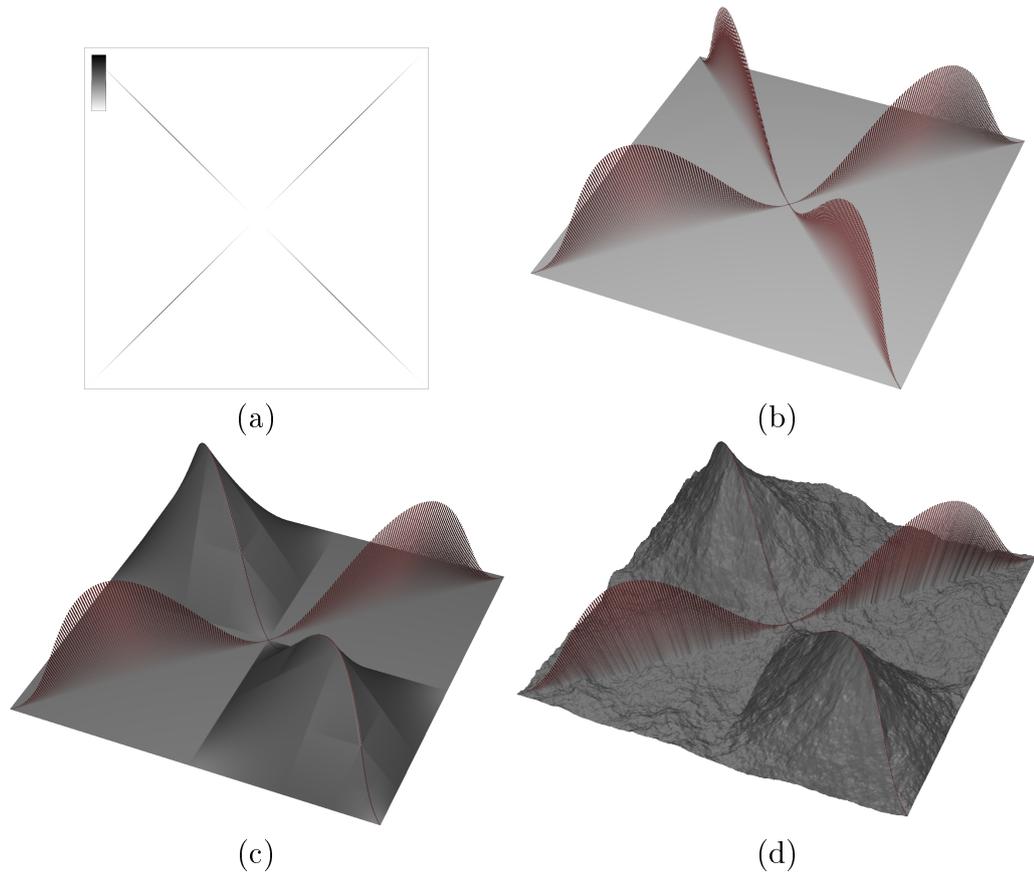


FIG. 1.2 – Première forme de contraintes : fixer certaines altitudes du MNT à générer. (a) / (b) Le MNT-squelette en 2D/3D représentant les contraintes initiales; (c) / (d) le MNT généré (complété) par une interpolation en Triangle-Edge sans / avec déplacement des milieux (la petite languette dans le coin supérieur à gauche de la figure 1.2(a) est un repère d'échelle).

deux premières méthodes satisfont les contraintes initiales mais produisent une courbe de classe  $\mathcal{C}^0$  tandis que la seconde méthode ne fait qu’approcher la contrainte en produisant une surface continue d’ordre 1. La subdivision en *Triangle-Edge* produit une interpolation facettée, ou avec fractures, tandis que la *Diamond-Square* génère une surface parsemée de pics. Beaucoup de paysages naturels sont composés de fractures et de pics et l’érosion peut éventuellement gommer ces aspérités. D’autre part, l’introduction des déplacements de milieux approxime un  $fBm$  produisant par définition des surfaces non dérivables. Nous souhaitons aboutir à une méthode contraignable, paramétrable et rapide produisant des MNTs visuellement proches des reliefs naturels. Ne pouvant pas contraindre les altitudes produites par une subdivision en *Square-Square*, nous utiliserons seulement les deux premières méthodes dans la suite de nos travaux.

A ce stade, les deux méthodes choisies satisfont des contraintes évidentes ; les cinq cellules initiales étant les premières à être utilisées lors des processus de subdivision (cf. section 3.1). En particulier, les contraintes sont satisfaites quand elles sont posées le long des diagonales. Nous posons donc, en croix, deux diagonales de contraintes de manière à simuler quatre crêtes<sup>6</sup>. Ici le MNT est de dimensions  $L = H = 2^9 + 1$ . Les cellules de la contre-diagonale sont décalées d’une unité à droite afin de mettre en évidence la particularité des diagonales. Nous ajoutons un nouvel état noté *crete* afin de différencier, après la génération, les cellules contraintes des autres cellules à état *connu*. La figure 1.2(a) montre<sup>7</sup> le MNT initial utilisé comme ensemble de contraintes ; nous utiliserons aussi le terme de *MNT-squelette*. La figure 1.2(b) montre une vue 3d exprimant les contraintes initiales, les cellules à état *crete*, en rouge foncé et les cellules à état *inconnu* en gris clair. Le résultat de l’interpolation par une subdivision en *Triangle-Edge* sans déplacement des milieux ( $s = 0$  ou  $H \gg 1$ ) est donné figure 1.2(c). Enfin la figure 1.2(d) montre une vue 3d du modèle obtenu avec déplacement des milieux où  $\mathcal{H} = 0.5$ ,  $s = 0.4$  et  $t = 0$ . Notons que la courbe garde un aspect continu le long de la diagonale ; lors du processus de subdivision les altitudes imposées sont utilisées pour les calculs

---

6. Chaque diagonale de contraintes est décrite dans le repère orthonormé centré au milieu du MNT. Les altitudes sont données par une sinusoïdale d’amplitude maximum  $2^{15}$ , de phase nulle et de période  $\frac{2}{\sqrt{L^2+H^2}}$ .

7. Les altitudes étant exprimées par des entiers courts signés, nous donnons une représentation  $2D$  en niveaux de gris 16 bits en translatant les valeurs dans le domaine des non-signés et en affectant une valeur nulle aux cellules à état *inconnu*. Les cartes sont ainsi sauvegardées en *png* (*Portable Network Graphics, Iso-15948*). Enfin, en raison de l’impression papier, nous utilisons le négatif des MNTs produites. Par ailleurs, ces MNTs peuvent être sauvegardées dans un format propre à la structure sous-jacente (nous gardons ainsi toutes les informations concernant la nature des cellules) ou exportées au format *USGS-dem*, *SRTM-hgt* ou *Terragen-ter*.

d'interpolation. Au contraire, le voisinage de la contre-diagonale décalée ne prend en compte aucune des contraintes imposées. Ce simple décalage fait que l'altitude contrainte n'est jamais prise en compte au cours du processus récursif de subdivision ; ces valeurs n'interviennent qu'en fin de récursion. Ainsi des ruptures se créent quand les contraintes interviennent tardivement dans le processus de subdivision. Elles sont potentiellement proportionnelles à la profondeur à laquelle apparaît la contrainte. Dans le cas précis de cette contre-diagonale décalée, la rupture est maximale : la contrainte n'est jamais prise en compte lors du calcul d'interpolation.

Nous proposeront, dans la suite de nos travaux, plusieurs solutions permettant de remédier à ce problème de rupture.

## 1.2 Première approche avec génération de rivières

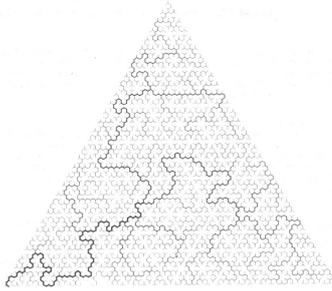
Notre première approche consiste à supprimer les problèmes de ruptures liées aux contraintes en intégrant ces dernières dans le processus de subdivision. Les contraintes sont fixées par une courbe fractale sans boucle décrivant le cours d'une rivière. Cette méthode est initialement proposée par P. Prusinkiewicz et M. Hammel dans [98] en 1993 où un problème de ruptures<sup>8</sup> restait ouvert (cf. la figure 1.3, l'image de droite). Aussi, le choix d'une structure en triangle pour la représentation du terrain limitait dans le choix des méthodes de subdivision à la seule subdivision en *Triangle-Edge*. Ici, nous présentons notre contribution à l'amélioration de la méthode en élargissant ce choix à la subdivision en *Diamond-Square* mais aussi en modifiant le calcul des interpolations afin de réduire les ruptures produites.

### 1.2.1 Le collier de Mandelbrot

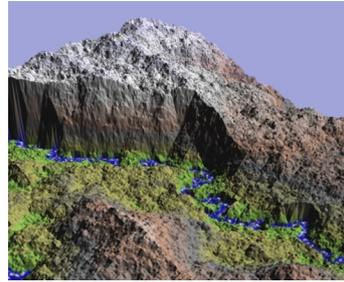
En 1978, B. Mandelbrot introduit dans une note parue dans les Comptes Rendus de l'Académie des Sciences de Paris [67] une variante des *promenades au hasard sans boucle* appelée les *cordonnets discrets et fractals*. Il définit ces objets comme des polygones aléatoires décrits sur un réseau régulier. Un chemin discret est détaillé lors des subdivisions successives du maillage ; lors de cette construction, les probabilités de visiter un sommet voisin non encore visité sont égales. La méthode est donnée dans le cas du triangle équilatéral,

---

8. L'expression utilisée par P. Prusinkiewicz et M. Hammel pour décrire le phénomène de rupture est *vallées asymétriques*.



Un *cordonnet fractal* ou le collier de Mandelbrot par B. Mandelbrot [67].



Un paysage fractal avec rivière par P. Prusinkiewicz et M. Hammel [98].

FIG. 1.3 – Une courbe fractale sans boucle pour créer des modèles de rivières.

le carré et le cube<sup>9</sup>. Elle est détaillée dans le cas du triangle équilatéral ; le schéma gauche de la figure 1.3 montre un réseau fluvial construit selon cette méthode. Jacques Peyrière approfondie l'étude de ces objets fractals qu'il nomme *le collier de Mandelbrot* dans [94].

### 1.2.2 Modèle fractal de terrain avec rivière

Dans [98], Prusinkiewicz et Hammel proposent une méthode générant, lors d'un unique processus, un modèle fractal de terrain avec une rivière. La méthode est basée sur le déplacement des milieux en *Triangle-Edge* et produit simultanément un cordonnet discret fractal, la figure 1.3 à droite montre un des modèles obtenus.

Ici, les auteurs utilisent une propriété commune aux deux méthodes, la *Triangle-Edge* et le cordonnet fractal ; ces deux méthodes étant assimilées à une grammaire générative dont les règles de production sont sensibles au contexte. A une étape  $n$ , un triangle  $T_n^i$ , avec  $i \in \mathbb{N}$  et  $0 < i < 4^n$ , est subdivisé en quatre sous triangles. Les triangles  $T_n^i$  étiquetés comme passage de rivière sont constitués d'une arête *entrée*, d'une arête *sortie* et d'une arête *neutre*. Lors de la subdivision, une arête *neutre* produit un couple d'arêtes *neutres* ; dans le cas d'une arête *entrée/sortie* le couple d'arêtes résultant sera composé d'une arête *entrée/sortie* et d'une arête *neutre*. Selon la composition nous obtenons différentes combinaisons pour le triplé d'arêtes internes de manière à relier l'*entrée* à la *sortie*. Par ailleurs, les altitudes des milieux des segments étiquetés *neutre* se trouvent déplacées selon des paramètres classiques de la méthode *Triangle-Edge* (cf. section 1.1) alors que les altitudes

9. Une variante des cordonnets dans le cube consiste à générer des surfaces *aléatoires* lors des subdivisions récursives du cube en *octree*. Ces surfaces, appelées surfaces chiffonnées, pourraient être contraignables et représenteraient une approche intéressante pour la génération de modèles fractals de terrain ; cette piste reste, à notre avis, à explorer.

des milieux des segments étiquetés comme *entrée* ou *sortie* seront toujours déplacées négativement ; ce déplacement vaut  $\sum_{i=0}^n d_i$  où les  $d_i$  sont les déplacements négatifs des milieux successifs (les prédécesseurs plus le milieu en cours de déplacement) amenant à ce sommet ; arrivé au plus bas niveau de récursion, l'altitude de ces sommets est directement positionnée à 0. Un problème se pose au niveau des adjacences entre les sous triangles quand les subdivisions sont considérées indépendantes les unes des autres. Concernant les altitudes des sommets, le problème peut être résolu en utilisant le système d'étiquetage des cellules proposé en section 1.1.1 mais cette solution est insuffisante dans le cas de l'étiquetage des arêtes. La solution proposée dans [98] utilise deux tables de hachage respectivement pour stocker les déplacements pseudo-aléatoires des sommets et étiqueter les arêtes *neutre*, *entrée*, *sortie* ou *entrée/sortie* ; les arêtes internes au triangle initial étant étiquetées *entrée* et *sortie*. Afin de réaliser notre modèle, nous utiliserons la structure présentée dans la section 1.1.1 couplée à un arbre binaire.

### 1.2.3 Notre modèle fractal de terrain avec rivière

Nous proposons une alternative à la méthode présentée dans [98]. Notre modèle est construit sur une base rectangulaire ; cette configuration admet l'intégration de la subdivision en *Diamond-Square* au modèle proposé initialement. Cette subdivision permet, lors de l'étape en diamant, une meilleure prise en compte du voisinage pendant l'interpolation des altitudes se trouvant sur les milieux des bords. Ainsi, dans un premier temps, les ruptures produites par le modèle sont atténuées. Puis, nous apportons des modifications au niveau des interpolations afin de les supprimer. La figure 1.6 illustre, du haut vers le bas et de gauche à droite, l'évolution des résultats sur un même modèle. Ces résultats sont détaillés en fin de section.

#### 1.2.3.1 Découpe en rectangle

Nous décrivons la méthode des cordonnets discrets et fractals dans le cas d'un carré<sup>10</sup> ainsi que les détails de l'implémentation réalisée.

---

10. Le cas utile est celui du carré ; le découpage du rectangle est effectué de la même manière (en utilisant les mêmes règles de substitution). Dans un espace discret, ce découpage devient impossible : horizontalement quand le rectangle est allongé et d'épaisseur 1 ou verticalement quand le rectangle est plat et d'épaisseur 1. Dans ces cas la subdivision est plus simple car les combinaisons possibles sont réduites.

### Principes de la méthode

Au premier ordre  $n = 0$ , dans un carré unitaire<sup>11</sup>  $C_0^0$ , deux côtés (arêtes) sont sélectionnés au hasard et respectivement étiquetés comme *entrée* et *sortie* ; les deux côtés restant sont étiquetés *neutre*. Lors de la première subdivision  $n = 1$ ,  $C_0^0$  est découpé en quatre carrés  $C_1^0, C_1^1, C_1^2$  et  $C_1^3$  de côté égal à  $\frac{1}{2}$ . Les côtés étiquetés *neutre* produisent deux côtés de moitié étiquetés *neutre*. Les côtés étiquetés *entrée* ou *sortie* produisent respectivement et si possible *au hasard* une moitié étiquetée *entrée* ou *sortie* ; l'autre est étiquetée *neutre*. Afin de relier l'entrée à la sortie, il reste à étiqueter les quatre arêtes au centre de  $C_0^0$  ; posons temporairement une étiquette *inconnue*, notons donc que chaque carrée  $C_1^i$  contient deux arêtes étiquetées *inconnue* et que ces arêtes sont communes à deux carrés. A partir du carré  $C_1^i$  contenant l'entrée, cherchons la sortie : *a*) soit la sortie est trouvée parmi les trois autres côtés (il suffit de regarder le seul côté ne se trouvant pas à l'intérieur de  $C_0^0$  les autres étant étiquetés *inconnue*) *b*) soit la sortie ne fait pas partie des trois arêtes restantes, dans ce cas nous sélectionnons au hasard un des deux côtés à étiquette *inconnue* (soit une arête interne à  $C_0^0$ ) ; cette arête est commune à  $C_1^i$  et  $C_1^j$ , elle est marquée *sortie* du côté  $C_1^i$  et *entrée* du côté  $C_1^j$ . Nous reprenons la recherche à partir de  $C_1^j$ .

Dans tous les cas nous arrivons à relier l'entrée à la sortie soit en traversant zéro, un, deux ou trois arêtes internes à  $C_0^0$  ; les arêtes internes restant étiquetées *inconnue* passent à *neutre*. A l'étape suivante,  $n = 2$ , nous poursuivons la subdivision en partant indépendamment de  $C_1^0, C_1^1, C_1^2$  et  $C_1^3$ . Nous continuons ainsi jusqu'à atteindre le niveau de récursion souhaité.

La figure 1.4 montre, selon la situation initiale, les différentes combinaisons produites lors de la subdivision : les segments en gras représentent des arêtes dont l'étiquetage est *entrée* ou *sortie* et le chemin en pointillés montre le parcours de l'entrée vers la sortie ; les segments restant étant des arêtes étiquetées *neutre*. L'ensemble des cas possibles peut être obtenu par symétrie et rotation des règles de production de la figure 1.4.

Revenons brièvement sur la question de notre préférence pour le choix *au hasard* effectué au moment de la découpe d'une arête interne étiquetée *entrée* ou *sortie* : la sélection de la moitié recevant la même étiquette n'est pas toujours réalisée au hasard ; ces arêtes étant communes à deux carrés adjacents, le choix n'est effectué qu'une seule fois, c'est à dire lors de la subdivision du premier des deux carrés. Dans le cas du second, l'information

---

11. Une notation du type  $C_p^i$  est une abréviation commode nous permettant à toutes les étapes de la découpe d'un carré en sous-carrés, de désigner la face  $i$  associée à une profondeur de découpe  $p$ .

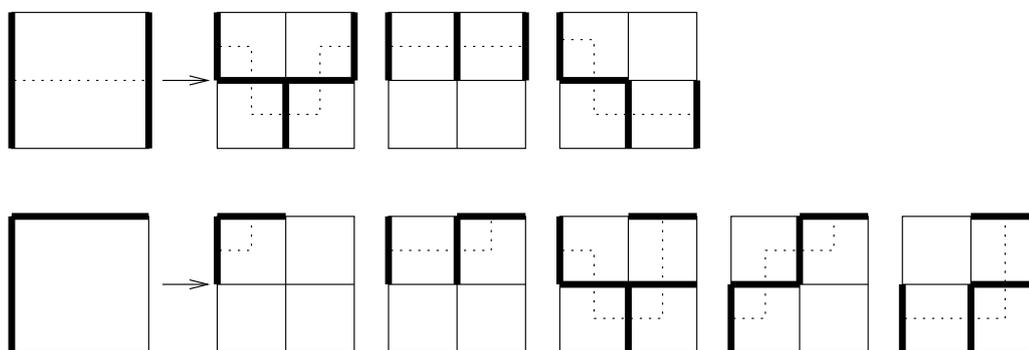


FIG. 1.4 – Règles de production du découpage en rectangle.

est lue et son symétrique (i.e. une *entrée* est considérée comme *sortie* et inversement) est utilisé pour l'étiquetage.

### Notre implémentation

Nous décrivons les spécificités de notre implémentation des cordonnets discrets et fractals. Ainsi, nous utilisons pour la représentation d'un MNT la même structure de données que celle utilisée dans la section 1.1.1. Le tableau de cellules  $\mathcal{T}[y][x]$  où  $0 \leq x \leq L$  et  $0 \leq y \leq H$  est pris comme carré initial  $C_0^0$ ; nous allons le subdiviser récursivement tant que la largeur ou la hauteur des sous rectangles à subdiviser est supérieure à 1. A ce niveau de subdivision (la distance entre les deux extrémités d'un segment est inférieure ou égale à 1), les arêtes  $A_{P_{xy}, P_{x'y'}}$  par lesquelles transite une rivière (soit *entrée* ou *sortie*) sont alors utilisées pour changer l'étiquetage des cellules du MNT. Dans ce cas nous posons :  $\mathcal{T}[y][x].e = \mathcal{T}[y'][x'].e = \text{riviere}$  où  $|x - x'| \leq 1$ ,  $|y - y'| \leq 1$  et *riviere* est un état dénotant le passage d'une rivière en cette cellule.

La structure de données utilisée pour la représentation du MNT est insuffisante pour réaliser une implémentation de la méthode des cordonnets discrets et fractals. Il est nécessaire lors de la subdivision des rectangles de savoir si les arêtes qui le composent sont étiquetées *entrée*, *sortie*, *neutre* ou encore *inconnue*. Pour le faire, la méthode proposée dans [98] est basée sur une fonction de hachage qui, en fonction du segment donné en entrée, renvoie une valeur pseudo-aléatoire stockée dans un tableau et exprimant l'étiquette liée au segment. Cette méthode est rigide et rend difficile la pose de contraintes au niveau des points de passage de la rivière. Ici, nous proposons une solution de stockage alternative basée sur un arbre binaire de recherche. Chaque nœud  $N$  de l'arbre est identifié par un nombre, noté  $N.id$ , lié au seg-

ment qu'il représente. Un nœud contient aussi une étiquette, notée  $N.e$  et liée au segment, ainsi que deux pointeurs  $N.fg$  et  $N.fd$ , vers ses deux fils. Pour des raisons pratiques liées aux architectures actuelles, nous stockons l'identifiant d'un nœud dans un champ 64 bits représentant un entier très long non signé (cf. le type *unsigned long long int* en C ANSI). Aussi, nous limitons les dimensions des MNTs à  $2^{16} - 1$  de manière à pouvoir, pour un segment  $A_{P_{xy}, P_{x'y'}}$  et un nœud  $N$ , stocker une permutation de  $\boxed{x|y|x'|y'}$  dans  $N.id$ . En utilisant une numérotation croissante des octets de poids faible aux octets de poids fort, la permutation est telle que nous stockons dans cet ordre la plus grande abscisse puis la plus grande ordonnée, la plus petite abscisse puis la plus petite ordonnée; l'usage de cette configuration permet qu'une arête  $A_{P_{xy}, P_{x'y'}}$  soit représentée par le même nœud qu'une autre arête  $A_{P_{x'y'}, P_{xy}}$ . Dans le cas présent, l'arbre binaire que nous produisons est construit de manière à trier la donnée stockée dans le champ  $id$  dans l'ordre croissant. Ainsi les fils ne sont pas forcément des fils au sens géométrique du terme (c'est-à-dire que les fils d'un segment seraient les deux moitiés de segment qui le composent).

Concernant l'étiquetage, une simplification est apportée en supprimant la distinction entre l'étiquetage *entrée* et l'étiquetage *sortie*. En effet, lors d'une subdivision en sous rectangles, partant du premier sous rectangle  $R_0$  contenant une *entrée/sortie* nous cherchons une seconde arête marquée *entrée/sortie*; si cette dernière n'existe pas alors une arête d'adjacence avec le sous rectangle  $R_x$ , tel que  $R_x \in \{R_1, R_2, R_3\}$ , dont l'étiquette est *inconnue* servira d'*entrée/sortie*. Le processus est répété sur  $R_x$  jusqu'à avoir, sans opération de changement d'étiquette (i.e. à l'arrivée dans le sous rectangle), deux *entrée/sortie* dans le même sous rectangle.

Ainsi, comme décrit précédemment, au premier ordre, nous partons d'un MNT initial de dimensions  $L \times H$ : l'ensemble des arêtes est stocké dans l'arbre binaire de recherche où elles sont étiquetées au hasard, deux en *entrée/sortie* et deux en *neutre* puis le MNT est subdivisé en quatre sous rectangles. Selon la méthode décrite plus haut, les sous arêtes issues de la subdivision des arêtes initiales sont étiquetées soit *entrée/sortie* soit *neutre*, les arêtes internes sont étiquetées *inconnue* et le chemin est recherché. Une fois trouvé, le restant des arêtes internes est étiqueté *neutre* et l'opération est répétée récursivement sur chacun des sous rectangles tant que la largeur ou la hauteur de ces derniers est supérieure à 1. Au dernier niveau de récursion, nous enregistrons un état *riviere* dans les cellules pour lesquelles un segment de longueur inférieure ou égale à 1 est étiqueté *entrée/sortie*. Le MNT à gauche de la figure 1.5 montre le cordonnet discret fractal ainsi obtenu.

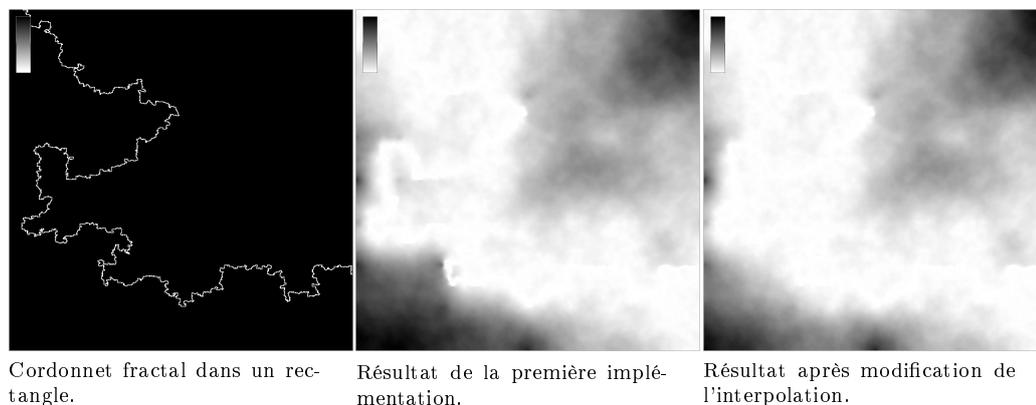


FIG. 1.5 – *MNTs obtenus par notre modèle fractal de terrain avec rivière.*

### 1.2.3.2 Intégration des méthodes de déplacement de milieux

Nous intégrons les méthodes de déplacement des milieux en *Triangle-Edge* et *Diamond-Square* à notre méthode de cordonnets fractals. Cette dernière est utilisée afin d'illustrer les modifications amenant au modèle final. La figure 1.6 montre les surfaces générées en utilisant un même modèle de cordonnet<sup>12</sup>. Pour ce modèle, la cellule du coin haut gauche est positionnée à l'altitude maximale, la cellule du coin bas droit est positionnée à une altitude médiane et les deux coins restant sont positionnés à une altitude basse; toutes ces cellules sont étiquetées *connu*.

#### Sans changements dans l'interpolation

Les méthodes de déplacement des milieux décrites plus haut utilisent la même structure de donnée pour les MNTs. Les déplacements s'intercalent avec le processus de recherche de chemin du modèle de cordonnets. Ainsi, lors de la subdivision des rectangles, nous modifions les étiquettes des arêtes puis nous déplaçons les sommets en fonction de l'étiquetage. Le déplacement donné par l'équation 1.1 est utilisé dans le cas du déplacement des sommets d'une arête étiquetée *neutre*. Une valeur négative  $\delta^-$ , telle que décrite dans [98], est utilisée pour le déplacement des sommets d'une arête étiquetée *entrée/sortie*, elle est telle que  $d^- = \sum_{i=0}^n d_i$  où les  $d_i$  sont les déplacements négatifs des milieux successifs amenant à cette arête. Dans l'équation 1.1, un

12. Un générateur de nombres pseudo-aléatoires est réservé exclusivement à l'usage du générateur de cordonnets. Ainsi les modifications apportées au niveau de la méthode de déplacement des milieux n'affectent pas le modèle de cordonnet et pourront être comparées sur une même base.

$d_i$  est obtenu en posant  $t = -1$ ,  $s = \frac{1}{2}$  et  $r = i$ . Enfin, au plus bas niveau de récursion, les sommets des arêtes étiquetées *entrée/sortie* prennent l'état *riviere*. Ici, cet état ne sert qu'à différencier, au moment du rendu, la rivière du reste du terrain.

Le MNT central de la figure 1.5(b) ainsi que la figure 1.6(a) montrent respectivement le modèle obtenu selon cette méthode ainsi que son rendu tridimensionnel. Nos paramètres sont  $t = 0$ ,  $s = 1$  et  $\mathcal{H} = 0.4565$  et le modèle de  $513 \times 513$  cellules a été généré en 1447 millisecondes sur notre architecture de référence (cf. la section 1.1.1.2).

### Modification de l'interpolation

Certaines régions du modèle rendu en figure 1.6(a) montrent des ruptures brutales aux limites du lit de la rivière ; par exemple la falaise sur le quart haut gauche de la figure. Dans le but de réduire l'intensité de ces ruptures, nous devons améliorer le paramétrage de la vitesse à laquelle le terrain converge vers la rivière, i.e. vers l'altitude nulle. Ainsi nous modifions la façon de calculer les déplacements négatifs des sommets des arêtes étiquetées *entrée/sortie*, i.e.  $d^-$ .

La somme des déplacements négatifs des milieux successifs amenant à une arête étiquetée *entrée/sortie* annule la proportionnelle entre la quantité de déplacement et le niveau de récursion. Ainsi, nous remplaçons les déplacements  $d^-$  par  $\delta^-$  tel que :

$$\delta^- = -X \times S \times 2^{r^f - rn\mathcal{H}} \quad (1.2)$$

nous obtenons approximativement le même modèle de terrain que celui généré précédemment pour  $f = 1$  (sachant que  $n = 2$  et  $\mathcal{H} \approx \frac{1}{2}$ ). En posant  $f = 2$  la quantité de déplacements négatifs augmente de manière quadratique en fonction du niveau de récursion, ainsi nous accélérons la convergence des subdivisions vers le lit de la rivière. La figure 1.6(b) montre un rendu du modèle obtenu ; les paramètres  $t, s$  et  $\mathcal{H}$  sont les mêmes que précédemment. Cette méthode réduit l'intensité des ruptures produites et permet le paramétrage de la vitesse de convergence mais, en contre partie, elle a tendance à générer de larges vallées autour du lit de la rivière pour un paramètre  $f$  trop grand.

La convergence des altitudes vers le lit de la rivière ne devrait pas être liée aux déplacements des milieux (soit aux paramètres  $s$  et  $\mathcal{H}$ ). En posant  $\mathcal{H} > 1$  ou  $s = 0$  nous obtenons une surface d'interpolation entre les coins du MNT, les cellules étiquetées *riviere* sont positionnées en altitude nulle et nous obtenons le modèle rendu en figure 1.6(c). Dans ce cas, afin de faire converger

les altitudes vers le lit de la rivière, il est nécessaire de modifier le calcul d'interpolation pour qu'il prenne en compte cette convergence en fonction du niveau de subdivision. Ainsi, pour une échelle et pour une méthode de subdivision données, le calcul de l'altitude en une cellule  $\mathcal{T}[y][x]$  du MNT étiquetée *inconnu* dépend de la moyenne, généralement pondérée selon la distance, des altitudes d'une combinaison de cellules voisines. La moyenne, notée  $m_{xy}$ , est utilisée dans le calcul de l'altitude  $alt_{xy} = m_{xy} + \delta_{xy}$  du sommet  $(x,y)$  (i.e.  $\mathcal{T}[y][x].a = alt_{xy}$ ). Cette moyenne doit être pondérée dans le cas des sommets des arêtes étiquetées *entrée/sortie*; cette pondération est telle que l'altitude converge vers 0 en fonction du niveau de récursion  $r$ .

En synthétisant l'ensemble de ces données nous obtenons lors de l'étiquetage des arêtes et le déplacement de milieux dont l'état est *inconnu* :

$$alt_{xy} = \begin{cases} m_{xy} + \delta_{xy} & \text{quand le sommet } (x,y) \text{ appartient à une arête} \\ & \text{étiquetée } \textit{neutre}. \\ p \times m_{xy} + \delta_{xy}^- & \text{quand le sommet } (x,y) \text{ appartient à une arête} \\ & \text{étiquetée } \textit{entrée/sortie}. \end{cases} \quad (1.3)$$

avec :

$$p = \left(1 - \frac{r}{r_{max}}\right)^I$$

où  $r$  est le niveau de récursion,  $r_{max}$  le plus bas niveau de récursion<sup>13</sup> et  $I \geq 1$  permet de modifier la vitesse de convergence lors de l'interpolation.

En introduisant ces changements dans le calcul de l'interpolation nous obtenons les différents résultats de la figure 1.6(d),(e) et (f). Ainsi la figure 1.6(d) montre le modèle obtenu sans déplacement des milieux (en posant  $\mathcal{H} > 1$  ou  $s = 0$ ); les altitudes convergent vers le lit de la rivière et la vitesse de convergence est gérée par le paramètre  $I$  (pour ce modèle  $I = 16$ ). Notons l'apparition de piques propres à la méthode de subdivision en *Diamond-Square*. Ces piques n'apparaissent plus lors de l'introduction des déplacements; le modèle de la figure 1.6(e) est obtenu pour  $\mathcal{H} = 0.4565$ ,  $s = 1$ ,  $t = 0$  et  $f = 1$ . Enfin, pour obtenir le MNT à droite de la figure 1.5 soit le modèle 3D de la figure 1.6(f), nous utilisons une puissance légèrement supérieure, soit  $f = 1.2$ , pour augmenter la variation de vitesse dans les déplacements négatifs. Ce dernier paramétrage produit un modèle ne souffrant d'aucune rupture tout en limitant la largeur des vallées.

### 1.2.3.3 Les limitations

Notre méthode améliore le modèle proposé par P. Prusinkiewicz et M. Hammel dans [98]. Ainsi, nous avons un modèle permettant d'éviter la pro-

13. Le plus bas niveau de récursion  $r_{max}$  est calculé en fonction des dimensions du MNT. Ainsi, pour un MNT de dimensions  $L \times H$ ,  $r_{max} = \lceil \log_2 N \rceil$  avec  $N = \max(L,H)$ .

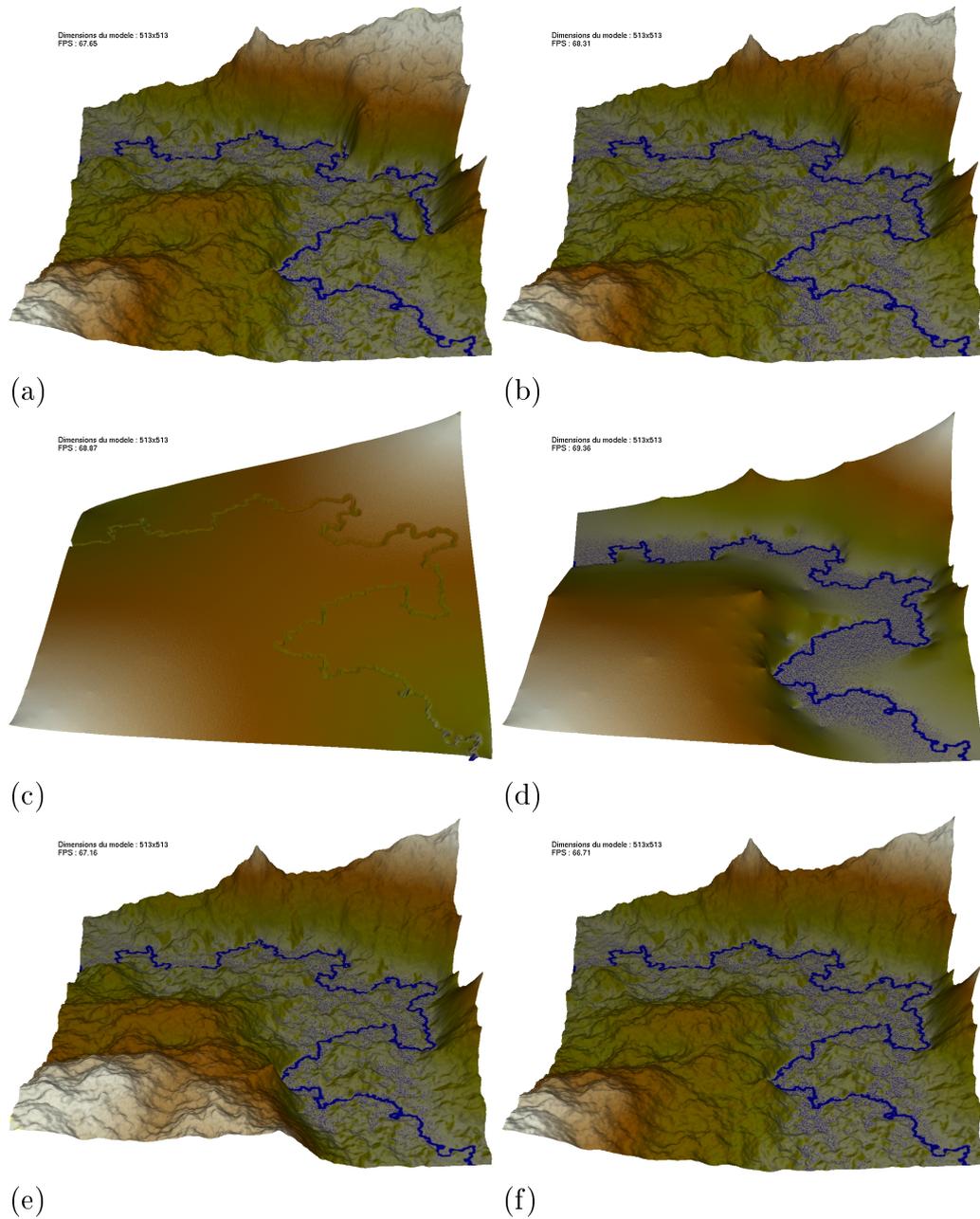


FIG. 1.6 – Notre modèle générateur de terrains fractals avec rivière: illustration des modifications amenant au modèle final de terrain pour un même modèle de cordonnet fractal.

duction de ruptures aux limites de la rivière. Par ailleurs, notre implémentation des cordonnets fractals est contraignable. Nous pourrions fixer, à l'avance, les points de passage du cordonnet ; fixer un point de passage revient à poser une étiquette *entrée/sortie* sur le segment de longueur 1 représentant un passage obligé pour le cordonnet ainsi que sur tous les segments menant, par subdivisions successives, à ce dernier. À l'inverse de la table de hachage proposée dans le premier modèle, l'arbre binaire introduit dans notre modèle permettrait la prise en compte de ces contraintes.

Néanmoins, notre modèle souffre toujours de certaines limitations déjà rencontrées dans [98]. Ainsi, nous générons seulement un modèle décrivant une seule rivière ; nous avons tenté de générer plusieurs rivières en utilisant le modèle de réseau fluvial décrit dans [67] ou sous forme de couches pouvant éventuellement se rejoindre sur les cellules d'intersection. La difficulté rencontrée réside dans les altitudes des torrents et des petits cours d'eau ; nous ne pouvons connaître à l'avance le sens menant vers le cours d'eau principal et ainsi contraindre localement la pente du terrain afin d'obtenir un résultat consistant. Par ailleurs, ramener tous les torrents et cours d'eau à une altitude proche de zéro ne produirait finalement que des vallées et serait contradictoire avec la définition même d'un torrent.

### 1.3 Qu'est-ce qu'un modèle de scène naturelle ?

La modélisation de scènes naturelles fait partie, depuis plus de quatre décennies, des sujets les plus abordés en informatique graphique [36]. Les modèles détaillés et réalistes de terrain ont une large part dans la description de ces scènes. L'acronyme MNT, pour modèle numérique de terrain, est communément utilisé pour décrire le format digital dans lequel est représenté le terrain. Nous proposons de définir, dans sa forme plus générale, la modélisation automatique de terrain, aussi appelée la génération automatique de modèles numériques de terrain, comme la discipline dont la fonction est d'analyser, de restaurer ou de générer des modèles numériques de surfaces rugueuses, érodées, produits d'une évolution géologique et météorologique.

Les MNTs trouvent des applications dans des domaines tels que la réalité virtuelle [1, 10], le militaire [87, 133, 143], l'exploration spatiale [104, 82, 129, 11], les systèmes de géolocalisation [46, 95], la planification urbaine [86, 116], la géologie [14, 13], le cinéma et les jeux vidéo<sup>14</sup> (cf. l'usage des suites commerciales comme : *Terragen*<sup>TM</sup> [124, 10], *Bryce* [21] ou *MojoWorld* [74, 36, 41]). Ces applications sont directement liées au problème sous-jacent de la visua-

---

14. La fonction *noise* (cf. section 1.3.1.1) est décrite dans les spécifications de l'*OpenGL Shading Language* [100].

lisation des données en question. Des solutions sont proposées en fonction de l'importance accordée aux trois aspects suivants : le degré de réalisme, le temps réel et la quantité et le mode de transfert des données traitées. La structure particulière d'un MNT (i.e. généralement une surface représentant un maillage régulier) permet d'optimiser les calculs réalisés par des techniques d'illumination globale telles que les algorithmes de lancer de rayon. Des textures procédurales sont utilisées pour améliorer le réalisme : les propriétés de matière sont calculées en fonction du relief (cf. *MojoWorld: Building Procedural Planets* dans [41]). Par ailleurs, des techniques spécifiques pour le rendu temps réel de MNTs sont développées [40, 54, 63] ; des améliorations de la performance, du niveau de détail et du réalisme du rendu (par exemple par calcul dynamique de textures procédurales) sont réalisées à l'aide d'optimisations spécifiques aux nouvelles générations de cartes graphiques [65, 32, 28, 85]. De plus, la méthode proposée dans [65], le *geometry clipmap*, permet de réaliser, en flux continu, le transfert des données compressées du relief entre les unités de stockage (i.e. la mémoire centrale ou le disque) et les unités d'affichage (i.e. la mémoire graphique) ; une autre approche par flux continu est appliquée à des configurations matérielles variables [96] où le transfert et la qualité de l'affichage sont adaptés en fonction de l'architecture utilisée pour la visualisation, des tests sont réalisés sur *PocketPC* ; enfin, l'approche présentée dans [15] propose une exploration temps réel de reliefs terrestres texturés en haute définition, les auteurs utilisent une architecture basée client-serveur appliquée dans le cas de réseaux de type ADSL.

Ces MNTs peuvent être totalement générés par un processus stochastique, recalculés à partir de données partielles ou approximatives (par exemple, les données manuscrites telles que les cartes de lignes de niveaux) ou simplement chargés et éventuellement retravaillés à partir d'une donnée complète numérisée à un degré de détail donné. Aujourd'hui, les satellites<sup>15</sup> sont capables de numériser plusieurs aspects de la surface de l'astre autour duquel ils gravitent en giga-octets de données la décrivant. Cette quantité de données disponibles ne réduit en rien l'importance du sujet et offre au contraire plus de perspectives dans l'analyse [134, 123] ou la restauration [118, 3, 130, 95, 7] de la donnée numérique correspondante. Par exemple, la compression de données en est une application directe.

---

15. L'utilisation accessible aux civils des données satellites est connue. Par contre, la situation du militaire ne peut être qu'extrapoler par rapport à celle du civil.

### 1.3.1 Les trois approches existantes

La génération automatique de modèles numériques de terrain est une discipline que nous proposons de classer en trois familles de méthodes :

1. Les méthodes basées exclusivement sur des modèles fractals.
2. Les méthodes basées sur des modèles physiques ; certaines utilisent, en plus de l'approche physique, des processus fractals.
3. Les approches utilisant la dizaine de méthodes de calcul standard pour la génération de surfaces ; ces dernières peuvent avoir des points communs avec les deux familles citées plus haut.

Par ailleurs, indépendamment de la classe de méthodes choisie, des solutions peuvent être apportées aux différentes problématiques de la discipline : l'analyse, la restauration ou la génération.

#### 1.3.1.1 Les méthodes fractales

Dans [66], Benoît Mandelbrot définit la côte bretonne par une courbe fractale<sup>16</sup> générée par un mouvement Brownien fractionnaire, noté  $fBm$  [69] *fractional Brownian motion*. Ce même principe est appliqué, sous différentes formes d'approximations, dans le cas des surfaces afin de générer des modèles fractals de terrain. Les premières approximations, données dans [68, 131, 20], décrivent un mouvement Brownien multidimensionnel ; soit le  $fBm$  est fonction d'une coordonnées  $(x,y)$  sur le plan décrit par le modèle numérique de terrain. Cette approximation est réalisée en considérant la forme spectrale d'un  $fBm$  unidimensionnel (i.e. l'analyse spectrale du mouvement Brownien lié à une variable de temps  $t$ ) et en appliquant une transformée spectrale bidimensionnelle inverse. Cette approche du  $fBm$  multidimensionnel est coûteuse en temps de calcul. Des alternatives sont rapidement proposées et prennent la forme de méthodes de subdivision<sup>17</sup> [23, 43, 73, 88, 89, 115], ou de réécriture [98]. Ces méthodes approximent un  $fBm$  par déplacement des milieux<sup>18</sup>. La méthode proposée dans [98] génère, en un seul processus de réécriture sensible au contexte, un terrain traversé par une rivière. La méthode a été

---

16. Le terme fractal créé, par B. Mandelbrot, à partir de l'adjectif latin *fractus* signifie brisé.

17. Les méthodes de subdivision sont utilisées pour approximer les courbes lisses (au minimum  $C^1$  continues) contrôlées par une suite de points décrivant un polygone arbitraire. Elles ont été introduites par Chaikin en 1974 [25] puis des généralisations au problème des surfaces lisses sont proposées par Catmull & Clarck [24], Doo & Sabin [39] et Loop & DeRose [64].

18. Nous avons détaillé une partie des méthodes de subdivision existantes en section 1.1 ; elles sont utilisées comme élément de base de nos algorithmes fractals présentés dans les chapitres 2 et 3.

développée en section 1.2 et nous avons proposé une amélioration des résultats obtenus par ce type de processus.

Dans [61], J.P. Lewis présente une généralisation des méthodes de déplacement des milieux ; il propose une meilleure intégration du voisinage lors du calcul d'interpolation et introduit la possibilité de modifier les harmoniques de la surface produite. Un sur-échantillonnage est appliqué pour fabriquer une surface fractale haute définition contrainte par les valeurs de luminance d'une image basse définition. Cette forme de contraintes (locales) est particulière ; les valeurs sont disposées de façon régulière sur une grille plus grande. Une première généralisation des contraintes locales est donnée sous forme d'enveloppe d'altitudes imposées dans [58].

D'autres méthodes, dites procédurales<sup>19</sup> [41], utilisent des fonctions de bruit<sup>20</sup> pour produire des surfaces fractales. Le bruit de Perlin<sup>21</sup>, proposé par Ken Perlin [90,92,91], est fréquemment employé dans le domaine de la génération de scènes naturelles. Une approche multi-fractale, proposée dans [41,36], améliore le réalisme des modèles de terrain générés. Dans [30], Cook et DeRose proposent une meilleure définition du bruit où la somme des fréquences n'est pas bornée : ils utilisent une méthode basée sur les ondelettes.

---

19. Le terme procédural pourrait servir à définir la majorité des méthodes citées ou décrites dans ce document ; la modélisation procédurale concerne toute génération ou amplification de modèles numériques. Néanmoins, nous l'utilisons ici dans le contexte très spécifique à la génération de textures solides (i.e. invariantes aux rotations et aux translations) ; il prend le sens *éditeur de flux de pixels* (PSE : *Pixel Stream Editor*) introduit par Ken Perlin dans [90] où le calcul d'une couleur n'est pas directement lié aux contraintes imposées par son voisinage dans la grille de couleurs (par exemple, aujourd'hui ces méthodes s'adaptent facilement à la programmation, dans le *GPU*, de *Fragment Shaders* [100]). Ainsi, une distinction est faite par rapport aux méthodes de déplacement des milieux. Ces dernières amplifient stochastiquement la donnée à l'étape  $n + 1$  (par exemple la profondeur de l'arbre de récursion) en fonction des données de l'étape  $n$  ; soit, dans ce cas, une analogie avec la propriété markovienne est possible.

20. Les fonctions de bruit, ou *Noise Function*, ont été introduites par Ken Perlin en 1985 dans [90] ; le bruit de Perlin est depuis largement diffusé et utilisé pour générer des textures solides. Le mot bruit est emprunté à la discipline du traitement du signal ; une analogie est possible avec le domaine de représentation des couleurs, par exemple : une même densité spectrale de puissance dénote un bruit blanc ; une densité spectrale de puissance inversement proportionnelle au carré de la fréquence est appelée bruit rouge ou bruit brownien.

21. Du point de vue fréquentiel, le bruit de Perlin est obtenu par une somme finie de bruits blancs pris à différentes échelles : l'amplitude est inversement proportionnelle à la fréquence (la fréquence est exprimée en octaves) ; sa décroissance est contrôlée par un paramètre appelé la persistance. D'un point de vue géométrique, pour chaque coordonnée, est calculée une suite d'interpolations dans des maillages de plus en plus fins. Les valeurs interpolées sont pondérées en fonction de la persistance et du niveau d'interpolation.

### 1.3.1.2 Les méthodes physiques

Les approches physiques sont utilisées pour contraindre ou pour façonner, au fil du temps, la surface d'un terrain.

D'une part, nous avons, les méthodes considérées statiques où un modèle physique est utilisé pour contraindre ou déformer une surface en fonction des forces exercées. Dans [57], Kelley et al. proposent de générer, via un processus récursif, un réseau de drainage contraignant les altitudes d'une surface initialement plane. Un bassin versant initial est donné par une ligne, c'est l'affluent principal, séparant le terrain en deux versants. Ce bassin est développé par génération récursive d'affluents ; les sous bassins sont obtenus en soumettant le terrain à une tension proportionnelle à l'affluent. Dans [78], l'auteur propose une approche similaire où le terrain initial est façonné en fonction du flux des cours d'eau. Ces derniers sont initialisés à l'aide d'un  $fBm$  unidimensionnel circulant le long d'une surface plane. D'autres approches visent à contraindre les surfaces fractales obtenues, dans [119, 118] et [3, 130] le terrain est considéré comme une surface sous tension (i.e. soumise aux forces données sous la forme d'altitudes imposées) où l'état d'équilibre est ramené en utilisant un échantillonnage de Gibbs<sup>22</sup>. Ces méthodes permettent la reconstruction de terrains à partir d'échantillons uniformément distribués ou peuvent être utilisées comme outils d'analyse de la rugosité (i.e. la dimension fractale) des surfaces, mais demeurent trop complexes en temps de calcul pour être utilisés de manière interactive. Stachniak et Stuerzlinger proposent une première solution en temps différé [113] allant dans le sens de l'interactivité, leur méthode permet de créer des terrains sous contraintes par déformations mais souffre du même problème de complexité.

D'autre part, des approches dynamiques réalisent des approximations des processus d'érosion ; des modèles physiques sont utilisés pour simuler ces phénomènes. Ainsi, dans [77], Musgrave et al. proposent le premier modèle utilisant des simulations d'érosion thermique et hydraulique pour générer des modèles réalistes de terrain : l'érosion est réalisée à partir d'un modèle fractal basé sur le bruit de Perlin, des écoulements d'eau sont simulés et produisent des déplacements de sédiments, le modèle est enfin modifié par l'action locale de l'érosion thermique produisant des effets d'éboulements. Ce modèle d'érosion est d'abord généralisé par Roudier et al. dans [101] ; Chiba et al. introduisent l'utilisation de champs de vitesse pour simuler l'écoulement des particules d'eau [27] ; les structures de données utilisées sont parfois plus complexes qu'un simple MNT, Beneš et al. utilisent une représentation

---

<sup>22</sup> L'échantillonnage de Gibbs est une méthode itérative produisant une séquence d'échantillons conjointement à une distribution, prédéfinie, de probabilités, i.e. un histogramme.

par couches [14, 13] afin de mieux approximer certains modèles géologiques ; dans [80] l'érosion est appliquée de manière interactive, ainsi la mécanique du fluide réalisant l'érosion est simulée par une approximation de solveur *Navier-Stokes* [114] ; un autre modèle interactif, cette fois basé sur des simulations approximant des eaux peu profondes [56], est proposé dans [12] ; enfin, l'approche présentée dans [71] est basée sur des champs de vitesse et présente la particularité d'être implémentée dans le *GPU*, la méthode est non-interactive mais l'effet de l'érosion est visualisé en temps réel. Ainsi, cette catégorie d'approches est idéale pour réaliser l'évolution d'un terrain au cours du temps mais reste trop complexe, en temps de calcul, pour arriver à un premier résultat réaliste en temps réel.

### 1.3.1.3 Les autres approches

Finalement, nous présentons, par l'intermédiaire des six prochains travaux, les quelques approches produisant, sous contraintes, des modèles de terrain et utilisant d'autres méthodes de calcul, différentes de celles citées plus haut. Ainsi, une approche par réseaux de neurones est utilisée dans [95] afin de réaliser des approximations de surfaces lisses à partir de contraintes données sous la forme de nuages de points ou de lignes de niveau : des modèles de terrain reconstruits à partir de données partielles sont obtenus, ces modèles sont visuellement proches des modèles originaux, la qualité du résultat et la complexité du calcul sont relatives au nombre de contraintes imposées, nous réalisons, en section 5.2.1, pour un exemple de données proposé dans [95] une comparaison entre cette méthode et notre modèle fractal proposé au chapitre 3. Dans [83], les auteurs proposent une approche basée sur les algorithmes génétiques : à partir d'une enveloppe définie par l'utilisateur, un premier processus de sélection raffine le squelette de l'enveloppe, le deuxième processus construit la surface à partir d'un ensemble d'échantillons de modèles de terrain. Nous retrouvons dans [142, 19, 26] l'utilisation d'échantillons de terrains utilisés pour synthétiser de nouveaux modèles : dans [142] un nouvel MNT est synthétisé à l'aide de patches de MNTs existants sélectionnés puis appliqués au voisinage d'un croquis donné comme forme à suivre ; une approche similaire est proposée dans [26] ; la méthode présentée dans [19] applique les détails, i.e. les hautes fréquences, d'un terrain existant afin de raffiner le modèle de terrain synthétisé. Dans [33], la surface d'un terrain est analysée puis décomposée à partir d'une représentation en maillage irrégulier donné au format TIN, *Triangulated Irregular Network* : un modèle de terrain multi-résolution est obtenu.

Nous avons résolu, dans le cadre de notre étude, le problème de déduction de contraintes par un processus dynamique dans lequel contraintes et terrain s'engendrent et se modifient mutuellement en va et vient au cours du temps.

## Chapitre 2

# Notre modèle fractal à contraintes progressives : vers l'érosion

Nous proposons, dans ce chapitre, une approche fractale basée sur les méthodes de subdivision présentées dans le chapitre 1. Cette méthode génère des modèles de terrain à partir de contraintes locales<sup>1</sup> ou bien elles-mêmes générées ou bien, dans une certaine limite<sup>2</sup>, décrites par l'utilisateur. Ces contraintes prennent la forme de MNT-squelettes où certaines altitudes de cellules sont imposées. Nous présentons une méthode rapide permettant de produire ces modèles-squelettes : des lignes de crête sont générées par des particules simulant, le long du MNT, un mouvement brownien unidimensionnel ; les bassins versants<sup>3</sup> correspondant aux lignes de crêtes sont modélisés ; enfin, nous calculons l'hydrologie de la surface obtenue, i.e. le réseau fluvial. A partir de ces derniers résultats, nous montrons le premier modèle réaliste de terrain érodé obtenu en associant ce générateur de MNT-squelettes avec la méthode fractale proposée.

### 2.1 Fenêtrage croissant

Nous présentons notre première méthode fractale permettant de générer des modèles de terrain où certaines positions sont contraintes en altitude ;

---

1. Nous définissons la notion de contrainte locale comme l'ensemble des positions, dans le plan décrit par le MNT, dont l'altitude est imposée. La structure de données utilisée pour représenter ce type de contraintes a été exposée dans la section 1.1.1 et 1.1.2.

2. Nous illustrons, dans la section 2.1.3, en utilisant un exemple de contraintes serrées, les limitations (i.e. la méthode produit des ruptures) de la méthode.

3. Un *bassin versant* est la région délimitée, drainée par un cours d'eau et ses tributaires, dont elle constitue l'aire d'alimentation [29]. Les termes bassin fluvial ou bassin hydrographique sont des synonymes.

ces contraintes sont données en entrée du modèle sous la forme d'un MNT-squelette. Les contraintes gérées ici doivent être souples (i.e. elles sont progressives) : nous montrerons un premier exemple de contraintes non gérées par un tel modèle fractal.

### 2.1.1 Une génération orientée contraintes

L'idée générale de notre seconde approche est de construire de proche en proche une surface satisfaisant un ensemble initial de contraintes données sous forme d'altitudes fixes. La méthode agit par dilatation des contraintes données en entrée du modèle. Ces contraintes d'altitude sont sans limitations et peuvent être imposées à n'importe quelle position du MNT initial : les contraintes posées ici sont à prendre au même sens que les contraintes données en section 1.1.2. Ainsi nous utilisons, dans un premier temps, le MNT-squelette proposé en figure 1.2(a) afin de montrer que cette approche offre une solution (cf. le modèle rendu à gauche sur la figure 2.2) au problème de discontinuité présent sur l'interpolation de la contre-diagonale décalée lorsqu'une simple méthode de subdivision est appliquée (cf. figure 1.2(c)). Enfin, nous obtenons le modèle fractal correspondant en introduisant les déplacement des milieux (cf. le modèle rendu à droite sur la figure 2.2).

Ainsi, notre démarche par fenêtrage croissant consiste à extrapoler, pour une échelle donnée, les altitudes des cellules étiquetées *inconnu* se trouvant au voisinage de cellules étiquetées non-*inconnu*. Une interpolation est ensuite réalisée dans chaque quadrillage décrit par la (ou les) cellule étiquetée non-*inconnu* et la (ou les) cellule à l'altitude extrapolée. Ce processus est réalisé pour des échelles croissantes ; notons ici que la notion de voisinage est aussi liée à l'échelle.

La figure 2.1 montre, de gauche à droite et du haut vers le bas, toutes les étapes de la méthode par fenêtrage croissant. Dans cet exemple, la méthode est appliquée au MNT-squelette posé comme contrainte initiale en section 1.1.2 - figure 1.2. Ainsi, à partir du MNT-squelette donné en entrée nous effectuons une dilatation progressive de ce squelette. Le voisinage créé autour des cellules contraintes est formé par un patchwork de petites interpolations locales. Ici nous utilisons nos implémentations des méthodes de subdivision vues en section 1.1.1 : la *Triangle-Edge* ou la *Diamond-Square* (cf. (\*) en fin de sous-section). Les nouvelles altitudes calculées renforcent le squelette initial qui est réinjecté en entrée du processus de dilatation. Généralisée, cette dilatation est appliquée en plusieurs passes chaque passe agissant à une échelle plus grande. Ainsi, de proche en proche, la dimension des patches utilisés croît en fonction de la distance au squelette initial. En pratique nous utilisons une échelle commençant à 1 (i.e. le voisinage sera pris à une dis-

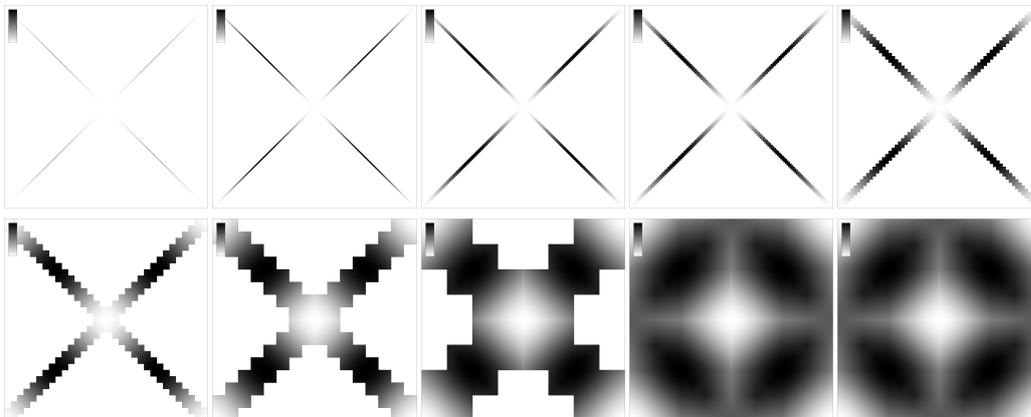


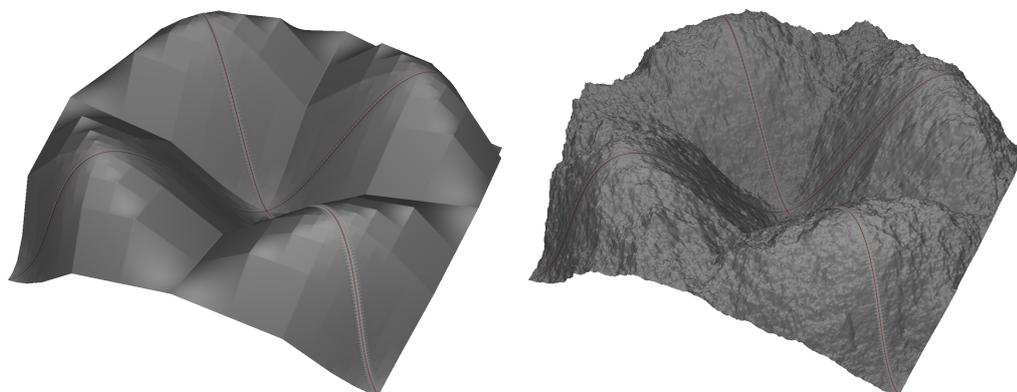
FIG. 2.1 – *Progression de la méthode par fenêtrage croissant appliquée aux contraintes (le MNT-squelette) posées en section 1.1.2.*

tance 1 en 8-connexité) et doublant à chaque étape ; les patches utilisés sont représentés par des carrés de côté égal à l'échelle et cette dernière est une puissance de 2. L'image à gauche sur la figure 2.2 montre le résultat d'une telle interpolation ; nous distinguons légèrement les patches de taille croissante utilisés pour l'interpolation de la surface. Pour obtenir un résultat approximant une surface fractale, nous réalisons un déplacement de milieux dans la phase d'interpolation ainsi que des déplacements proportionnels à l'échelle pour la phase d'extrapolation. L'image à droite sur la figure 2.2 montre une vue 3D du modèle obtenu en utilisant un paramétrage  $\mathcal{H} = 0.5$ ,  $s = 0.4$  et  $t = 0$  lors des calculs de déplacement des milieux. Ces deux modèles ont été générés en 110 millisecondes sur notre architecture de tests (cf. la section 1.1.1.2) ; la méthode de subdivision utilisée ici est la *Triangle-Edge*.

(\*) Notons que l'utilisation des méthodes de subdivision vues en section 1.1.1 est ici particulière. En effet, la subdivision est appliquée qu'à une partie du MNT donné en entrée. Ainsi, nous devons spécifier un sous-rectangle dans lequel la subdivision (l'interpolation et éventuellement le déplacement des milieux) sera effective.

### 2.1.2 Notre algorithme par fenêtrage croissant

Afin d'effectuer un traitement uniforme pour toutes les cellules et à chaque étape  $p$  donnée (i.e. pour une échelle donnée  $pas = 2^p$ ), nous utilisons un second tableau de cellules, noté  $\mathcal{I}_c[N][N]$ , contenant la copie faite à l'étape  $p-1$  du MNT  $\mathcal{T}$  en cours de traitement. A chaque échelle, l'algorithme se compose



Modèle obtenu sans déplacement des milieux.

Modèle obtenu avec déplacement des milieux.

FIG. 2.2 – La méthode par fenêtrage croissant appliquée au modèle squelette proposé en section 1.1.2 (les cellules du squelette initial sont colorées en rouge foncé).

de deux parties bien distinctes. La première partie consiste à extrapoler des altitudes prises à un voisinage proportionnel à l'échelle ; i.e. l'altitude d'une cellule étiquetée *inconnu* est extrapolée (puis étiquetée *connu*) dans  $\mathcal{T}_c$  par rapport à ses voisines étiquetées non-*inconnu* dans le quadrillage d'unité égale à *pas* et prises dans  $\mathcal{T}$ . Une fois ce processus effectué sur tout le MNT pour cette échelle *pas* une copie est faite en sens inverse de  $\mathcal{T}_c$  vers  $\mathcal{T}$ . La seconde partie de l'algorithme consiste à réaliser des interpolations à l'intérieur du quadrillage d'unité *pas* (quand *pas*  $\neq 1$ ) pour les carrés dont les quatre cellules-coins sont étiquetées non-*inconnu*. L'ensemble est répété pour toutes les échelles dans l'ordre croissant.

Dans **Algorithme 1**, nous choisissons de simplifier l'expression de l'algorithme en prenant  $L = H = N = 2^n + 1$  avec  $n \in \mathbb{N}$ . Dans le cas où  $L = 2^l + 1$  et  $H = 2^h + 1$  avec  $(l, h) \in \mathbb{N}^2$  et  $l \neq h$ , il suffit de modifier les limites des variables de boucle  $i$  et  $j$  ainsi que poser une limite pour  $p$  égale à  $\min(l, h)$ , soit un *pas* majoré par  $2^{\min(l, h)}$  (ou  $\min(L, H)$ ). Enfin dans le cas où au moins une des dimensions du MNT n'est pas traduisible sous la forme  $2^n + 1$  alors un processus de déplacement des milieux est appliqué sur la totalité du MNT (c'est-à-dire en spécifiant le rectangle composé des quatre coins du MNT comme rectangle initial à subdiviser) à la suite de l'algorithme par fenêtrage croissant.

---

**Algorithme 1** Déplacement des Milieux par Fenêtrage Croissant.

---

**ENTRÉES:** le triplé  $(s, t, \mathcal{H})$  de l'équation 1.1 ainsi que des contraintes d'altitudes décrites dans un MNT-squelette  $\mathcal{T}$ .

**SORTIES:**  $\mathcal{T}$  complété.

```

pour  $pas \leftarrow 1$  à  $N - 1$  incrément  $pas$  faire
  copier  $\mathcal{T}$  dans  $\mathcal{T}_c$ 
  pour  $i \leftarrow 0$  à  $N - 1$  incrément  $pas$  faire
    pour  $j \leftarrow 0$  à  $N - 1$  incrément  $pas$  faire
       $C_{(i,j,pas)} \leftarrow$  carré de côté  $pas$  et de coin haut-gauche  $(i,j)$ 
       $nb_{ni} \leftarrow 0$  et  $alt \leftarrow 0$ 
      pour tout  $(x,y)$  coin de  $C_{(i,j,pas)}$  faire
        si  $\mathcal{T}[y][x].e \neq \text{inconnu}$  alors
           $nb_{ni} \leftarrow nb_{ni} + 1$  et  $alt \leftarrow alt + \mathcal{T}[y][x].a$ 
        finsi
      fin pour
      si  $nb_{ni} > 0$  alors
        pour tout  $(x,y)$  coin de  $C_{(i,j,pas)}$  faire
          si  $\mathcal{T}[y][x].e = \text{inconnu}$  alors
             $\mathcal{T}[y][x].a = \frac{alt}{nb_{ni}} + pas \times s \times (X+t)$  et  $\mathcal{T}[y][x].e = \text{connu}$ 
          finsi
        fin pour
      finsi
    fin pour
  fin pour
  copier  $\mathcal{T}_c$  dans  $\mathcal{T}$ 
  si  $pas = 1$  alors CONTINUER L'ITÉRATION
  pour  $i \leftarrow 0$  à  $N - 1$  incrément  $pas$  faire
    pour  $j \leftarrow 0$  à  $N - 1$  incrément  $pas$  faire
       $C_{(i,j,pas)} \leftarrow$  carré de côté  $pas$  et de coin haut-gauche  $(i,j)$ 
      si les quatre coins de  $C_{(i,j,pas)}$  sont non-inconnu alors
        faire : déplacement des milieux  $\text{dm}(C_{(i,j,pas)}, s, t, \mathcal{H})$ 
      finsi
    fin pour
  fin pour
fin pour

```

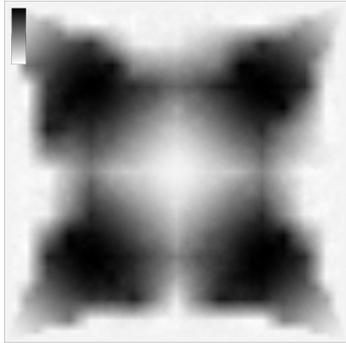
---

### 2.1.3 Un défaut de jointure

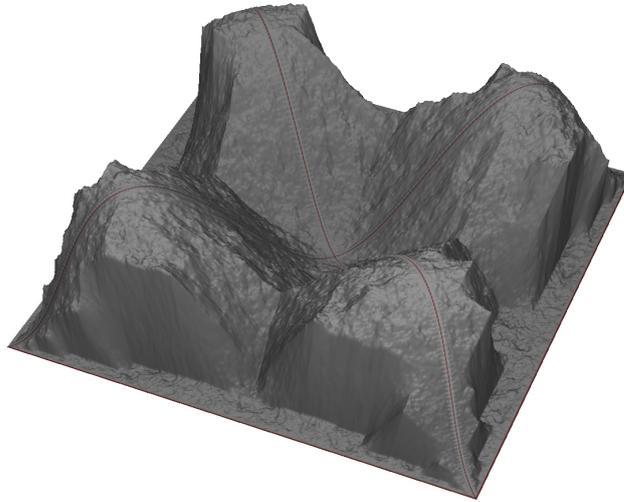
Comme nous pouvons le constater sur la figure 2.2, notre méthode par fenêtrage croissant arrive à produire une surface sans ruptures là où une utilisation classique d'une méthode de subdivision comme la *Triangle-Edge* ou la *Diamond-Square* échoue. Ainsi, l'interpolation est progressive même au niveau de la contre-diagonale décalée. Par contre l'allure de la surface obtenue n'est plus la même. Visuellement, en tenant seulement compte de la diagonale non-décalée, les modèles de la figure 2.2 sont très différents des modèles des figures 1.2(c) et (d). L'usage classique de ces méthodes de subdivision amène à propager des valeurs (des altitudes) se trouvant aux emplacements clés de la subdivision vers l'ensemble du MNT généré. Ainsi, les plus importantes valeurs se trouvant aux coins du MNT initial, une majorité des altitudes interpolées est tirée vers le bas; les coins du MNT initial étant à une altitude nulle. Nous pouvons constater une hiérarchisation **globale** des positions; cette hiérarchisation est sous-jacente à l'ordre dans lequel les subdivisions sont réalisées. Au contraire, la démarche par fenêtrage croissant construit la surface de proche en proche en partant des contraintes initiales et en augmentant d'échelle en échelle la dimension des patches utilisés pour l'interpolation. Ici toutes les contraintes (les altitudes étiquetées non-*inconnu*) ont la même importance et influent seulement sur leur voisinage immédiat (à une échelle donnée). Cette approche, orientée contraintes, est donc une approche **locale**; l'allure de la courbe obtenue renforce cette constatation.

Une approche locale est satisfaisante quand les contraintes fixées ne sont pas concurrentes. Prenons la situation où deux cellules relativement proches en position contiennent des contraintes d'altitudes opposées. L'extrapolation des altitudes des cellules voisines aux deux positions construit, de proche en proche, deux sous-surfaces évoluant l'une vers l'autre à des altitudes diamétralement opposées. La proximité des positions initiales implique l'utilisation d'un patch de petite dimension au moment où une interpolation de jonction est réalisée entre les deux sous-surfaces. Ce patch de petite dimension se trouve donc étiré de bas en haut sur une courte distance produisant ainsi une déchirure (ou rupture) au niveau de la surface finale.

Nous modifions notre MNT-squelette afin d'illustrer cet effet de déchirure. Aux deux diagonales de contraintes initialement proposées, nous ajoutons des contraintes sur toutes les cellules des bords du MNT-squelette; toutes ces cellules sont positionnées à une altitude nulle. La méthode par fenêtrage croissant est utilisée (avec déplacement des milieux) pour compléter le MNT initial ainsi défini. La figure 2.3 met en évidence les ruptures produites entre les crêtes et les bords du MNT. Ce problème peut être contourné si une jonction progressive existait entre ces deux familles de contraintes; les diagonales



MNT obtenu avec notre méthode par fenêtrage croissant.



Vue 3D du MNT obtenu ; les contraintes initiales sont posées sur les deux diagonales ainsi que sur les bords.

FIG. 2.3 – *Défaut de jointure pour le Déplacement des Milieux par Fenêtrage Croissant*: le MNT-squelette proposé en section 1.1.2 est contraint à une altitude nulle aux bords.

représentent les crêtes et les bords représentent les vallées.

Ainsi, une solution est apportée dans la section suivante. En effet, nous pouvons éviter la production de ruptures en ajoutant des contraintes de liaison entre les contraintes concurrentes. Ces contraintes de liaison représentent une autre forme d'interpolation. Dans le MNT-squelette, cette nouvelle forme de contraintes doit avoir une densité à peine suffisante pour pouvoir, d'une part, réaliser progressivement la jonction entre les contraintes opposées et d'autre part, s'intégrer naturellement au paysage.

Dans la perspective de générer des modèles réalistes de terrain, un réseau fluvial représente la solution idéale à ce problème de ruptures.

## 2.2 Réseau de crêtes et hydrologie : la méthode des écoulements

Nous proposons une méthode générant un modèle de terrain composé d'un ensemble de crêtes pour lequel un réseau fluvial est calculé. Ce modèle, réduit aux lignes de crêtes et au réseau fluvial, est utilisé en entrée de la méthode par fenêtrage croissant afin de générer un modèle réaliste de terrain.

### 2.2.1 Un modèle de crêtes

Une ligne de crête est un ensemble de points hauts séparants deux versants opposés. Ces lignes peuvent être extraites à partir données satellites (cf. section 5.2.2), spécifiées par l'utilisateur (cf. section 3.1.2) ou encore générées à l'aide d'une méthode stochastique. Cette troisième solution est proposée ci-dessous.

Nous présentons une méthode rapide pour créer des lignes de crêtes et les bassins versants associés. Ces derniers sont utilisés dans la section 2.2.2 pour déterminer l'hydrologie de la surface créée.

#### 2.2.1.1 Un mouvement brownien

Notre méthode génère simultanément plusieurs lignes de crêtes. Sur un plan bidimensionnel, ces crêtes représentent un ensemble de lignes brisées. Deux segments, chacun appartenant à une ligne brisée, ne s'intersectent que s'ils représentent une extrémité commune aux deux lignes, i.e. le début ou la fin des deux lignes brisées. En trois dimensions, les altitudes des lignes brisées peuvent varier indépendamment de leurs positions dans le plan.

Ainsi, une approximation de mouvement brownien fractionnaire est utilisée pour simuler la ligne de crête vue du dessus. Nous utilisons une liste de points mobiles appelés *mobiles de crête*, chacun simule une particule subissant un mouvement brownien. Soit à chaque instant  $t$ , un mobile  $P$  possède, dans le plan horizontal, une position  $\mathcal{X}(P_t)$ , une orientation  $\Theta(P_t)$  et une vitesse constante  $\mathcal{V}(P)$ ; indépendamment, ce mobile possède une altitude notée  $\mathcal{Z}(P_t)$ . Posons l'ensemble des  $2n$  mobiles à un instant  $t$ , notés  $P_t^{(i)}$  avec  $0 \leq i \leq 2n$  et  $(t,n) \in \mathbb{N}^2$ . A  $t = 0$ , les mobiles sont placés par couple et aléatoirement sur un MNT vide de dimensions  $L \times H$ . A chaque position  $(x,y)$  correspond une cellule  $\mathcal{T}[y][x]$  du MNT. Nous ajoutons un champ identifiant, donné par  $\mathcal{T}[y][x].id = k$ , à la structure de données sous-jacente à une cellule<sup>4</sup>; cet identifiant est unique pour chaque couple de mobiles  $(P_0^{(2k)}, P_0^{(2k+1)})$ ,  $0 \leq k \leq n$  positionnés à l'initialisation au même point  $(x,y)$  chacun est orienté dans une direction opposée. Soit pour tous les couples de

---

4. Soit la cellule  $\mathcal{T}[y][x]$  positionnée en coordonnées  $(x,y)$  d'un MNT  $\mathcal{T}[H][L]$  de dimensions  $L \times H$ . Cette cellule contient un champ altitude sur 16 bits noté  $\mathcal{T}[y][x].a$ , un champ état sur 2 bits noté  $\mathcal{T}[y][x].e$  et pouvant prendre les valeurs *inconnu*, *connu*, *crete* ou *riviere* et enfin un champ identifiant, noté  $\mathcal{T}[y][x].id$ , sur 14 bits.

particules  $(P_0^{(2k)}, P_0^{(2k+1)})$  en  $t = 0$  nous posons :

$$\begin{aligned}
\text{pour} \quad & \mathcal{X}(P_0^{(2k)}) = \mathcal{X}(P_0^{(2k+1)}) = (x, y) \\
\text{avec} \quad & x = \lfloor Y \times L \rfloor \text{ et } y = \lfloor Y \times H \rfloor \text{ où } \mathcal{T}[y][x].e \neq \text{crete} \\
\text{et} \quad & \Theta(P_0^{(2k+1)}) = \pi + \Theta(P_0^{(2k)}) \text{ et } \Theta(P_0^{(2k)}) = Y \times \pi \\
\text{nous avons} \quad & \begin{cases} \mathcal{T}[y][x].e = \text{crete} \\ \mathcal{T}[y][x].id = k \\ \mathcal{T}[y][x].a = \mathcal{Z}(P_0^{(2k)}) = \mathcal{Z}(P_0^{(2k+1)}) = \lfloor Y \times \dim(\mathcal{T}[y][x].a) \rfloor \end{cases}
\end{aligned} \tag{2.1}$$

où  $Y$  est un nombre pseudo-aléatoire compris dans l'intervalle  $[0, 1]$ .

Pour générer les lignes de crêtes, nous déplaçons à chaque instant  $t$ , selon leur orientation et leur vitesse, toutes les particules  $P_t^{(i)}$  appartenant à la liste des points mobiles. Si la nouvelle position entière  $(x', y')$  donnée à partir de  $X(P_{t+1}^{(i)})$  est différente de la position entière précédente  $(x, y)$  donnée à partir de  $X(P_{t+1}^{(i)})$  alors les positions  $(x, y)$  et  $(x', y')$  sont reliées par un segment de droite discrète. Tant qu'aucune intersection n'est détectée, toute cellule traversée par ce segment sera étiquetée *crete* et son identifiant positionné à  $\lfloor \frac{i}{2} \rfloor$ . Si une intersection est détectée, la progression du mobile est stoppée, il est ensuite supprimé de la liste des mobiles. Une intersection est détectée quand les cellules traversées, ou intersectées (i.e. les deux cellules de la diagonale opposée si le déplacement est diagonal) par la droite discrète sont déjà étiquetées *crete* et possèdent un identifiant différent de celui de la particule. Aussi, nous considérons comme intersection une particule sortie du domaine de définition du MNT.

Ces déplacements effectués pour tous les mobiles de la liste, nous faisons varier leur orientation selon deux constantes, notées  $\delta\Theta_l(P^{(i)})$  et  $\delta\Theta_g(P^{(i)})$ , préalablement fixées, comme pour la vitesse, pour chaque couple de particules. Pour tout instant  $t$ ,  $\delta\Theta_l(P^{(i)})$  est le maxima de la variation locale  $|\Theta(P_{t+1}^{(i)}) - \Theta(P_t^{(i)})|$  et  $\delta\Theta_g(P^{(i)})$  est le maxima de la variation globale  $|\Theta(P_{t+1}^{(i)}) - \Theta(P_0^{(i)})|$ . Le processus est ainsi répété tant qu'il reste des particules dans la liste des points mobiles. La figure 2.4 donne, selon le paramétrage de la vitesse  $\mathcal{V}(P^{(0)}) = \mathcal{V}(P^{(1)})$  des variations d'angle  $\delta\Theta_l(P^{(0)}) = \delta\Theta_l(P^{(1)})$  et  $\delta\Theta_g(P^{(0)}) = \delta\Theta_g(P^{(1)})$ , quelques exemples de lignes brisées obtenues pour un couple de particules. De haut en bas, le paramétrage est le suivant :

1.  $\mathcal{V} = 2$ ,  $\delta\Theta_l = \frac{\pi}{10}$  et  $\delta\Theta_g = \frac{\pi}{2}$  ;
2.  $\mathcal{V} = 2$ ,  $\delta\Theta_l = \frac{\pi}{2}$  et  $\delta\Theta_g = \frac{\pi}{2}$  ;

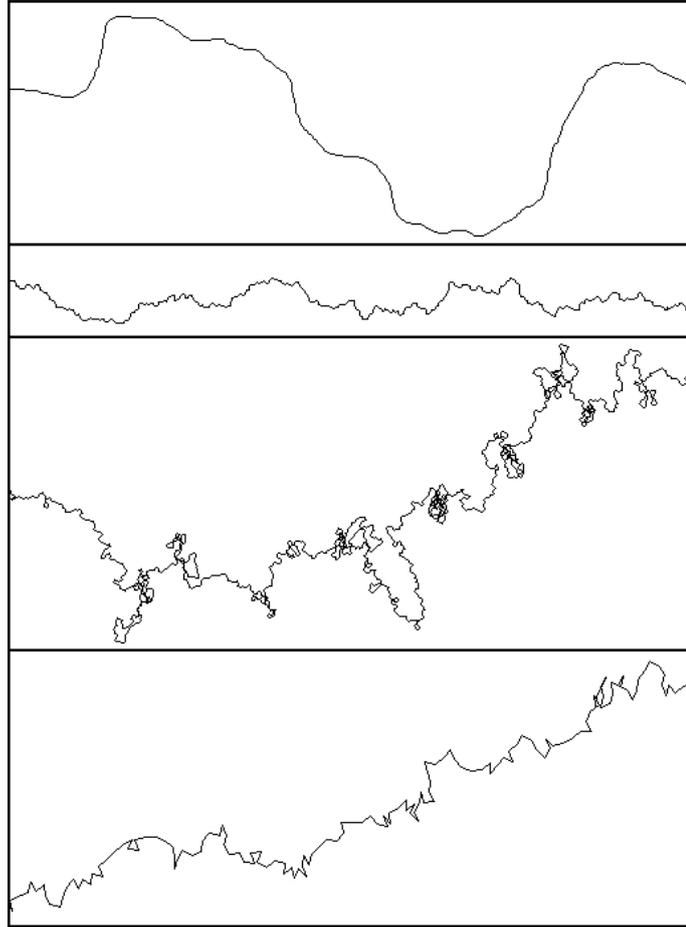


FIG. 2.4 – Vues de haut des lignes brisées obtenues avec notre générateur de crêtes.

3.  $\mathcal{V} = 2$ ,  $\delta\Theta_l = \frac{\pi}{2}$  et  $\delta\Theta_g = \frac{10\pi}{11}$  ;

4.  $\mathcal{V} = 8$ ,  $\delta\Theta_l = \pi$  et  $\delta\Theta_g = \frac{2\pi}{3}$  ;

dans tous les cas  $\Theta(P_0) = 0$  et  $\Theta(P_0') = \pi$ .

Enfin, concernant l'altitude d'une particule, sa valeur peut varier indépendamment de sa position sur le plan, ici nous choisissons de la faire décroître en fonction de la distance parcourue (soit du temps) mais aussi en fonction du calcul des altitudes des cellules des versants. **Algorithme 2** donne le détail du processus de génération des lignes de crêtes dans un MNT vide ; le calcul des altitudes du segment de la ligne de crête et des bassins versants est fait lors de l'appel à la procédure `calculer_versants` dont le détail est donné dans **Algorithme 3**.

---

**Algorithme 2** Processus de génération de lignes de crête.
 

---

**ENTRÉES:**  $\mathcal{L}$  la liste des  $2n$  indices des mobiles  $P^{(i)}$  ainsi que  $\mathcal{T}[H][L]$  un MNT vide initialisés tel que décrit dans (2.1);  $X$  est un nombre pseudo-aléatoire compris dans l'intervalle  $[-1,1]$ .

**SORTIES:** un MNT-squelette décrit dans  $\mathcal{T}$ .

$t \leftarrow 0$

**tantque**  $\mathcal{L} \neq \emptyset$  **faire**

**pour tout** indice  $i$  dans  $\mathcal{L}$  **faire**

$(x,y) \leftarrow \lfloor \mathcal{X}(P_t^{(i)}) \rfloor$

$\mathcal{X}(P_{t+1}^{(i)}) \leftarrow \mathcal{X}(P_t^{(i)}) + \mathcal{V}(P_t^{(i)}) \times (\cos(\Theta(P_t^{(i)})), \sin(\Theta(P_t^{(i)})))$

$(x',y') \leftarrow \lfloor \mathcal{X}(P_{t+1}^{(i)}) \rfloor$

**répéter**

$\delta\theta \leftarrow X \times \delta\Theta_l(P_t^{(i)})$

$\delta\theta_0 \leftarrow \Delta(\Theta(P_t^{(i)}) + \delta\theta, \Theta(P_0^{(i)}))$

**jusqu'à**  $\delta\theta_0 \leq \delta\Theta_g(P_t^{(i)})$

$\Theta(P_{t+1}^{(i)}) \leftarrow \Theta(P_t^{(i)}) + \delta\theta_0$

**si**  $(x',y') = (x,y)$  **alors**

**CONTINUER L'ITÉRATION**

**finsi**

$id \leftarrow \lfloor \frac{i}{2} \rfloor$

**pour tout**  $(x_j, y_j)$  de la droite discrète  $((x,y), (x',y'))$  **faire**

**si**  $\left( \begin{array}{l} (\mathcal{T}[y_j][x_j].e = crete \text{ et } \mathcal{T}[y_j][x_j].id \neq id) \text{ ou} \\ (\mathcal{T}[y_{j-1}][x_j].e = crete \text{ et } \mathcal{T}[y_{j-1}][x_j].id \neq id \text{ et } \\ \mathcal{T}[y_j][x_{j-1}].e = crete \text{ et } \mathcal{T}[y_j][x_{j-1}].id \neq id \end{array} \right)^{(*)}$  **alors**

      supprimer  $i$  de  $\mathcal{L}$

**ARRÊTER L'ITÉRATION**

**finsi**

$\mathcal{T}[y_j][x_j].e \leftarrow crete$  **et**  $\mathcal{T}[y_j][x_j].id \leftarrow id$

**fin pour**

  calculer\_versants( $P^{(i)}$ )

**fin pour**

$t \leftarrow t + 1$

**fin tantque**

---

(\*) Test à effectuer uniquement lors d'un déplacement oblique.

### 2.2.1.2 Le calcul des bassins versants<sup>5</sup>

Nous proposons une méthode rapide pour le calcul des altitudes des versants aidant à simuler la création d'un réseau d'écoulement de fluides pour un ensemble de crêtes données. Pour illustrer la méthode, nous utiliserons les lignes de crêtes générées précédemment ainsi que la notation associée<sup>6</sup>. Soit à l'instant  $t$  une particule de ligne de crête notée  $P_t$ . Cette particule se déplace à partir d'une position dans le plan notée  $\mathcal{X}(P_t)$  dans une direction donnée par  $\Theta(P_t)$  et à une vitesse  $\mathcal{V}(P_t)$ . A  $t + 1$  la particule se trouve en position  $\mathcal{X}(P_{t+1})$  et nous calculons les versants de chaque côté du segment  $P_tP_{t+1}$ . Nous posons une largeur de versant<sup>7</sup> égale à  $4 \times \sigma_1$  où  $\sigma_1$  est une constante donnée pour chaque particule. Nous calculons la position de quatre points  $A, B, C$  et  $D$  servant à délimiter la zone contenant les cellules des versants correspondants au segment de ligne de crête  $P_tP_{t+1}$ . Soit  $A$  et  $D$  appartiennent à la droite orthogonale à  $(P_t, \Theta P_t)$ ; ils se trouvent à une distance  $4 \times \sigma_1$  et sont respectivement placés à gauche et à droite du mobile  $P_t$ .  $B$  et  $C$  sont obtenus de la même manière mais pour l'instant  $t+1$ . Ainsi, selon la variation de l'orientation de la particule nous pouvons obtenir une des deux situations décrites sur le schéma en haut de la figure 2.5. Soit  $AD$  et  $BC$  ne s'intersectent pas, alors les cellules comprises dans les triangles  $ADB$  et  $BDC$  sont considérées comme les versants correspondant au segment  $P_tP_{t+1}$ . Soit  $AD$  et  $BC$  s'intersectent en  $I$  et deux cas se présentent :  $I$  est à l'opposé de  $AB$  par rapport à  $P_tP_{t+1}$ , alors nous considérons seulement le triangle  $AIB$ ; ou bien  $I$  est à l'opposé de  $DC$  par rapport à  $P_tP_{t+1}$ , alors nous considérons seulement le triangle  $IDC$  (situation décrite sur le schéma haut-gauche de la figure 2.5). Ainsi nous obtenons pour chaque déplacement un ou deux triangles délimitant la zone des versants à calculer. Le schéma en bas à gauche de la figure 2.5 montre la zone délimitant les versants pour une plus longue séquence de déplacements; les parties entourées par un cercle dénotent les triangles non pris en compte en cas d'intersection.

5. Un bassin versant est la région délimitée, drainée par un cours d'eau et ses tributaires, dont elle constitue l'aire d'alimentation [29]. Les termes bassin fluvial ou bassin hydrographique sont des synonymes. Par commodité, la note concernant cette notion est répétée ici.

6. La méthode s'étend aisément au cas générale où des lignes de crêtes seraient prédéfinies autrement qu'avec le générateur précédemment présenté (cf. section 2.2.1.1). L'utilisateur peut, par exemple, dessiner de manière interactive son croquis de crêtes obtenant instantanément les bassins versants correspondants. Il pourra paramétrer la pente de chaque versant le long des lignes de crête avant de calculer le réseau fluvial correspondant et enfin le modèle réaliste de terrain.

7. La largeur d'un versant est obtenue pour une coupe en largeur d'une crête. C'est la distance horizontale séparant le sommet de la crête (soit la ligne de crête) du point le plus bas.

Afin de déterminer les cellules du MNT appartenant aux versants, un parcours horizontal ligne par ligne, ou *scanline* [4, 125], est effectué pour chaque triangle. Nous posons le point  $M$  correspondant à une cellule  $\mathcal{T}[y_M][x_M]$  appartenant à un des versants de la ligne de crête  $P_t P_{t+1}$ . Puis nous calculons la projection orthogonale, notée  $M'$ , du point  $M$  sur l'axe  $(P_t, \overrightarrow{P_t P_{t+1}})$ . Le calcul de l'altitude de la cellule est pondéré par le produit de deux fonctions gaussiennes  $\mathcal{G}_1$  et  $\mathcal{G}_2$ . Chacune varie proportionnellement à une distance à la particule  $P_t$ . Ces deux distances, notées  $d_1$  et  $d_2$ , sont respectivement les distances euclidiennes  $\|\overrightarrow{MM'}\|$  et  $\|\overrightarrow{M'P}\|$  où  $P$  est le point aux coordonnées entières  $[\mathcal{X}(P_t)]$ . Ainsi pour une particule  $P_t$  dont l'altitude est donnée par  $\mathcal{Z}(P_t)$  et pour une cellule  $\mathcal{T}[y_M][x_M]$  appartenant au versant de  $P_t P_{t+1}$  son altitude est calculée par :

$$\begin{cases} \mathcal{T}[y_M][x_M].a = \mathcal{Z}(P_t) \times \mathcal{G}_1(d_1) \times \mathcal{G}_2(d_2) \\ \mathcal{G}_1(x) = e^{-\left(\frac{x}{\sigma_1}\right)^2} \\ \mathcal{G}_2(x) = e^{-\left(\frac{x}{\sigma_2}\right)^2} \end{cases} \quad (2.2)$$

puis nous posons :

$$\mathcal{Z}(P_{t+1}) = \mathcal{Z}(P_t) \times \mathcal{G}_2(\|\overrightarrow{P_t P_{t+1}}\|) \quad (2.3)$$

Soit, pour un couple de particules donné et pour  $\sigma_2$  constante, l'altitude de la ligne de crête générée décroît en fonction de la distance<sup>8</sup> à la position en  $t = 0$  et selon l'allure décrite par  $\mathcal{G}_2$ . Le maillage donné en bas à gauche de la figure 2.5 montre une vue tridimensionnelle du résultat obtenu pour un couple de particules  $(P, P')$ ,  $\sigma_1$  et  $\sigma_2$  constantes. L'**Algorithme 3** résume la méthode proposée pour le cas d'une particule en déplacement entre l'instant  $t$  et  $t + 1$ . Enfin, la figure 2.6 donne un exemple de crête obtenue quand les altitudes successives de la particule décroissent aléatoirement ; soit  $\sigma_2$  varie aléatoirement et  $\mathcal{G}_2$  excentrée. Pour cet exemple, des discontinuités sur la dérivée de la courbe apparaissent aux jonctions des segments. Dans ce cas, un lissage par filtre passe-bas [22] moyennant de taille  $3 \times 3$  (érosion par diffusion thermique) est appliqué afin de supprimer les minima locaux présents aux jonctions et ne pas influencer le calcul du réseau fluvial.

### 2.2.2 Hydrologie de surface

Nous proposons une méthode pour calculer un réseau fluvial à partir d'un MNT décrivant des bassins versants. Le réseau obtenu est composé d'un ou

8. La distance à la position initiale, donnée pour  $t = 0$ , est calculée en sommant toutes les distances euclidiennes entre les positions successives ; soit  $\sum_{i=1}^t \|\overrightarrow{P_{i-1} P_i}\|$ .

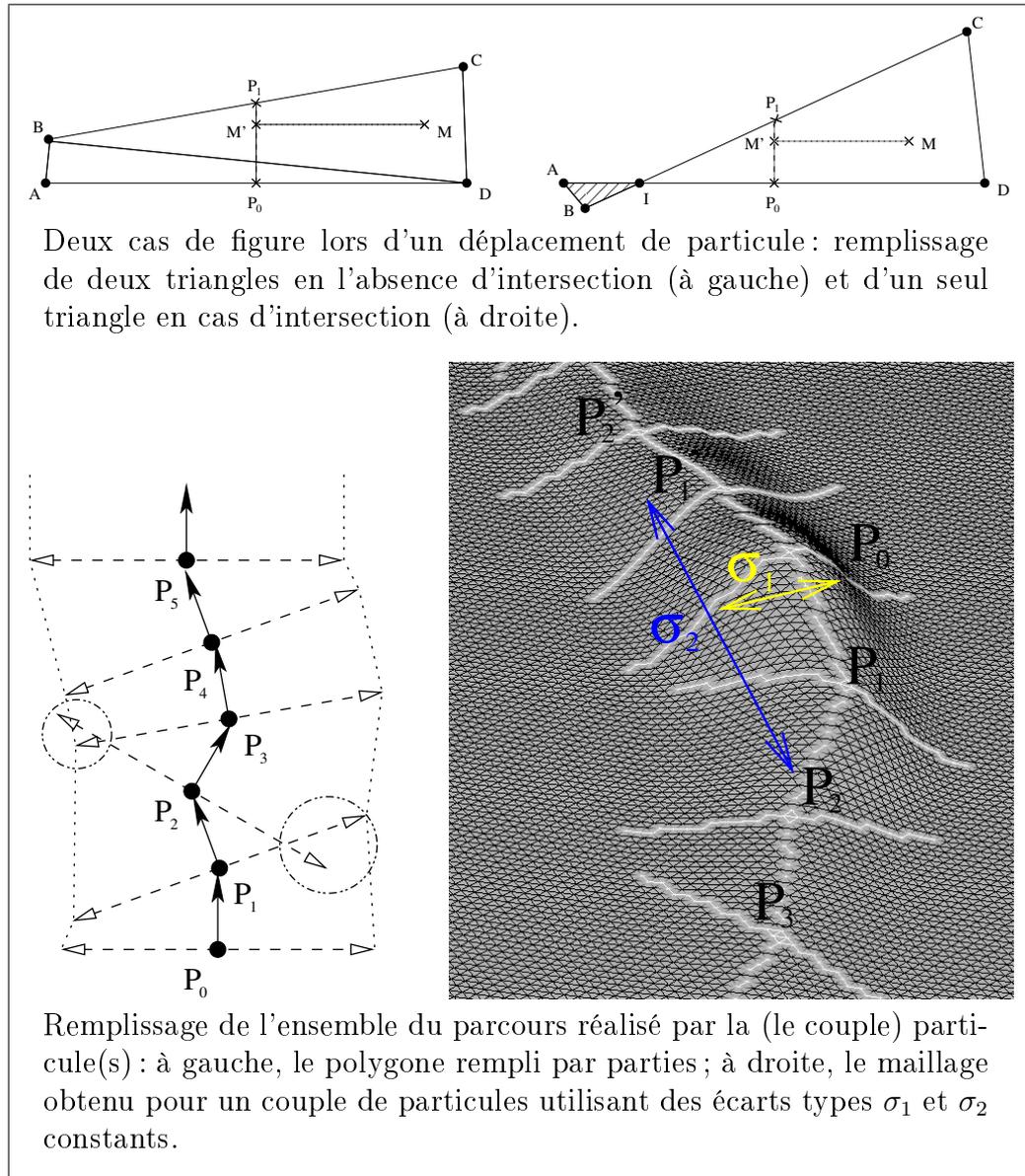


FIG. 2.5 – Calcul des bassins versants par remplissage de polygones.

---

**Algorithme 3** Calcul des bassins versants d'une ligne de crête ; la procédure correspond à l'appel de `calculer_versants` effectué dans **Algorithme 2**.

---

**ENTRÉES:** Une particule  $P$  donnée à un instant  $t$ .

**SORTIES:** Modifie les altitudes des cellules appartenant aux versants du segment de ligne de crête  $P_t P_{t+1}$ .

$P \leftarrow \lfloor \mathcal{X}(P_t) \rfloor$

Calculer les quatre points  $A, B, C$  et  $D$  tels que :

$$A \leftarrow \mathcal{X}(P_t) + 4 \times \sigma_1 \times (\cos(\Theta(P_t) + \frac{\pi}{2}), \sin(\Theta(P_t) + \frac{\pi}{2}))$$

$$D \leftarrow \mathcal{X}(P_t) + 4 \times \sigma_1 \times (\cos(\Theta(P_t) - \frac{\pi}{2}), \sin(\Theta(P_t) - \frac{\pi}{2}))$$

$$B \leftarrow \mathcal{X}(P_{t+1}) + 4 \times \sigma_1 \times (\cos(\Theta(P_{t+1}) + \frac{\pi}{2}), \sin(\Theta(P_{t+1}) + \frac{\pi}{2}))$$

$$C \leftarrow \mathcal{X}(P_{t+1}) + 4 \times \sigma_1 \times (\cos(\Theta(P_{t+1}) - \frac{\pi}{2}), \sin(\Theta(P_{t+1}) - \frac{\pi}{2}))$$

**si**  $AD$  intersecte  $BC$  en  $I$  **alors**

**si**  $I$  est à l'opposé de  $AB$  par rapport à  $P_t P_{t+1}$  **alors**

$$\mathcal{L} \leftarrow \{AIB\}$$

**sinon**

$$\mathcal{L} \leftarrow \{IDC\}$$

**finsi**

**sinon**

$$\mathcal{L} \leftarrow \{ADB, BDC\}$$

**finsi**

**pour tout**  $\Delta$  triangle de  $\mathcal{L}$  **faire**

**pour tout**  $M$  point à coordonnées discrètes  $(x, y)$  du  $\Delta$  **faire**

$$M' \leftarrow \perp (M, P_t P_{t+1})$$

$$\mathcal{T}[y][x].a \leftarrow \max(\mathcal{Z}(P_t) \times \mathcal{G}_1(\|\overrightarrow{MM'}\|) \times \mathcal{G}_2(\|\overrightarrow{M'P}\|), \mathcal{T}[y][x].a) \text{ (cf. équation 2.2)}$$

**fin pour**

**fin pour**

$$\mathcal{Z}(P_{t+1}) \leftarrow \mathcal{Z}(P_t) \times \mathcal{G}_2(\|\overrightarrow{P_t P_{t+1}}\|)$$


---

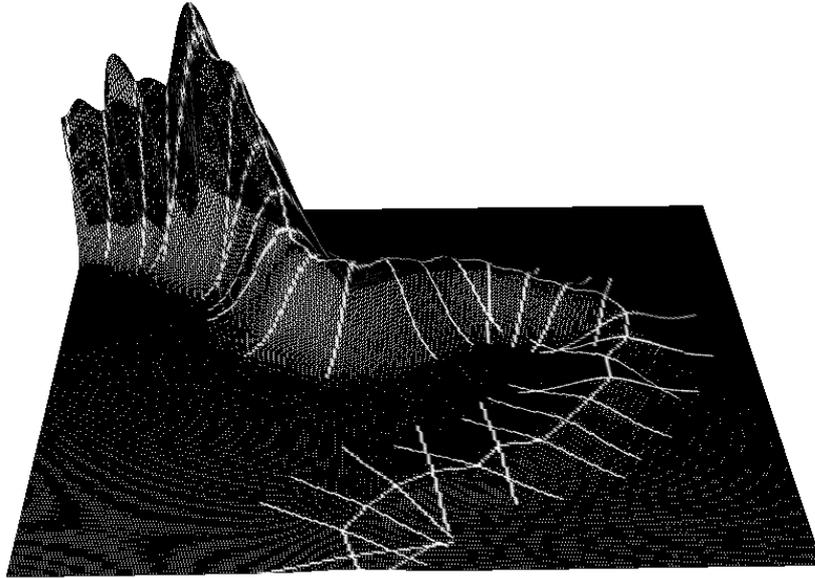


FIG. 2.6 – *Exemple de crête où les altitudes successives du couple de particules décroissent aléatoirement.*

plusieurs assemblages<sup>9</sup> de cours d'eau de différentes catégories ; il est généré à partir d'un ensemble fini de petits cours d'eau donnés à l'initialisation. Ainsi notre approche se place à l'opposé de la majorité des approches modélisant des terrains érodés. Nous ne simulons pas l'érosion d'un modèle initial de terrain dans le but de le rendre plus réaliste ; nous générons un modèle réaliste de terrain à partir d'un modèle minimaliste d'érosion. Soit, l'objectif de la méthode proposée est de produire, le plus rapidement, un modèle hydrologique plausible pour un modèle donné de bassins versants. Cette base prend la forme d'un MNT-squelette composé de cellules de lignes de crête étiquetées *crete* et de cellules du réseau fluvial étiquetées *riviere*. Alors, un modèle réaliste de terrain peut, par la suite, être obtenu à l'aide d'un des algorithmes fractals présentés dans **Algorithme 1**, **Algorithme 10** ou encore **Algorithme 11**.

Soit un modèle hydrologique est généré dans un MNT décrivant un modèle de bassins versants où les lignes de crête sont identifiées telles que décrit dans la partie précédente (cf. section 2.2.1). A l'initialisation, nous posons à intervalles préfixés et de chaque côté le long des lignes de crête des particules de type point mobile simulant l'écoulement d'un cours d'eau. Ces particules sont appelées particules d'écoulement et notées  $M^{(i)}$ . A chaque instant, ces

---

9. Le terme assemblage, par transposition au domaine de la théorie des graphes, est équivalent à la définition d'un graphe à une composante connexe.

particules se déplacent sur la surface du terrain et peuvent interagir entre-elles selon les informations stockées dans le MNT. Dans leur ensemble, ces déplacements donnent le modèle hydrologique final.

Ainsi, dans un premier temps, nous donnons le détail du modèle physique associé à chacune des particules d'écoulement (cf. section 2.2.2.1). Puis nous proposons un algorithme permettant de gérer leurs interactions au cours du temps jusqu'à la stabilisation du système. Enfin, nous montrons différents exemples de modèles obtenus.

### 2.2.2.1 Notre approximation du modèle physique

Dans cette partie, nous nous intéressons au calcul de la trajectoire d'un cours d'eau obtenue en reliant deux à deux les positions successives d'une particule d'écoulement. Ces positions sont calculées incrémentalement. Nous utilisons un modèle cinématique discret d'un point mobile.

Soit une particule d'écoulement  $M_t^{(i)}$  représentant un point mobile à un instant  $t$ ; cette particule est définie par :

- $i$  son identifiant ;
- $\mathcal{X}(M_t^{(i)})$ ,  $\vec{\mathcal{V}}(M_t^{(i)})$ ,  $\mathcal{A}(M_t^{(i)})$  donnent respectivement la position, la vitesse et l'accélération de la particule dans le plan décrit par le MNT.  $\mathcal{Z}(M_t^{(i)}) \in [0,1]$  est l'altitude de la particule, elle est obtenue après mise à l'échelle de l'altitude correspondante dans le MNT ;
- $Stop(M_t^{(i)})$  indique si la particule est stoppée, i.e. le système ne la considère plus au temps  $t$ . Cette propriété est initialisée à *faux* ;
- $Boucles(M_t^{(i)})$  indique le nombre de boucles effectuées par la trajectoire de la particule ;
- $m(M_t^{(i)})$  la masse de la particule à l'instant  $t$  ;
- $Confluent(M_t^{(i)})$  indique si le cours d'eau sous-jacent à la trajectoire de la particule est un affluent pour un autre cours d'eau ; dans ce cas,  $Confluent(M_t^{(i)})$  donne une coordonnée valide dans le MNT où la cellule correspondante contient l'identifiant du cours d'eau destination ;  $Confluent(M_t^{(i)})$  contient  $(-1, -1)$  dans le cas contraire ;
- $\mathcal{L}(M_t^{(i)})$  est la liste des positions discrètes,  $id$  et masses successives prises par la particule ; une position est notée  $\mathcal{X}(\mathcal{L}(M_t^{(i)})[j])$  avec  $j \in \{0, \dots, n-1\}$  et  $n$  le nombre de cellules traversées par la particule ; chaque élément de la liste informe sur l'éventuel identifiant  $id$  d'un affluent intersectant la trajectoire de la particule en cette position, soit nous notons  $Affluent(\mathcal{L}(M_t^{(i)})[j]) = id$  s'il y a eu intersection ou  $-1$  sinon ; enfin,  $m(\mathcal{L}(M_t^{(i)})[j])$  indique la masse de la particule en cette position discrète ;

- $Valide(M_t^{(i)})$ , cette propriété est initialisée à *faux*. Elle intervient en fin de processus et indique si le cours d'eau sous-jacent à la trajectoire de la particule est valide et sera utilisé dans la définition du MNT-squelette. Nous invalidons les cours d'eau taris, i.e. soit les cours d'eau n'arrivant pas au limites du MNT, ou toute position considérée comme embouchure, ou bien ceux ne constituant pas un affluent pour un plus gros cours d'eau valide ; cette définition est récursive.

A l'initialisation, le mobile est positionné à proximité d'une ligne crête avec une vitesse nulle. La position est à coordonnées entières  $(x,y)$  et la cellule correspondante dans le MNT est étiquetée *inconnu*. A chaque instant, nous calculons le vecteur d'accélération dans le plan selon le schéma donné figure 2.7 ; ce schéma détaille le calcul de l'accélération sur l'axe des  $x$ . Ainsi, pour une position  $(x,y,z)$  où  $(x,y) = \lfloor \mathcal{X}(M_t^{(i)}) \rfloor$  et  $z = \mathcal{Z}(M_t^{(i)})$ ,  $a_x$  la composante en  $x$  de l'accélération est calculée en fonction de la direction (ici sur l'axe des  $x$ ) empruntée par la particule à l'instant  $t$ . Le choix de la direction est conditionné par les différences de pente  $dz_1$  et  $dz_2$  respectivement obtenues pour un incrément en  $x$ , noté  $dx$ , et valant  $(-1)$  ou  $(+1)$ . Nous choisissons la plus grande pente descendante ou la plus petite pente ascendante ; soit le  $\min(dz_1, dz_2)$  avec  $(dz_1, dz_2) \in [-1,1]^2$ . Le choix de pente effectué, le calcul de la composante d'accélération se fait selon cette pente et utilisant un modèle newtonien ; la figure 2.7 montre la configuration des forces utilisée pour  $dx = -1$ . Il en résulte le bilan des forces :  $\sum \vec{F} = \vec{P}_\alpha = m \times \vec{a}$  ;  $a$  est réduit à sa composante en  $x$  et nous posons l'approximation  $a_x = dx \times \|\vec{P}_\alpha\|$ . Nous appliquons le même procédé pour le calcul de  $a_y$ , la composante en  $y$  de l'accélération. Nous obtenons finalement  $\vec{\mathcal{A}}(M_t^{(i)}) = (a_x, a_y)$ .

Ainsi, suivant ce modèle cinématique, une particule  $M_t^{(i)}$  se déplace dans le plan décrit par le MNT et inscrit sa trajectoire sur l'ensemble des cellules traversées. Nous introduisons une composante stochastique dans le calcul de la trajectoire du mobile ; cette composante stochastique nous permet d'obtenir des formes de méandres : il s'agit de courbes à l'aspect sinueux et adoptées par des cours d'eau dits paresseux. Cet aspect sinueux est plus prononcé au niveau des vallées ; ainsi, pour une particule donnée, nous perturbons seulement l'orientation du vecteur vitesse rendant l'effet plus prononcé pour les zones à faible pente. Soit, à un instant  $t$  une forte accélération donnée par une forte pente réduit l'effet du vecteur vitesse calculé à l'instant  $t - dt$ .

D'autre part, nous freinons et maximisons le mouvement de la particule en introduisant un calcul de frottements dont le coefficient  $k$  est lié à une constante notée  $r_{max}$  et dénotant la quantité de déplacement souhaitée maximale lors d'une itération (soit  $t = t + dt$ ). Dans la pratique  $r_{max}$  est positionnée à une dizaine d'unités-terrain. Ce calcul de frottement, effectué au même

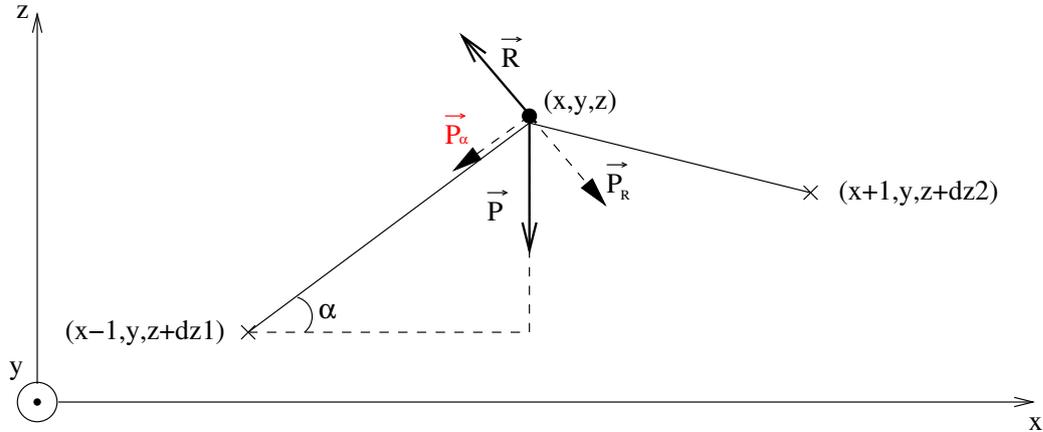


FIG. 2.7 – *Modèle cinématique discret utilisé pour un point mobile simulant l'écoulement d'un cours d'eau.*

moment que la perturbation de la trajectoire, intervient seulement après un déplacement amenant la particule vers une nouvelle position à coordonnées entières. Pendant ce changement de coordonnées entières, nous modifions les états et les identifiants des cellules traversées par la particule. Les positions successives traversées sont stockées dans  $\mathcal{L}(M_t^{(i)})$  afin de pouvoir défaire le déplacement en cas de retour sur position ; i.e. le cas d'un minima local. Dans ce cas  $Boucles(M_t^{(i)})$  est incrémentée, la boucle est défaite et la particule repart si  $B\_MAX$  n'est pas atteint (cette valeur est fixée de manière empirique à une trentaine de boucles par particule pour un terrain composé d'un million de cellules). Dans le cas contraire, la particule est stoppée et sa trajectoire est invalidée. Par ailleurs, dans le cas d'une intersection avec la trajectoire d'une autre particule, l'intersectant  $M_t^{(i)}$  devient affluent de l'intersecté et la confluence est marquée au niveau de la liste  $\mathcal{L}(M_t^{(j)})$  où  $j$  est l'identifiant du cours d'eau intersectée (cf. **Algorithme 9**).  $M_t^{(i)}$  est stoppée et  $Confluent(M_t^{(i)})$  reçoit la coordonnée du point de confluence. Enfin, une particule est stoppée puis sa trajectoire ainsi que celles de ses éventuelles affluents sont validées à condition d'atteindre les limites du MNT considérées comme embouchures.

Nous donnons dans **Algorithme 6** le détail du processus de déplacement pour une particule et pour un laps de temps  $dt$  ; dans cet algorithme nous utiliserons les procédures décrites dans **Algorithme 4** et **5**. L'ensemble du système, gérant les interactions entre les différentes particules, est détaillé dans la partie ci-après (cf. section 2.2.2.2).

---

**Algorithme 4** Procédure de calcul de l'accélération d'une particule simulant l'écoulement d'un cours d'eau.

---

**ENTRÉES:** Un mobile  $M^{(i)}$  représentant une particule donnée à un instant  $t$  et se déplaçant sur un MNT  $\mathcal{T}[H][L]$ .

**SORTIES:** Calcule  $\vec{A}(M_t^{(i)})$  l'accélération de la particule.

$(x, y) \leftarrow \lfloor \mathcal{X}(M_t^{(i)}) \rfloor$

$z_0 \leftarrow \frac{\mathcal{T}[y][x].a}{2^{16}-1}$

**si**  $(x-1, y) \in \mathcal{T}$  **alors**  $z_1 \leftarrow \frac{\mathcal{T}[y][x-1].a}{2^{16}-1}$  **sinon**  $z_1 \leftarrow -1$

**si**  $(x+1, y) \in \mathcal{T}$  **alors**  $z_2 \leftarrow \frac{\mathcal{T}[y][x+1].a}{2^{16}-1}$  **sinon**  $z_2 \leftarrow -1$

**si**  $(dz_1 \leftarrow z_1 - z_0) < (dz_2 \leftarrow z_2 - z_0)$  **alors**

$dx \leftarrow -1$  **et**  $dz \leftarrow dz_1$

**sinon**

$dx \leftarrow +1$  **et**  $dz \leftarrow dz_2$

**finsi**

$a_x \leftarrow dx \times \mathcal{G} \times \sin(\tan^{-1}(|dz|))$

**si**  $(x, y-1) \in \mathcal{T}$  **alors**  $z_1 \leftarrow \frac{\mathcal{T}[y-1][x].a}{2^{16}-1}$  **sinon**  $z_1 \leftarrow -1$

**si**  $(x, y+1) \in \mathcal{T}$  **alors**  $z_2 \leftarrow \frac{\mathcal{T}[y+1][x].a}{2^{16}-1}$  **sinon**  $z_2 \leftarrow -1$

**si**  $(dz_1 \leftarrow z_1 - z_0) < (dz_2 \leftarrow z_2 - z_0)$  **alors**

$dy \leftarrow -1$  **et**  $dz \leftarrow dz_1$

**sinon**

$dy \leftarrow +1$  **et**  $dz \leftarrow dz_2$

**finsi**

$a_y \leftarrow dy \times \mathcal{G} \times \sin(\tan^{-1}(|dz|))$

$\vec{A}(M_t^{(i)}) \leftarrow (a_x, a_y)$

---

---

**Algorithme 5** Procédure utilisée pour perturber la trajectoire d'une particule simulant l'écoulement d'un cours d'eau.

---

**ENTRÉES:** Un mobile  $M^{(i)}$  donnée à un instant  $t$  et se déplaçant dans un laps de temps  $dt$ ;  $r_{max}$  la quantité de déplacement souhaitée maximale en chaque itération;  $X$  est un nombre pseudo-aléatoire compris dans l'intervalle  $[-1,1]$ . Cette procédure est appelée seulement quand le mobile change de coordonnées entières dans le plan.

**SORTIES:** Perturbe la trajectoire du mobile en modifiant l'orientation de son vecteur vitesse; la vitesse est freinée et maximisée par l'introduction de frottements.

$\vec{V}(M_t^{(i)})$  est donné par  $(v_x, v_y)$

Réaliser le passage des coordonnées cartésiennes  $(v_x, v_y) \in \mathbb{R}^2$  aux coordonnées polaires  $(r, \theta) \in \mathbb{R}^+ \times [-\pi, \pi]$

$$\theta \leftarrow \theta + \frac{\pi}{2} \times X \times dt$$

$$r \leftarrow \min(r, r_{max})$$

$$k \leftarrow \frac{dt}{2.0 \times r_{max}}$$

$$r \leftarrow r - k \times r^2$$

$$\vec{V}(M_t^{(i)}) \leftarrow (r \times \cos(\theta), r \times \sin(\theta))$$


---

---

**Algorithme 6** Processus réalisé lors du déplacement d'une particule simulant l'écoulement d'un cours d'eau.

---

**ENTRÉES:** Un mobile  $M^{(i)}$  représentant une particule donnée à un instant  $t$  et se déplaçant dans un laps de temps  $dt$  sur un MNT  $\mathcal{T}[H][L]$ .

**SORTIES:** Cette fonction  $\text{deplacement}(M_t^{(i)}, \mathcal{T})$  inscrit le déplacement de la particule sur le MNT fourni ; elle modifie les propriétés de la particule ; elle retourne *vrai* si une autre trajectoire est intersectée sinon *faux*.

$(x,y) \leftarrow \lfloor \mathcal{X}(M_t^{(i)}) \rfloor$

Calculer l'accélération  $\vec{\mathcal{A}}(M_t^{(i)})$  comme détaillé dans **Algorithme 4**.

$\vec{\mathcal{V}}(M_t^{(i)}) \leftarrow \vec{\mathcal{V}}(M_t^{(i)}) + \vec{\mathcal{A}}(M_t^{(i)}) \times dt$

si  $\vec{\mathcal{V}}(M_t^{(i)}) \approx \vec{0}$  et  $\vec{\mathcal{A}}(M_t^{(i)}) \approx \vec{0}$  alors

$\text{Stop}(M_t^{(i)}) \leftarrow \text{vrai}$  et **RETOURNER** *faux*

**finsi**

si  $(\mathcal{X}(M_t^{(i)}) \leftarrow \mathcal{X}(M_t^{(i)}) + \vec{\mathcal{V}}(M_t^{(i)}) \times dt) \notin \mathcal{T}$  alors

$\text{Stop}(M_t^{(i)}) \leftarrow \text{vrai}$  et  $\text{Valide}(M_t^{(i)}) \leftarrow \text{vrai}$  et **RETOURNER** *faux*

**finsi**

**pour tout**  $(x_j, y_j)$  de la droite discrète  $((x,y), \lfloor \mathcal{X}(M_t^{(i)}) \rfloor)$  sauf  $(x,y)$  **faire**

si  $\left( \begin{array}{l} (\mathcal{T}[y_j][x_j].e = \text{riviere}) \quad \text{ou} \\ \left( \begin{array}{l} \mathcal{T}[y_{j-1}][x_j].e = \text{riviere} \text{ et} \\ \mathcal{T}[y_j][x_{j-1}].e = \text{riviere} \end{array} \right)^{(*)} \end{array} \right)$  alors

$ii \leftarrow$  est l'identifiant de la rivière intersectée

$(x_{ii}, y_{ii}) \leftarrow$  est la position intersectée

si  $ii \neq i$  alors

$\text{Confluent}(M_t^{(i)}) \leftarrow (x_{ii}, y_{ii})$

**RETOURNER** *vrai*

**finsi**

défaire la trajectoire de  $M_t^{(i)}$  jusqu'à  $(x_{ii}, y_{ii})$  (cf. **Algorithme 7**)

si  $(\text{Boucles}(M_t^{(i)}) \leftarrow \text{Boucles}(M_t^{(i)}) + 1) \geq B\_MAX$  alors

$\text{Stop}(M_t^{(i)}) \leftarrow \text{vrai}$

**finsi**

**RETOURNER** *faux*

**finsi**

$\mathcal{T}[y_j][x_j].e \leftarrow \text{riviere}$  et  $\mathcal{T}[y_j][x_j].id \leftarrow i$  et ajouter  $((x_j, y_j), -1, m(M_t^{(i)}))$  dans  $\mathcal{L}(M_t^{(i)})$

**fin pour**

Perturber la trajectoire de  $M_t^{(i)}$  comme détaillé dans **Algorithme 5**.

**RETOURNER** *faux*

---

(\*) Test à effectuer uniquement lors d'un déplacement oblique.

### 2.2.2.2 Retour sur trace et validation

Nous détaillons dans cette partie les interactions entre toutes les particules simulant l'écoulement de plusieurs cours d'eau dans un même MNT. Toutes les particules non stoppées se déplacent en même temps selon leurs paramètres ; nous utilisons le MNT pour détecter les éventuelles intersections.

Ainsi, pour une particule d'écoulement  $M^{(i)}$  intersectant la trajectoire décrite par une deuxième particule  $M^{(j)}$ ,  $M^{(i)}$  est stoppée puis la trajectoire stockée dans  $\mathcal{L}(M^{(j)})$  est défaite jusqu'au point d'intersection. Soit un retour sur trace est effectué au niveau du MNT pour une partie de la trajectoire réalisée par  $M^{(j)}$  ; les états des cellules sont remis à *inconnu* et  $M^{(j)}$  repart du point de confluence avec une masse et un vecteur vitesse modifiés en fonction de la masse et du vecteur vitesse de  $M^{(i)}$  comme calculés à l'instant  $t$ . La nouvelle vitesse est une somme des deux vitesses  $\vec{V}(M_t^{(j)})$  et  $\vec{V}(M_t^{(i)})$  pondérées par les deux masses  $m(M_t^{(j)})$  et  $m(M_t^{(i)})$ . Une nouvelle masse  $m(M_t^{(j)}) = m(M_t^{(j)}) + m(M_t^{(i)})$  est calculée et sera stockée dans  $\mathcal{L}(M_t^{(j)})$  lors des prochains déplacements discrets.

Une fois toutes les particules stoppées, nous testons la validité des trajectoires générées. Ainsi, une trajectoire est considérée comme :

- trajectoire **valide** si la particule est valide. Soit, elle a été stoppée une fois arrivée à une embouchure (dans le cas présent, seules les limites du MNT sont considérées comme embouchures) ;
- trajectoire **valide** s'il existe une confluence avec une autre particule ayant une trajectoire valide (cette définition est récursive, cf. **Algorithme 8**) ;
- trajectoire **non valide** sinon (elle est défaite dans ce cas).

Finalement, pour chacune des trajectoires valides (i.e. les particules valides) nous effectuons un dernier parcours de la liste des positions discrètes afin de dessiner<sup>10</sup> le cours d'eau correspondant. Dans cette partie, nous utilisons l'information de masse de particule pour décider de la catégorie (i.e. le débit et la largeur) des cours d'eau. Dans le cas présent, les masses des particules sont initialisées à 1 et une échelle logarithmique est utilisée pour changer de catégorie ; le diamètre du cours d'eau vaut  $\log_2(m)$  où  $m$  est la masse de la particule en une position donnée.

**Algorithme 9** détaille le processus principal générant un modèle hydrologique pour un MNT décrivant des bassins versants et des particules placées le long des lignes de crête.

---

10. La procédure de dessin de droites discrètes est remplacée par une procédure de remplissage de quadrilatère convexe quelconque ; les cellules parcourues sont étiquetées *riviere*.

La figure 2.8 montre les résultats obtenus selon trois types de modèles de bassins versants. Les images de gauche donnent les MNTs où les trajectoires des cours d'eau ne sont pas visibles ; les trois MNTs ont une résolution de  $513^2$  cellules. Les images de droite représentent la vue tridimensionnelle correspondante dans laquelle les trajectoires des rivières sont coloriées en bleu et les lignes de crête en rouge. Le modèle du haut et celui du milieu ont été obtenus en approximativement 351 millisecondes sur notre architecture de tests (cf. la section 1.1.1.2) ; nous avons utilisé un seul couple de particules  $(P^0, P^1)$  générant une seule ligne de crête, positionné les vitesses à 5 unités-terrain et nous posons un couple de particules d'écoulement à chaque déplacement de particule de ligne de crête. La ligne de crête du modèle du haut est créée à altitude constante mais subit une grande variation d'orientation. À l'inverse, la ligne de crête du modèle du milieu décroît rapidement en altitude et subit de légères variations d'orientation. Enfin, pour le modèle du bas, nous utilisons plusieurs particules générant des lignes de crête à altitude légèrement décroissante et avec une petite variation d'orientation. Aussi, le pas de déplacement des particules de lignes de crête est plus élevé (10 unités-terrain) et nous initialisons moins de particules d'écoulement. Le modèle, calculé en approximativement 487 millisecondes, est utilisé (cf. les figures 2.9, 2.10, 3.4 et 3.5) comme squelette pour nos algorithmes fractals de génération de terrain avec contraintes

### 2.2.3 Modèle de terrain à crêtes et réseau fluvial : l'érosion

Nous présentons notre premier modèle réaliste de terrain entièrement généré<sup>11</sup>. La méthode par fenêtrage croissant est ici appliquée à un MNT-squelette décrivant des lignes de crête et une hydrologie de bassins versants. Soit pour un modèle de crêtes et d'hydrologie généré par la méthode présentée précédemment (cf. le modèle du bas de la figure 2.8), nous considérons seulement les cellules étiquetées *crete* ou *riviere* ; seules ces cellules gardent leur donnée d'altitude, toutes les autres sont remises à zéro. Le MNT-squelette obtenu est alors utilisé en entrée de la méthode par fenêtrage croissant décrite dans **Algorithme 1**. La figure 2.9 montre, de gauche à droite, une progression pas à pas de la méthode : du MNT-squelette obtenu à partir du modèle de la figure 2.8-bas au modèle final.

Partant du MNT-squelette, le modèle final a été généré en approximativement 154 millisecondes sur notre architecture de référence (cf. section

---

11. Dans la réalité naturelle, nos résultats seraient la conséquence immédiate de l'action progressive des dégels des glaciers.

---

**Algorithme 7** Retour sur trace : procédure pour défaire une partie de la trajectoire décrite par une particule simulant l'écoulement d'un cours d'eau.

---

**ENTRÉES:** Une particule  $M^{(i)}$  donnée à un instant  $t$  et se déplaçant sur un MNT  $\mathcal{T}[H][L]$  ; une position à coordonnées entières  $(x_0, y_0)$  marquant le début de la sous trajectoire à défaire.

**SORTIES:** Défaire une partie de la trajectoire décrite par la particule  $M^{(i)}$ .

$n \leftarrow$  le nombre d'éléments dans  $\mathcal{L}(M_t^{(i)})$

$j = n - 1$

**répéter**

$(x, y) \leftarrow \mathcal{X}(\mathcal{L}(M_t^{(i)})[j])$

$\mathcal{T}[y][x].e = \text{inconnu}$

**si** ( $k \leftarrow \text{Affluent}(\mathcal{L}(M_t^{(i)})[j]) \neq -1$ ) **alors**

$\text{Stop}(M_t^{(k)}) \leftarrow \text{faux}$

$\text{Confluent}(M_t^{(k)}) = (-1, -1)$

$m(M_t^{(i)}) \leftarrow m(M_t^{(i)}) - m(M_t^{(k)})$

**fin**

**jusqu'à**  $(x, y) = (x_0, y_0)$

---

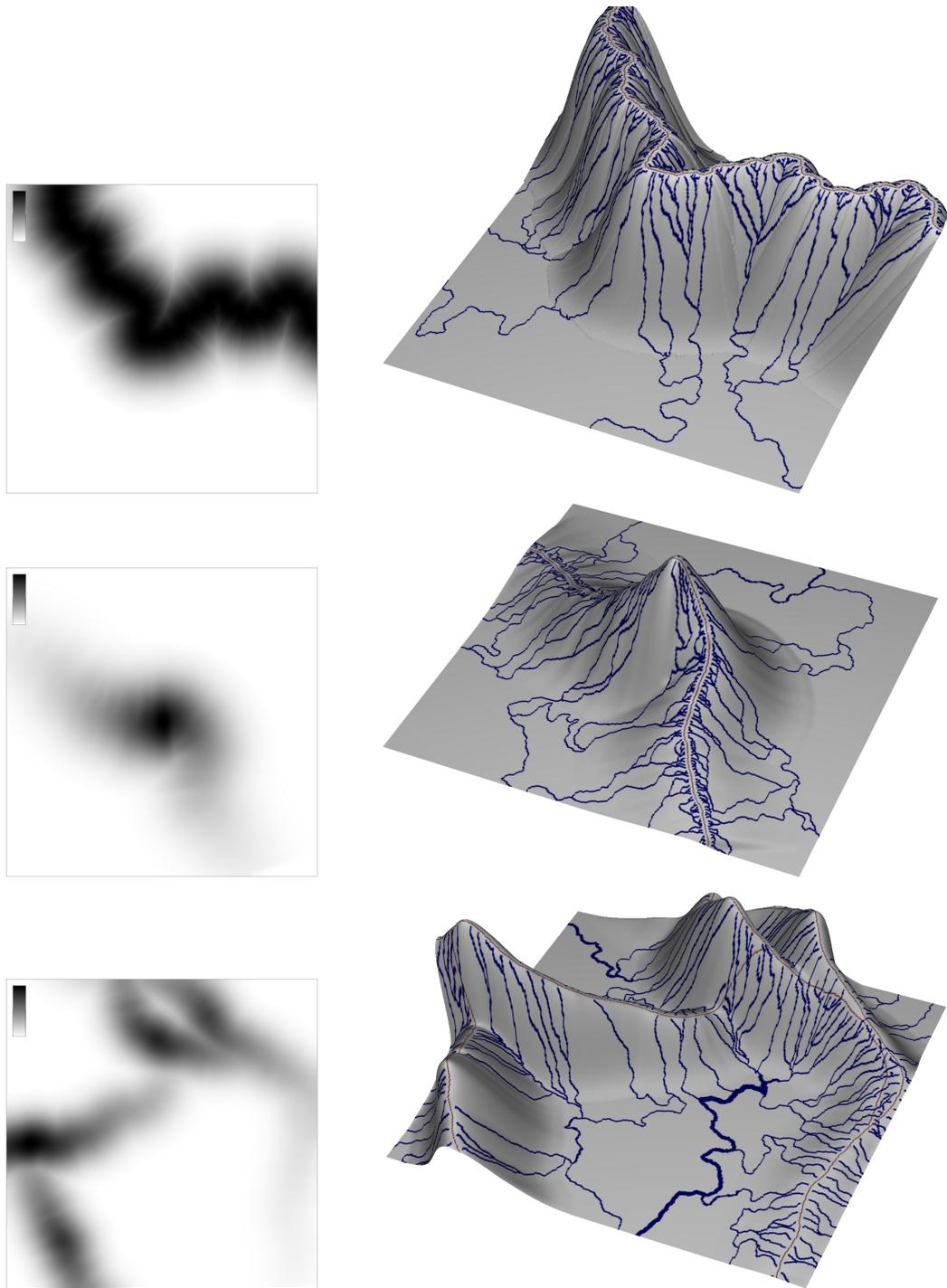


FIG. 2.8 – *Différents modèles de crêtes et hydrologies obtenues avec notre méthode des écoulements.*

---

**Algorithme 8** Validation de la trajectoire d'une particule simulant l'écoulement d'un cours d'eau.

---

**ENTRÉES:** Une particule  $M_t^{(i)}$  simulant l'écoulement d'un cours d'eau sur un MNT  $\mathcal{T}[H][L]$ .

**SORTIES:** Cette fonction  $\text{est\_valide}(M_t^{(i)}, \mathcal{T})$  retourne *vrai* si la trajectoire est validée, *faux* sinon.

**si**  $\text{Valide}(M_t^{(i)}) = \text{vrai}$  **alors**

**RETOURNER** *vrai*

**finsi**

$(x,y) \leftarrow \text{Confluent}(M_t^{(i)})$

**si**  $(x,y) = (-1, -1)$  **alors**

**RETOURNER** *faux*

**finsi**

$j \leftarrow \mathcal{T}[y][x].id$

$\text{Valide}(M_t^{(i)}) \leftarrow \text{est\_valide}(M_t^{(j)}, \mathcal{T})$

**RETOURNER**  $\text{Valide}(M_t^{(i)})$

---



FIG. 2.9 – Progression de la méthode par fenêtrage croissant appliquée à MNT-squelette décrivant des lignes de crête et une hydrologie de bassins versants.

---

**Algorithme 9** Génération d'un réseau fluvial pour un MNT décrivant des bassins hydrographiques (processus principal de simulation d'écoulement).

---

**ENTRÉES:**  $n$  mobiles initialisés et positionnés sur des versants à proximité de lignes crête décrites dans un MNT  $\mathcal{T}[H][L]$ .

**SORTIES:** Calcule un modèle d'hydrologie correspondant au MNT ; inscrit les étiquettes *riviere* pour les cellules appartenant au réseau résultant.

$t \leftarrow 0$

**répéter**

$stop \leftarrow vrai$

**pour tout**  $i \in \{0, \dots, n-1\}$  **faire**

**si**  $Stop(M_t^{(i)}) = vrai$  **alors CONTINUER L'ITÉRATION**

$stop \leftarrow faux$

$intersection \leftarrow \text{deplacer}(M_t^{(i)}, \mathcal{T})$  (cf. **Algorithme 6**)

**si**  $intersection = vrai$  **alors**

$(x, y) \leftarrow \text{Confluent}(M_t^{(i)})$

$j \leftarrow \mathcal{T}[y][x].id$

défaire la trajectoire de  $M_t^{(j)}$  jusqu'à  $(x, y)$  selon **Algorithme 7**

$h \leftarrow$  le nombre d'éléments dans  $\mathcal{L}(M_t^{(j)})$

$Affluent(\mathcal{L}(M_t^{(j)})[h-1]) = i$

$Stop(M_t^{(i)}) \leftarrow vrai$

$S \leftarrow m(M_t^{(i)}) + m(M_t^{(j)})$

$\vec{V}(M_t^{(j)}) \leftarrow \frac{m(M_t^{(j)})}{S} \times \vec{V}(M_t^{(j)}) + \frac{m(M_t^{(i)})}{S} \times \vec{V}(M_t^{(i)})$

$m(M_t^{(j)}) \leftarrow S$

**finsi**

**fin pour**

$t \leftarrow t + dt$

**jusqu'à**  $stop = vrai$

**pour tout**  $i \in \{0, \dots, n-1\}$  **faire**

**si**  $est\_valide(M_t^{(i)}, \mathcal{T}) = faux$  **alors**

défaire la trajectoire de  $M_t^{(i)}$  jusqu'à  $\lfloor \mathcal{X}(M_0^{(i)}) \rfloor$  (**Algorithme 7**)

**finsi**

**fin pour**

**pour tout**  $i \in \{0, \dots, n-1\}$  **faire**

**si**  $Valide(M_t^{(i)}) = vrai$  **alors**

dessiner le cours d'eau correspondant à  $\mathcal{L}(M_t^{(i)})$  : élargir selon les masses successives de la particule et par remplissage de quadrilatères le voisinage des cellules étiquetées *riviere*

**finsi**

**fin pour**

---

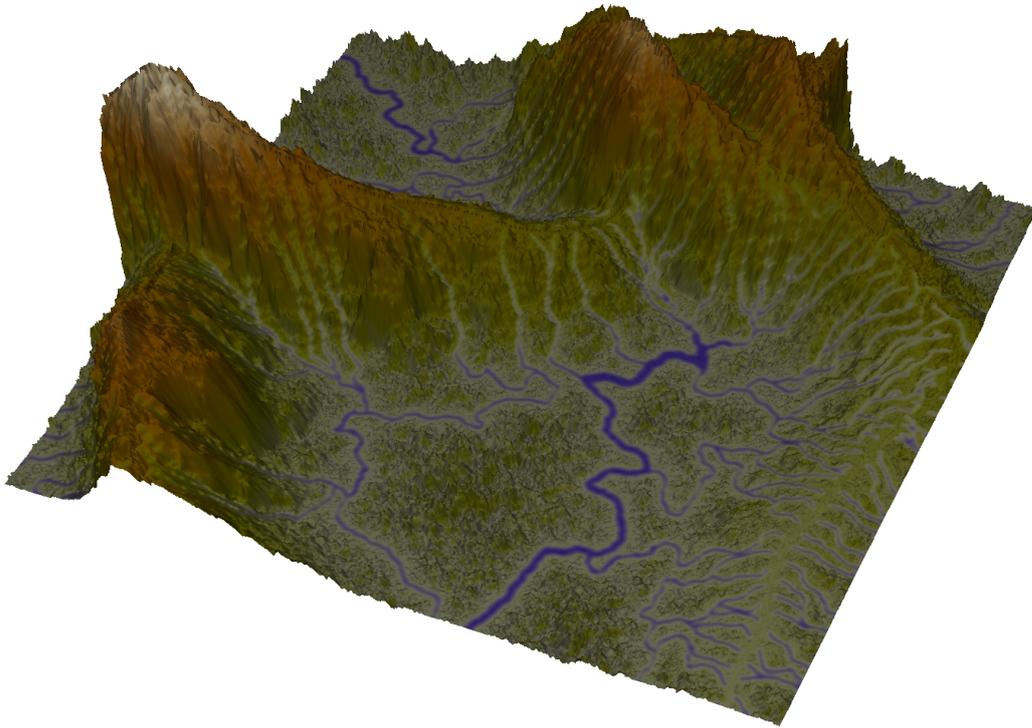


FIG. 2.10 – *Rendu colorié de notre premier modèle de terrain à crêtes et réseau fluvial. La méthode par fenêtrage croissant est appliquée à un MNT-squelette décrivant des lignes de crête et une hydrologie de bassins versants.*

1.1.1.2); soit un total de 641 millisecondes (487 millisecondes pour la modélisation du MNT-squelette) a été requis pour l'ensemble du processus. La figure 2.10 présente un rendu colorié du modèle généré. Un tableau de matériels (i.e. couleurs) est calculé en fonction des pentes et des états des cellules<sup>12</sup>. Des rendus en tracé de rayon sont présentés en section 5.1.

L'ensemble de la méthode décrite ici et amenant à ce résultat a fait l'objet de notre première publication [8].

Nous sommes parvenus à proposer une méthode fractale capable de produire des modèles réalistes de terrain érodé à partir de modèles-squelettes générés par notre méthode des écoulements. Cette dernière affaiblit les contraintes imposées par les lignes de crêtes : le réseau d'écoulement relie les contraintes entre-elles et les rend progressives.

---

12. Ce rendu est réalisé avec *OpenGL* où nous utilisons le *Vertex Buffer Object* [109] pour stocker les positions des sommets, les normales lissées et les couleurs du matériel.

## Chapitre 3

# Le modèle fractal fortement contraint : vers la continuité

Dans le chapitre 2, nous avons présenté notre premier algorithme fractal permettant de générer des modèles de terrain où certaines positions étaient contraintes par des altitudes imposées dans un MNT-squelette. Dans certaines conditions, cette méthode souffrait d'artefacts se présentant sous la forme de ruptures brutales. Ces ruptures sont produites lors du calcul de la jonction entre deux contraintes proches en position et opposées en altitude (cf. section 2.1.3) i.e. contraintes serrées. Les MNT-squelettes produits dans le chapitre précédent apportent une solution dans le cas particulier des modèles de terrain à bassins versants. Mais dans le cas général d'une contrainte quelconque, le traitement local réalisé par la méthode produit toujours les mêmes défauts de jointure.

Nous présentons donc dans ce chapitre notre troisième approche à la génération de modèles fractals de terrain. La méthode proposée aborde, dans sa globalité, le problème de satisfaction de contraintes numériques. Une généralisation de l'approche proposée aboutit à un algorithme appelé *Déplacement des Milieux Morphologiquement Contraint* (DMMC). La spécificité de ce dernier est de contraindre l'allure générale de la surface fractale obtenue tout en respectant les contraintes locales imposées au niveau de cellules du MNT-squelette.

### 3.1 Introduction à l'approche bas-haut-bas

Nous proposons, dans cette partie, une seconde méthode fractale avec contraintes. Ici l'approche au problème de satisfaction des contraintes exprimées sous la forme d'altitudes imposées est globale. Cette méthode corrige

les défauts de jointure apparaissant dans le cas de contraintes opposées (cf. figure 3.3) et produit des modèles équivalents de terrains à crêtes et réseau fluvial (cf. figure 3.5).

### 3.1.1 Notre approche globale

Dans les résultats obtenus figure 1.2 nous notons la possibilité de pose de contraintes le long de la diagonale. Mis à part la contre-diagonale décalée, ces contraintes sont satisfaites par une simple méthode de subdivision comme la *Triangle-Edge* ou la *Diamond-Square* et aucune rupture n'est produite au voisinage des cellules étiquetées non-*inconnu*. Lors du processus de subdivision, les cellules de la diagonale sont rapidement prises en compte et leurs altitudes utilisées pour l'interpolation d'autres cellules voisines à une échelle donnée (les échelles sont décroissantes). Ainsi, plus vite une contrainte intervient lors du processus de subdivision mieux sera prise en compte son altitude pour l'interpolation des cellules voisines. Le problème, extrême, rencontré sur la contre-diagonale s'explique par la profondeur, dans l'arbre sous-jacent au processus de subdivision, à laquelle sont traitées ces cellules. En effet, ces dernières représentent les feuilles de l'arbre et n'interviennent qu'à la fin du processus; leurs altitudes ne sont jamais prises en compte pour les calculs d'interpolation et la rupture est ainsi totale. Le principe de notre nouvelle méthode est de corriger les défauts d'une méthode de déplacement des milieux standard. Ainsi, pour un choix donné de méthode de subdivision, i.e. le processus haut-bas, nous réalisons un prétraitement sur le MNT-squelette, i.e. le processus bas-haut, afin de mieux prendre en considération les altitudes contraintes. Nous proposons un algorithme permettant de remonter, par extrapolation, du bas vers le haut de l'arbre de subdivision, l'information liée aux contraintes d'altitudes. Le processus de subdivision initialement choisi finalise le calcul de l'ensemble de la surface.

#### 3.1.1.1 En une dimension

Nous étudions le cas d'un processus de subdivision dans un espace unidimensionnel. Ici, les altitudes lues aux extrémités d'un intervalle sont utilisées pour interpoler l'altitude du milieu, soit la position médiane. Le processus est répété pour les deux sous intervalles obtenus. TAB.3.1-(a) donne l'ordre haut-bas dans lequel sont interpolées les positions d'un intervalle à 17 éléments<sup>1</sup>. Dans chaque cas, le premier chiffre indique la profondeur à laquelle l'indice apparaît dans l'arbre de subdivision. Les deux autres nombres informent de l'indice des nœuds parents, i.e. les indices des altitudes utilisées

---

1. Le nombre d'éléments prend la forme  $2^n + 1$  afin de faciliter l'exposé de la démarche.

pour interpoler la position. Ces indices sont positionnés au couple  $(-1, -1)$  pour des positions n'ayant pas de parents, i.e. les positions extrêmes de l'intervalle. Notons que la profondeur  $n$  apparaît aux positions impaires dans le cas d'un intervalle à  $2^n + 1$  éléments. En règle générale, pour cette forme de dimensions, nous pouvons directement extraire la profondeur à laquelle la position apparaîtra lors du processus de subdivision. Ainsi, pour un indice  $i$ , nous posons :

$$\begin{cases} i \neq 0 \Rightarrow \mathbb{A}_i \subset \{0, \dots, n\} \text{ tel que } i = \sum_{k \in \mathbb{A}_i} 2^k \\ i = 0 \Rightarrow \mathbb{A}_i = \{n\} \end{cases} \quad (3.1)$$

alors la profondeur à laquelle intervient l'indice  $i$  est donnée par :

$$p = n - \min \mathbb{A}_i \quad (3.2)$$

et les parents, respectivement gauche et droit, de la position sont :

$$\begin{cases} \min \mathbb{A}_i < n : & (i - 2^{\min \mathbb{A}_i}, i + 2^{\min \mathbb{A}_i}) \\ \text{sinon} : & (-1, -1) \end{cases} \quad (3.3)$$

Soit, lors d'un processus bas-haut, pour un intervalle où différentes altitudes sont préfixées, nous calculons pour chacune les indices des parents. Si ces indices sont valides (non négatifs) alors : pour chaque parent si la position ne contient pas de contrainte alors fixer l'altitude en l'extrapolant par rapport à l'altitude de la position de l'enfant (il est possible d'aboutir à la situation où plusieurs enfants extrapolent l'altitude d'un même parent). Nous répétons le processus pour les nouvelles positions fixées (tant qu'il y en a). Enfin, reprendre un processus haut-bas pour finaliser l'interpolation.

TAB.3.1-(b) montre l'ordre d'extrapolation et d'interpolation obtenu par un tel procédé. Ici, seule la position à l'indice 7 est imposée, 8 et 6 sont extrapolées par 7, 4 est extrapolée par 6 et 0 et 16 par 8 puis les subdivisions haut-bas sont réalisées, i.e. sans modifier les positions préfixées ou extrapolées. Notons qu'en préfixant, par exemple, la valeur en position d'indice 6, 8 et 4 seraient les premières positions extrapolées, elles sont suivies de 0 et 16. Dans ce dernier cas, l'altitude de la position 0 peut être extrapolée par 4 ou 8 ou les deux. Une première méthode d'extrapolation est proposée en section 3.1.1.3 ; l'altitude extrapolée est telle que calculée dans [9]. Nous y proposons une amélioration du calcul d'extrapolation. Ce calcul est donné en section 3.2 ; l'altitude des nœuds enfants est pondérée par un calcul de barycentre.

La méthode est généralisée à toutes les subdivisions d'intervalles ayant une taille quelconque. Il suffit de simuler (i.e. réaliser le processus de subdivision sans effectuer d'interpolation) un processus de subdivision pour obtenir

pour chaque position les indices des nœuds parents<sup>2</sup>.

La figure 3.1 montre les résultats obtenus sans, puis avec, notre nouvelle approche. Dans un intervalle quelconque, nous fixons aléatoirement plusieurs altitudes. Le graphique (a) montre la courbe obtenue en utilisant seulement un processus de subdivision haut-bas sans déplacement des milieux. Les contraintes initiales forment des pics de discontinuité ; aucune n'a été utilisée lors de l'interpolation ; toutes les autres positions sont à zéro. Le graphique (b) illustre le même processus mais cette fois avec déplacement des milieux ; la courbe hors contraintes a une allure fractale homogène et les pics de discontinuité sont toujours là (sauf pour la dernière contrainte). Enfin, pour les graphiques figure 3.1-(c) et (d), nous appliquons, en prétraitement, le processus bas-haut suivi du processus haut-bas sans déplacement des milieux pour (c) et avec déplacement des milieux pour (d). Les barres verticales colorées en vert expriment les altitudes extrapolées au cours du processus bas-haut. Les pics de discontinuité ont tous disparus. Dans cet exemple, nous extrapolons l'altitude du nœud parent selon les altitudes pondérées par la distance des nœuds enfants (cf. la généralisation de l'approche en section 3.2).

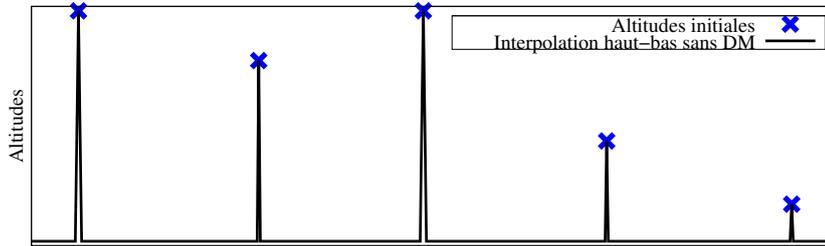
### 3.1.1.2 En deux dimensions

Dans un espace bidimensionnel, notre approche consiste à enrichir une donnée initiale exprimée sous la forme d'un MNT-squelette afin de satisfaire les dépendances enfants-parents requises par le processus de subdivision interpolant la surface. Un nœud parent doit exprimer une altitude et un état *non-inconnu* si au minimum un de ses nœuds enfants est *non-inconnu*. Cette condition est ainsi remplie par un processus bas-haut réalisant un pré-calcul au niveau du MNT-squelette et fixant par extrapolation les altitudes aux cellules correspondantes aux nœuds parents. Soit ce processus peut être considéré comme la fonction inverse du processus de subdivision réalisant les interpolations. Il est donc essentiel de connaître, à l'avance, quelle méthode de subdivision est choisie.

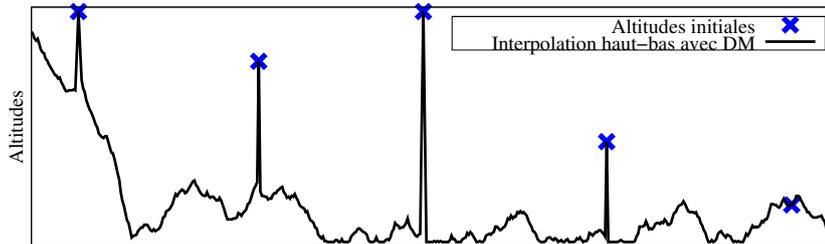
Ici, nous nous limitons aux deux méthodes de subdivision précédemment choisies (pour des raisons annoncées en section 1.1.2), c'est-à-dire la subdivision en *Triangle-Edge* et la subdivision en *Diamond-Square*. Ainsi, un milieu  $a$ , selon la méthode de subdivision, deux, trois ou quatre nœuds parents.

---

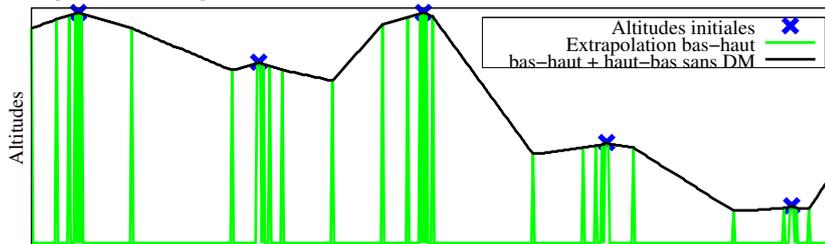
2. Notre première implémentation de la méthode stockait l'ensemble de l'arbre décrivant les relations parents-enfants. Cela permettait un très léger gain de performances dans le cas de MNTs de petites ou moyennes dimensions, par exemple jusqu'à un million de cellules dans le cas de notre architecture de référence. Cette situation s'inverse très vite avec des MNTs de plus grandes dimensions. Ainsi, nous considérons plus optimale la solution simulant une subdivision pour retourner les indices des parents d'une position.



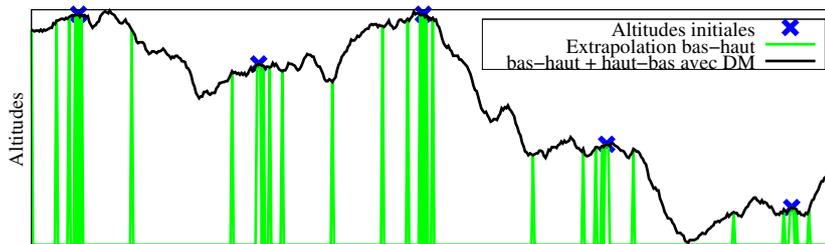
(a) Subdivision standard sans déplacement des milieux : l'intervalle initial est coupé au milieu en deux sous-intervalles, le milieu prend l'altitude moyenne des extrema, et l'opération est répétée récursivement sur les sous-intervalles.



(b) Subdivision standard avec déplacement des milieux : l'intervalle initial est coupé au milieu en deux sous-intervalles, le milieu prend l'altitude moyenne des extrema plus un déplacement vertical proportionnel à la taille de l'intervalle, et l'opération est répétée récursivement sur les sous-intervalles.



(c) Prétraitement bas-haut des contraintes initiales : des altitudes (les barres verticales vertes) sont ajoutées (extrapolées) à des positions clés de l'intervalle ; ces positions sont obtenues en parcourant l'arbre de subdivision en sens inverse i.e. bas-haut. Enfin, une subdivision standard sans déplacement des milieux est réalisée pour l'intervalle ainsi enrichi en altitudes imposées.



(d) Prétraitement bas-haut des contraintes initiales : des altitudes (les barres verticales vertes) sont ajoutées (extrapolées) à des positions clés de l'intervalle ; ces positions sont obtenues en parcourant l'arbre de subdivision en sens inverse i.e. bas-haut. Enfin, une subdivision standard avec déplacement des milieux est réalisée pour l'intervalle ainsi enrichi en altitudes imposées.

FIG. 3.1 – Application de notre approche bas-haut-bas à une courbe unidimensionnelle où différentes positions sont préfixées. Le précalcul réalisé par le processus bas-haut règle les problèmes de discontinuité tout en satisfaisant les contraintes d'altitudes imposées.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	4	3	4	2	4	3	4	1	4	3	4	2	4	3	4	0
-1,-1	0,2	0,4	2,4	0,8	4,6	4,8	6,8	0,16	8,10	8,12	10,12	8,16	12,14	12,16	14,16	-1,-1

(a)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	5	4	5	2	5	1	0	1	5	4	5	3	5	4	5	2
8	0,2	0,4	2,4	6	4,6	7	-1,-1	7	8,10	8,12	10,12	8,16	12,14	12,16	14,16	8

(b)

TAB. 3.1 – Ordres d'utilisation des indices lors d'interpolations. En haut, une méthode de subdivision standard réalise un parcours haut-bas de l'arbre sous-jacent au processus récursif d'interpolation ; la valeur obtenue est liée à la profondeur à laquelle apparaît l'indice de la position dans l'arbre. En bas, l'ordre d'apparition des positions est modifié par un premier parcours bas-haut de l'arbre réalisant des extrapolations ; ces extrapolations sont générées par la position à l'indice 7. La suite 0,4,3,4,2,4,... représente donc les profondeurs des positions 0,1,2,3,4,5,... dans l'arbre des subdivisions.

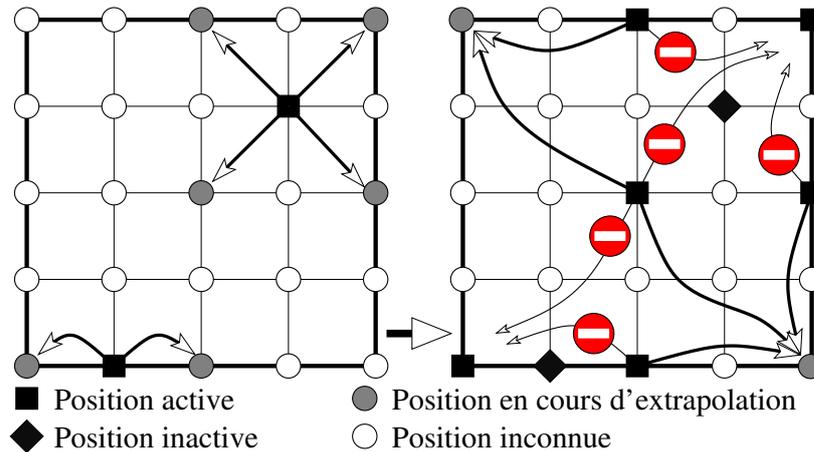


FIG. 3.2 – Exemple d'un processus Triangle-Edge-bas-haut appliqué dans le cas d'un MNT  $5 \times 5$  où les positions (3,1) et (1,4) sont contraintes. Nous obtenons un total de huit positions extrapolées.

Chaque nœud enfant possédant une étiquette non-*inconnu* contribue au calcul par extrapolation de l'altitude d'un ou plusieurs de ses parents étiquetés *inconnu*. Cette opération est effectuée à l'initialisation pour toutes les cellules du MNT-squelette ayant un état non-*inconnu* et des nœuds parents<sup>3</sup>. Nous obtenons<sup>4</sup> ainsi de nouvelles cellules étiquetées *connu*. Ces cellules peuvent avoir des nœuds parents dont l'état est *inconnu*. Soit l'opération doit être répétée tant que, d'un tour à l'autre, le processus produit de nouvelles positions étiquetées *connu*. Le schémas figure 3.2 montre les deux tours du processus bas-haut d'une subdivision en *Triangle-Edge*; ce processus est réalisé pour un MNT-squelette où les cellules en coordonnées (3,1) et (1,4) ont été contraintes. Ainsi, au premier tour (i.e. à l'initialisation), le MNT est parcouru à la recherche de positions où les cellules sont étiquetées non-*inconnu*. Chacune est insérée dans une *File de Positions Actives* notée  $\mathcal{FPA}$ . Soit, pour toutes les positions défilées de  $\mathcal{FPA}$  nous extrapolons en même temps les altitudes des nœuds parents dont l'état est *inconnu*. Une position défilée devient une position inactive. Chaque position correspondant à un nœud parent extrapolé (i.e. les positions dénotées *position en cours d'extrapolation* dans figure 3.2) est alors enfilée dans  $\mathcal{FPA}$  et sera considérée comme une position active du prochain tour. Nous répétons le processus tant que, d'un tour à l'autre,  $\mathcal{FPA}$  est non vide. Ce précalcul effectué, une interpolation par la méthode choisie est réalisée et produit la surface finale. Une surface fractale est obtenue en appliquant un déplacement des milieux.

### 3.1.1.3 Notre algorithme bas-haut-bas

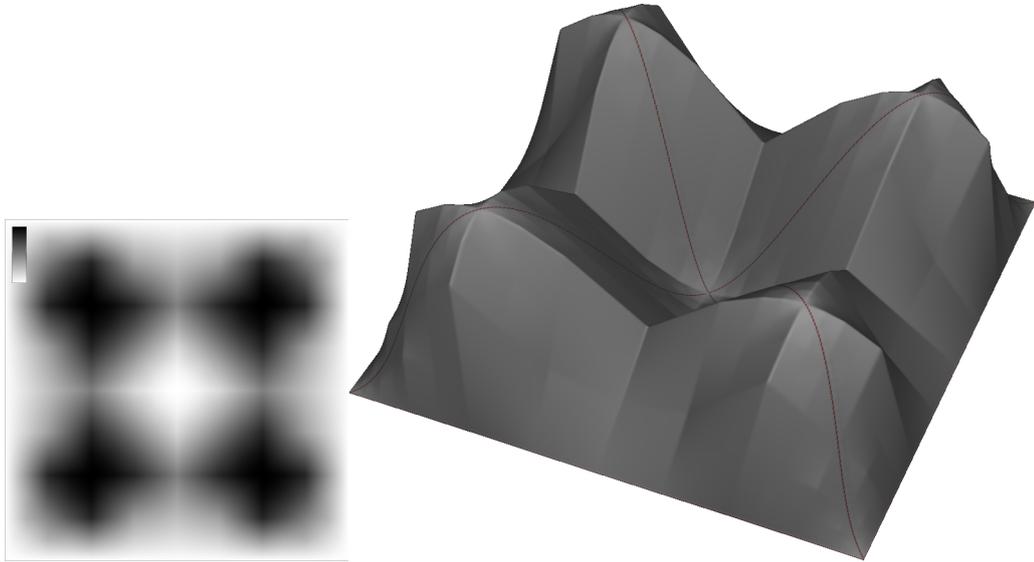
Avant de donner le détail de l'algorithme, nous définissons :

- $\mathcal{T}[H][L]$  le MNT-squelette donné ;
- $\mathcal{E}x[H][L]$  un tableau stockant pour chaque position  $(x,y)$  un triplé de valeurs  $\mathcal{E}x[H][L].a$  et  $\mathcal{E}x[H][L].n$  deux entiers respectivement signé et non signé sur 32 bits et  $\mathcal{E}x[H][L].d$  un flottant sur 32 bits ; les éléments du tableau sont initialisés à zéro ;
- $\mathbf{dm}(\mathcal{T}, s, t, \mathcal{H})$  la méthode de subdivision choisie (i.e. *Triangle-Edge* ou *Diamond-Square*) ; cf. équation 1.1 pour  $s, t$  et  $\mathcal{H}$  ;
- $\mathbf{simu\_dm}(L, H, (e_x, e_y))$  recherche, en simulant la subdivision, le nœud enfant en position  $(e_x, e_y)$  et renvoie la liste des positions de ses deux, trois ou quatre parents ;

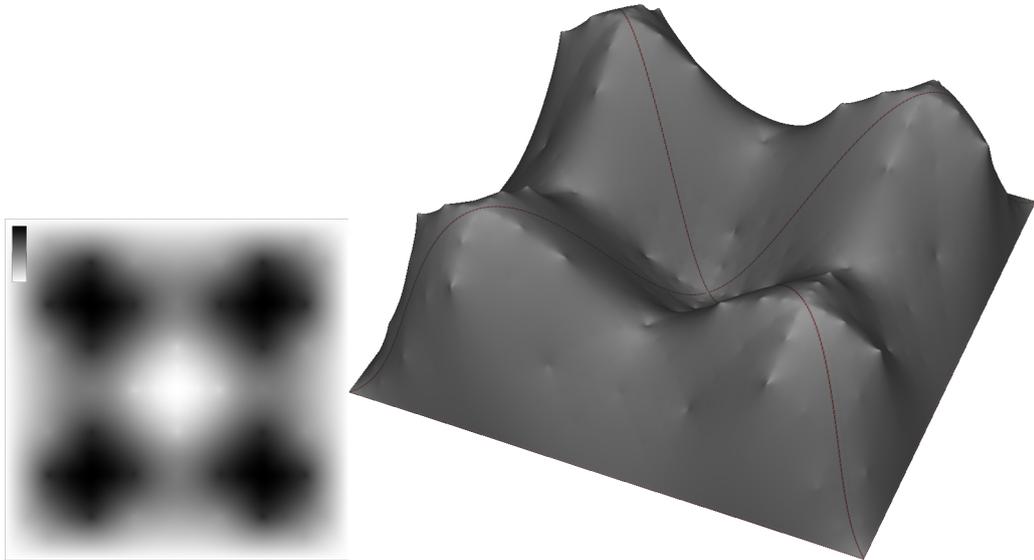
---

3. Dans le cadre des méthodes de subdivision choisies, seules les quatre cellules des coins du MNT ne possèdent pas de nœuds parents.

4. Aucune position *connu* n'est produite dans le cas particulier où pour toutes les contraintes les plus basses d'un sous-arbre, toutes les cellules rencontrées en remontant le sous-arbre sont aussi contraintes.



Modèle obtenu par l'approche bas-haut-bas de la subdivision en *Triangle-Edge*



Modèle obtenu par l'approche bas-haut-bas de la subdivision en *Diamond-Square*

FIG. 3.3 – L'approche bas-haut-bas appliquée au modèle de contraintes en croix avec contre-diagonale décalée où les bords ont été contraints à zéro.

- $X$  un nombre pseudo-aléatoire compris dans l'intervalle  $[-1,1]$ ;
- Nous utilisons une file  $\mathcal{FPA}$  pour stocker des coordonnées de cellules. Nous définissons également les fonctions suivantes :
  - `nb_elts( $\mathcal{F}$ )` cette fonction retourne le nombre d'éléments stockés dans la file  $\mathcal{F}$ ;
  - `enfiler( $\mathcal{F}$ ,  $(x,y)$ )` cette fonction ajoute la coordonnée  $(x,y)$  en fin de file  $\mathcal{F}$ ;
  - `defiler( $\mathcal{F}$ )` cette fonction extrait la première coordonnée de la file  $\mathcal{F}$ ;
  - `elt_file( $\mathcal{F}$ ,  $n$ )` cette fonction retourne une copie de la  $n^{ieme}$  coordonnée de la file  $\mathcal{F}$ ; i.e. laisse l'élément dans la file;

Ainsi, figure 3.3 montre l'application de notre nouvelle approche dans le cas du modèle-squelette en croix avec contre-diagonale décalée où les bords ont été contraints à zéro. Notre nouvelle approche, qu'elle soit appliquée à la méthode de subdivision *Triangle-Edge* ou *Diamond-Square*, ne souffre pas des défauts de la méthode par fenêtrage croissant; comme illustré sur la figure 2.3, cette dernière produit des ruptures au niveau des jonctions entre contraintes opposées.

### 3.1.2 Second modèle de terrain à crêtes et réseau fluvial

Pour notre second modèle réaliste de terrain à crêtes et réseau fluvial, nous appliquons l'approche bas-haut-bas au même MNT-squelette utilisé, en section 2.2.3, afin de produire notre premier modèle. Ainsi, la figure 3.4 montre différentes phases de la génération complètement automatique du MNT. Pour cet exemple, la méthode de subdivision choisie est, comme pour le premier modèle, la *Triangle-Edge*. La figure 3.4-(a) montre le MNT-squelette utilisé en entrée de l'**Algorithme 10**. La figure 3.4-(b) est obtenue après l'application du processus bas-haut; le MNT-squelette est ainsi enrichi par l'ajout de nouvelles contraintes. Finalement le processus haut-bas termine de générer l'ensemble de la surface fractale et produit le MNT de la figure 3.4-(d). La figure 3.4-(c) illustre le résultat d'une application directe du processus haut-bas au MNT-squelette de la figure 3.4-(a); la netteté des contours des crêtes et du réseau fluvial dénote les ruptures produites. Enfin, la figure 3.5 montre différentes vues 3D de phases de modélisation. Du point de vue des performances (cf. section 1.1.1.2), les temps de calcul obtenus par cette nouvelle méthode sont comparables à ceux de la méthode par fenêtrage croissant: pour le même MNT-squelette, nous générons le modèle final en approximativement 100 millisecondes.

---

**Algorithme 10** Déplacement des milieux, l'approche bas-haut-bas.

---

**ENTRÉES:** Les éléments définis en début de section (i.e. section 3.1.1.3).

**SORTIES:** le MNT  $\mathcal{T}$  complété.

```

pour tout  $y \in \{0,1,2,\dots,H-1\}$  faire
  pour tout  $x \in \{0,1,2,\dots,L-1\}$  faire
    si  $\mathcal{T}[y][x].e \neq \text{inconnu}$  alors  $\text{enfiler}(\mathcal{FPA}, (x,y))$ 
  fin pour
fin pour
tantque  $(n \leftarrow \text{nb\_elts}(\mathcal{FPA})) \neq 0$  faire
  pour tout  $i \in \{1,2,\dots,n\}$  faire
     $(e_x, e_y) \leftarrow \text{defiler}(\mathcal{FPA})$ 
    pour tout  $(p_x, p_y) \in \text{simu\_dm}(L, H, (e_x, e_y))$  faire
      si  $\mathcal{T}[p_y][p_x].e \neq \text{inconnu}$  alors CONTINUER
       $\mathcal{E}x[p_y][p_x].a \leftarrow \mathcal{E}x[p_y][p_y].a + \mathcal{T}[e_y][e_x].a$ 
       $\mathcal{E}x[p_y][p_x].n \leftarrow \mathcal{E}x[p_y][p_y].n + 1$ 
       $\mathcal{E}x[p_y][p_x].d \leftarrow \mathcal{E}x[p_y][p_y].d + \sqrt{(p_x - e_x)^2 + (p_y - e_y)^2}$ 
       $\text{enfiler}(\mathcal{FPA}, (p_x, p_y))$ 
    fin pour
  fin pour
   $n \leftarrow \text{nb\_elts}(\mathcal{FPA})$ 
  pour tout  $i \in \{1,2,\dots,n\}$  faire
     $(p_x, p_y) \leftarrow \text{elt\_file}(\mathcal{FPA}, i)$ 
     $\mathcal{T}[p_y][p_x].a \leftarrow \frac{\mathcal{E}x[p_y][p_x].a}{\mathcal{E}x[p_y][p_x].n} + \frac{\mathcal{E}x[p_y][p_x].d \times s}{\mathcal{E}x[p_y][p_x].n} \times X$ 
     $\mathcal{T}[p_y][p_x].e \leftarrow \text{connu}$ 
  fin pour
fin tantque
 $\text{dm}(\mathcal{T}, s, t, \mathcal{H})$ 

```

---

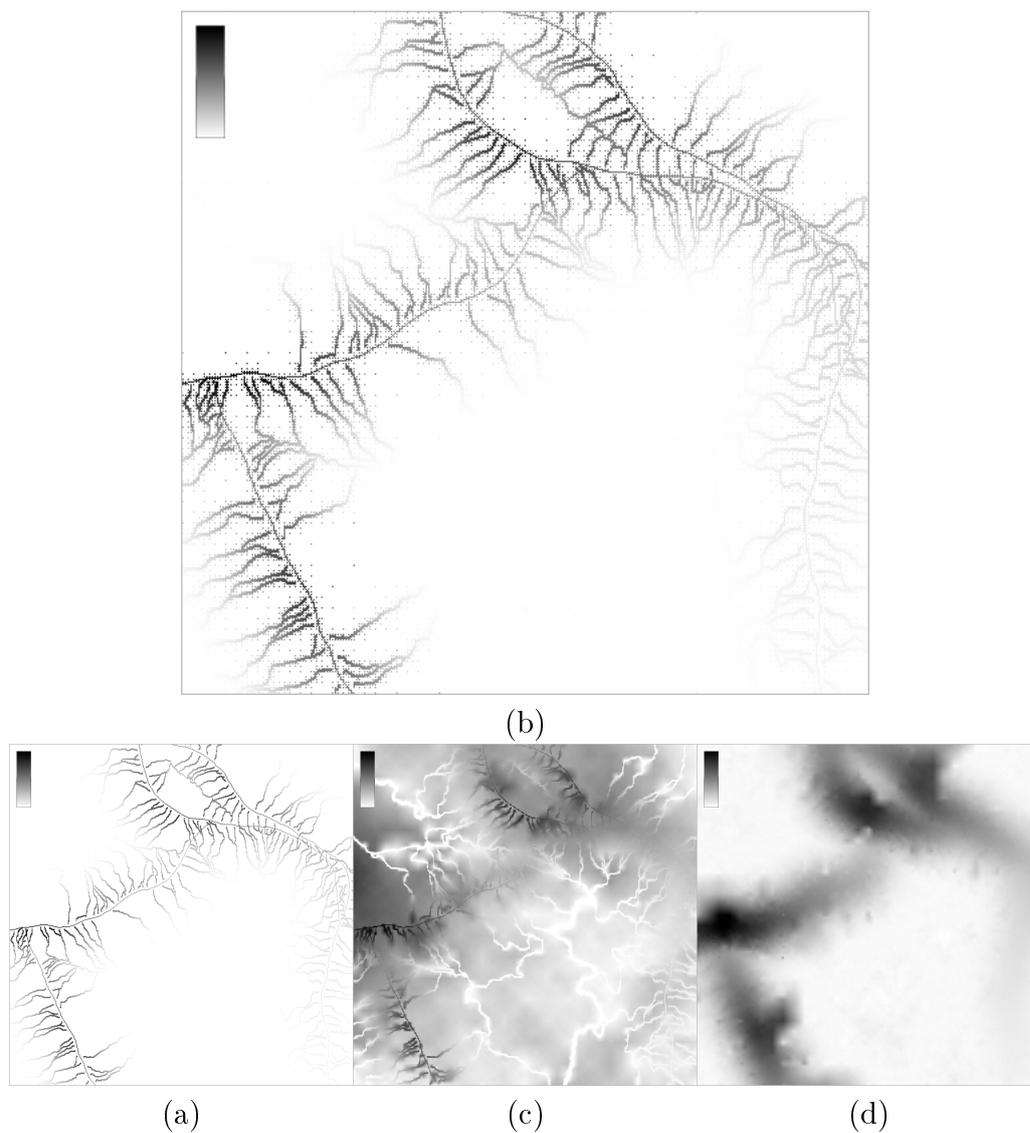


FIG. 3.4 – MNTs illustrant différentes phases de la génération du modèle de terrain: (a) le MNT-squelette (b) le MNT obtenu après le processus bas-haut (c) le MNT obtenu après (seulement) le processus haut-bas (d) le MNT final obtenu après l'ensemble du processus (i.e. bas-haut-bas).

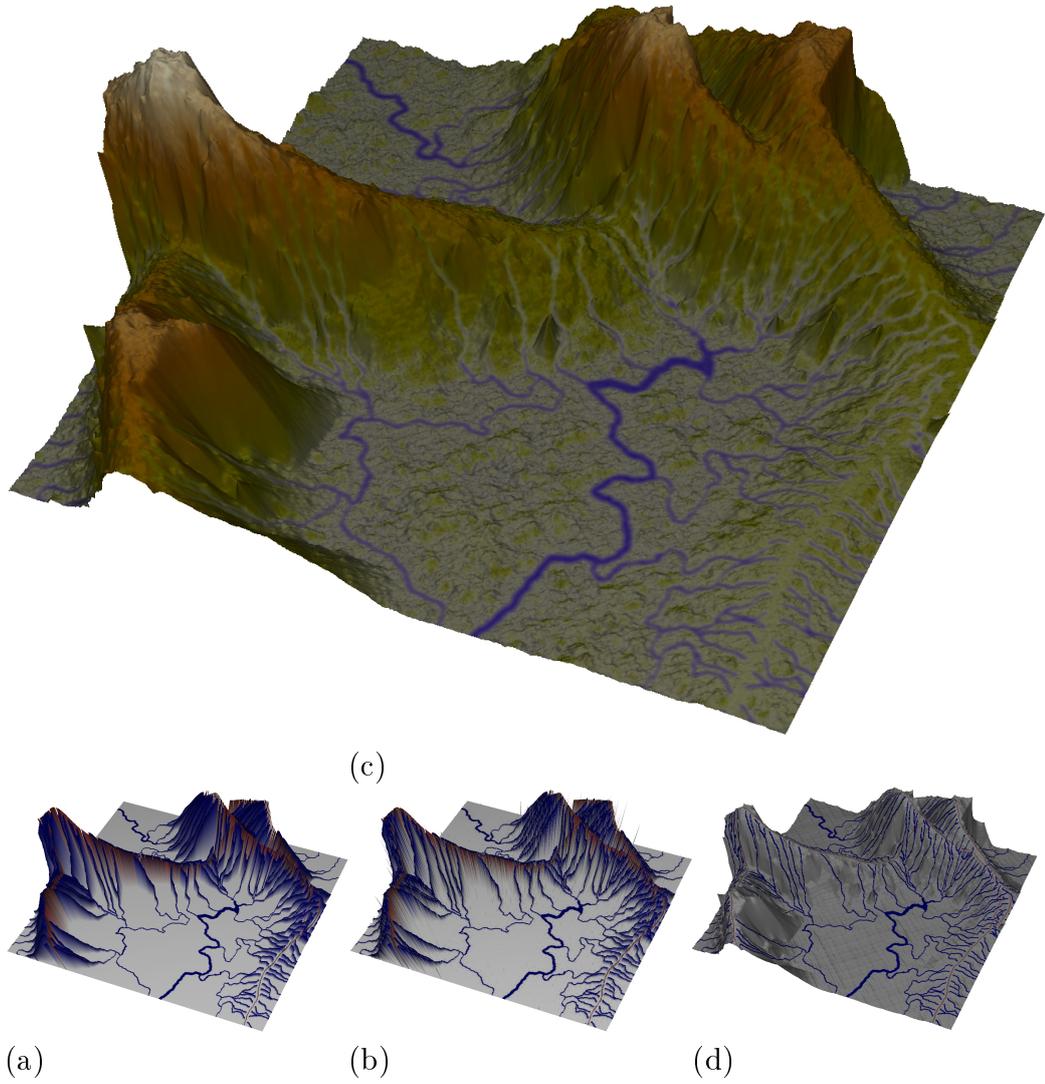


FIG. 3.5 – Vues 3D des différentes phases du processus bas-haut-bas : (a) le MNT-squelette (b) le MNT obtenu après le processus bas-haut (c)/(d) le MNT final obtenu avec/sans déplacement des milieux..

Notre nouvelle approche, apporte une meilleure gestion des contraintes locales (cf. le premier exemple donné en figure 3.3) et apporte plus de souplesse concernant la répartition des contraintes imposées. Nous utilisons un second modèle de MNT-squelette érodé pour illustrer notre premier pas vers la personnalisation des modèles générés. Le modèle obtenu en figure 3.6 est réalisé à partir d'une courbe de cercle vectoriel. La courbe vectorielle donnée en entrée est traduite en ligne de crête avec bassins versants et l'hydrologie correspondante est calculée. Ainsi nous obtenons le MNT-squelette dont une vue tridimensionnelle est donnée en haut à gauche de la figure. Le couple de processus bas-haut et haut-bas produisent le modèle au rendu coloré donné en bas de la figure. Ce modèle est généré en 111 millisecondes où la méthode de subdivision choisie est la *Diamond-Square*.

### 3.2 La généralisation de l'approche : vers le contrôle de la forme

La méthode proposée en section 3.1 présente l'inconvénient de ne pas pouvoir paramétrer l'allure générale d'une surface. Soit, hormis les cellules contraintes, le reste de la surface est créé par extrapolation moyennant des altitudes imposées puis par une composition d'interpolations appliquée récursivement. Ces deux parties peuvent être réalisées avec ou sans déplacements. Nous commençons par modifier les calculs d'interpolation donnés dans l'équation 1.1 et utilisés dans tous les précédents résultats. Ainsi, nous rendons paramétrable la courbe d'interpolation. Nous montrons, pour un processus haut-bas, un exemple où l'allure de la surface est modifiée pour un MNT-squelette contraint au centre et aux coins. Puis nous proposons une méthode équivalente dans le cas du processus bas-haut, soit le calcul d'extrapolation. Précédemment, une altitude de nœud parent est extrapolée par une moyenne des altitudes éventuellement déplacée ; ce déplacement est lui même proportionnel à la moyenne des distances. Nous proposons une autre approximation où chaque apport de nœud enfant est modifié selon une courbe d'extrapolation puis pondéré par la distance le séparant du nœud parent. Un nouvel algorithme, appelé le *Déplacement des Milieux Morphologiquement Contraint* (DMMC), est proposé puis développé en section 3.2.3 dans l'**Algorithme 11** ; nous donnons le détail de son processus bas-haut. Enfin, au travers de l'exemple du MNT-squelette en croix avec contre-diagonale décalée et bords à zéro, nous montrons la variété de surfaces obtenues à l'aide de cette généralisation de l'approche.

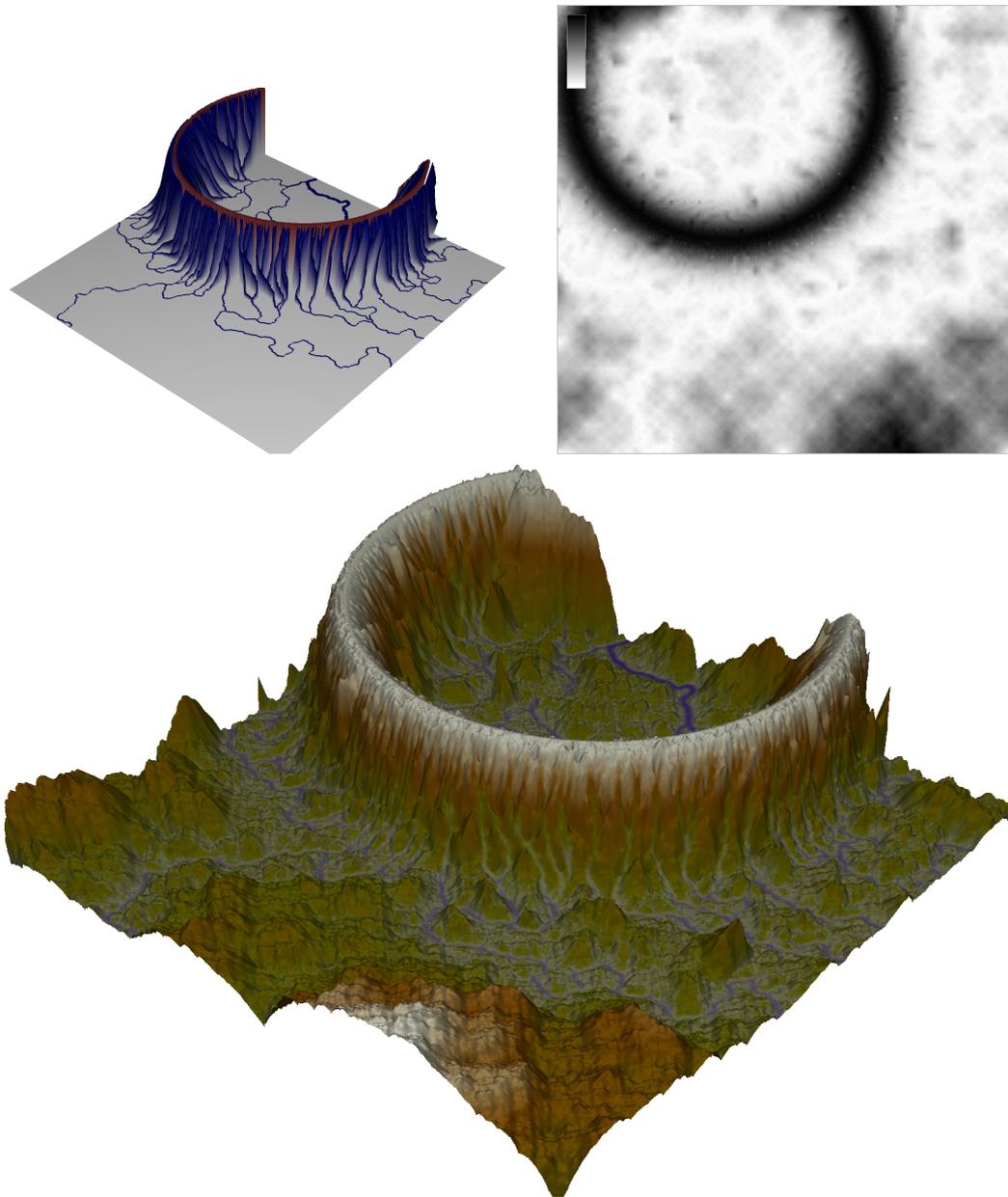


FIG. 3.6 – *Modèle de terrain spéculatif*: un squelette de crête est décrit par un cercle donné sous forme vectorielle ; nous modélisons les bassins versants afin de calculer l'hydrologie de la surface. Nous obtenons, en 1089 millisecondes (l'architecture de référence est décrite en section 1.1.1.2), le squelette composé de la ligne de crête et du réseau fluvial (image en haut à gauche) puis nous générons, en 111 millisecondes, le terrain final à l'aide de notre algorithme fractal.

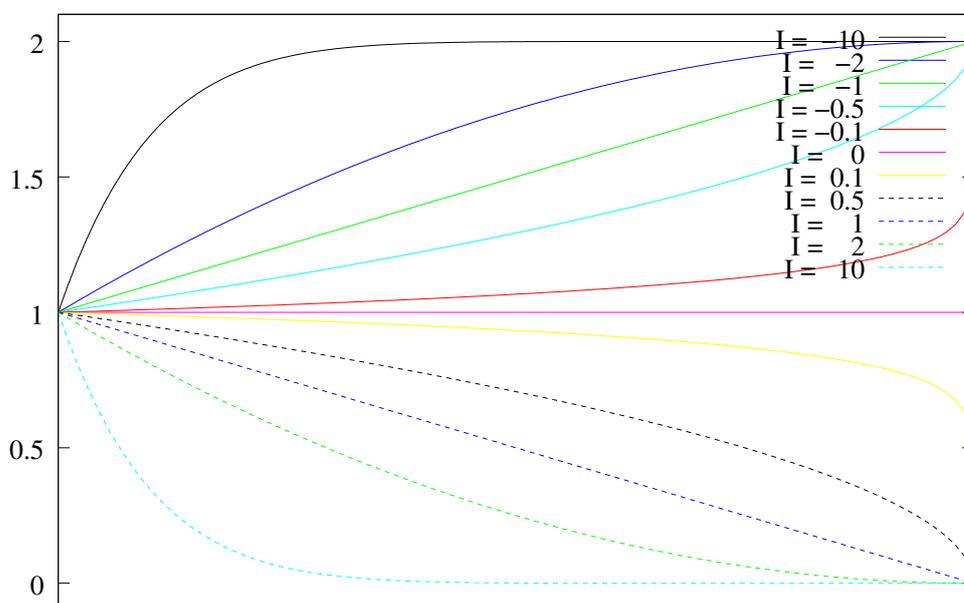


FIG. 3.7 – Variations de la courbe d'interpolation selon les variations du paramètre  $I$ . Pour  $I$  constant, nous obtenons la pondération de l'altitude moyenne en fonction de la distance euclidienne entre le(s) nœud(s) interpolant(s) et le nœud interpolé.

### 3.2.1 La modification de la courbe d'interpolation

Notre démarche est de proposer un contrôle au niveau de l'allure de la surface générée tout en gardant la possibilité de contraindre des altitudes via un MNT-squelette. Ainsi, nous proposons de modifier le calcul d'interpolation en le rendant, au besoin, non-linéaire.

Lors d'un processus haut-bas, une méthode de subdivision calcule, pour un milieu donné (i.e. soit un nœud enfant), son altitude à partir de la moyenne des altitudes des nœuds parents. Ce calcul donne, selon la méthode de subdivision choisie, une surface à l'allure propre à cette même méthode de subdivision (cf. figure 1.1). Ainsi la courbe obtenue par une subdivision en *Triangle-Edge* est linéaire, celle de la *Diamond-Square* est cubique et possède un point d'inflexion. Nous choisissons d'utiliser une courbe paramétrable pour pondérer la valeur de la moyenne calculée. La courbe doit pouvoir être stable, croissante, ou décroissante, linéaire ou non-linéaire. Nous posons la pondération variable par rapport aux distances euclidiennes calculées entre un nœud enfant et ces deux, trois ou quatre parents. Ces distances sont égales lors d'une subdivision en *Triangle-Edge* ou *Diamond-Square*. Soit pour une cellule à interpoler (i.e. nœud enfant), une altitude  $a$  représentant la moyenne

des altitudes des cellules des nœuds parents tous à une distance  $d$  du nœud enfant, nous calculons l'altitude de la cellule à interpoler  $f(a,d)$  telle que :

$$f(a,d) = a \times (1 - \sigma(I) \times (1 - (1 - \frac{d}{d_{max}})^{|I|}))$$

$$\sigma(I) = \begin{cases} 1 & I \geq 0 \\ -1 & I < 0 \end{cases} \quad (3.4)$$

où  $d_{max}$  est la distance euclidienne maximale mesurée pour un MNT  $\mathcal{T}[H][L]$  donné ; soit  $\sqrt{L^2 + H^2}$  la longueur de la diagonale.  $I$  sert à contrôler le comportement de la courbe. Ainsi nous obtenons, selon les valeurs prises par  $I$  :  $f(a,d) = a$  pour  $I = 0$ , sinon  $f(a,d)$  est respectivement décroissante ou croissante en fonction de la distance euclidienne  $d$  et pour  $I > 0$  ou  $I < 0$ . L'allure de la courbe est respectivement bombée, linéaire ou creuse selon  $0 < |I| < 1$ ,  $|I| = 1$  ou  $|I| > 1$ . La figure 3.7 illustre les variations de la courbe d'interpolation selon les variations du paramètre  $I$  et de la distance euclidienne.

Maintenant, pour  $s$ ,  $t$  et  $\mathcal{H}$  donnés, nous ajoutons, à ce calcul d'interpolation, un déplacement  $d \times \delta$  où  $s$ ,  $t$ ,  $\mathcal{H}$  et  $\delta$  sont décrits par l'équation 1.1. Nous obtenons une surface fractale pour un déplacement non nul ( $\mathcal{H} < 1$  et  $s > 0$ ). La figure 3.8 montre<sup>5</sup> les variations de la surface en fonction de l'ensemble des paramètres, pour une subdivision en *Triangle-Edge* et un MNT-squelette où la cellule au centre est positionnée à une altitude mi-haute et celles des coins à la même altitude mi-basse. Notons que pour un  $s$  nul,  $t$  et  $\mathcal{H}$  n'ont aucun effet sur la surface obtenue ; dans ce cas, ces paramètres n'apparaissent pas dans la légende.

### 3.2.2 La modification du calcul d'extrapolation

La même transformation est réalisée dans le cas du processus bas-haut ; elle permet de pondérer la valeur extrapolée en fonction de la valeur du paramètre  $I$  et d'une distance. Nous séparons les deux transformations de manière à pouvoir contrôler la forme générale de l'extrapolation sans pour autant modifier l'allure de l'interpolation. Nous donnons des exemples où seule l'extrapolation est pondérée par des valeurs variables en rapport à la distance et où la pondération de l'interpolation est constante (égale à 1 quelque soit la distance, soit pour  $I = 0$ ) : les images de la figure 3.3 sont à comparer aux éléments (a),(b),(g) et (h) de la figure 3.9 ; les deux modèles sont respectivement générés par un DMMC en *Triangle-Edge* et un DMMC en *Diamond-Square*

---

5. Nous pouvons obtenir des surfaces telles que le Mont de Takagi [6, 89] sans effectuer de déplacement des milieux : il suffit de poser  $I = 1$  et  $s = 0$ .

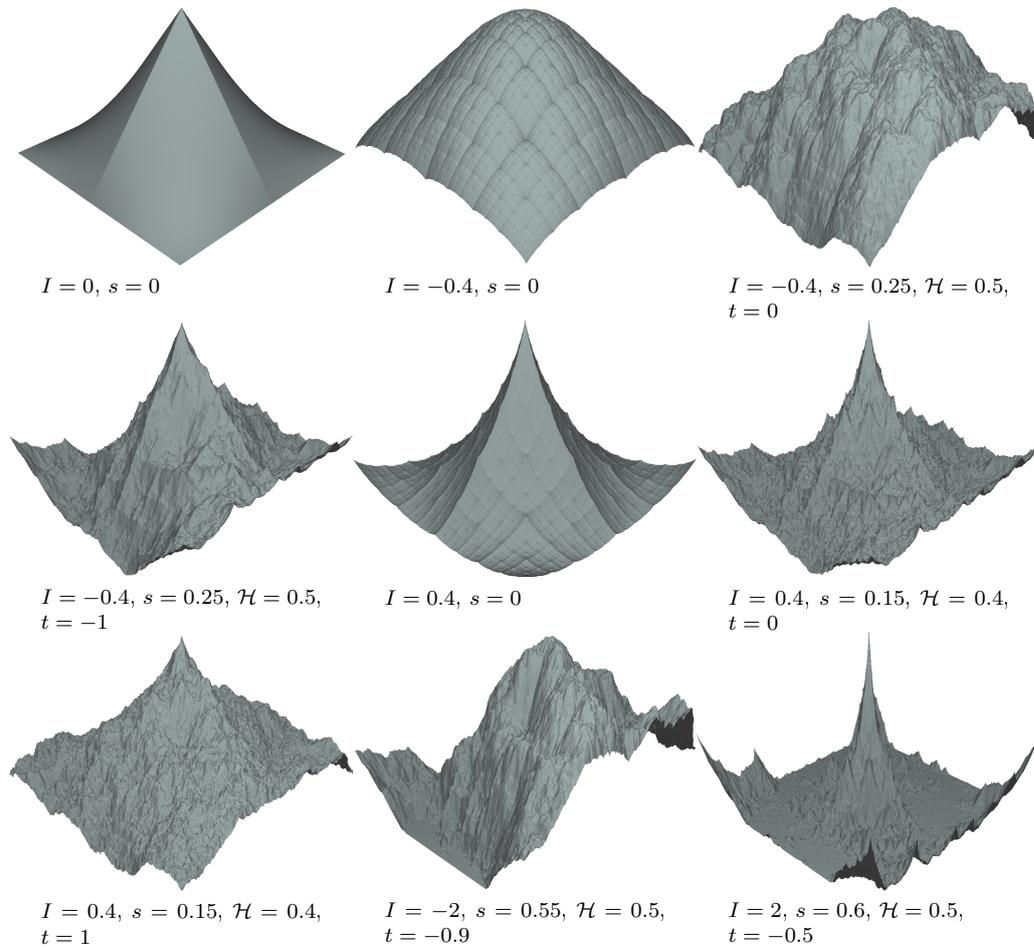


FIG. 3.8 – Variations de l'interpolation par Triangle-Edge; le calcul de moyennes est pondéré par la distance et la constante  $I$ .

et l'extrapolation est pondérée de manière à avoir une altitude décroissante par rapport à la distance euclidienne.

Ainsi, nous proposons de définir une fonction  $f_{-1}(a,d)$  modifiant l'apport de chaque nœud enfant à l'extrapolation de l'altitude du nœud parent. En effet, les nœuds enfants ne sont pas forcément à équidistance du nœud parent ; cette pondération est donc appliquée à chaque altitude de nœud enfant avant d'être utilisée pour l'extrapolation de l'altitude du nœud parent.

Comme pour  $f$  dans l'équation 3.4, nous donnons  $f_{-1}(a,d)$  telle que :

$$f_{-1}(a,d) = a \times (1 - \sigma(I_{-1}) \times (1 - (1 - \frac{d}{d_{max}})^{|I_{-1}|}))$$

$$\sigma(I_{-1}) = \begin{cases} 1 & I_{-1} \geq 0 \\ -1 & I_{-1} < 0 \end{cases} \quad (3.5)$$

où  $d_{max}$  est la distance euclidienne maximale ( $\sqrt{L^2 + H^2}$ ) pour un MNT  $\mathcal{T}[H][L]$ ,  $a$  l'altitude du nœud enfant,  $d$  la distance euclidienne entre le nœud enfant et le nœud parent et  $I_{-1}$  contrôle le comportement de la courbe d'extrapolation.

La figure 3.9 montre une variété de modèles générés par notre méthode DMMC ; l'**Algorithme 11** est donné ci-après. Nous utilisons l'ensemble des pondérations décrites dans cette section pour générer une variété de surfaces, toutes basées sur le modèle-squelette à crêtes en croix avec un segment décalé et les quatre bords fixés à zéro.

### 3.2.3 Notre algorithme généralisé et optimisé

Nous commençons par définir :

- $\mathcal{T}[H][L]$  le MNT-squelette donné ;
- $\text{dmm}(\mathcal{T}, I, s, t, \mathcal{H})$  la méthode de subdivision choisie (i.e. *Triangle-Edge* ou *Diamond-Square*) modifiée pour obtenir des interpolations non-linéaires ; cf. équation 1.1 pour  $s, t$  et  $\mathcal{H}$  et équation 3.4 pour  $I$  ainsi que les interpolations non-linéaires ;
- $f_{-1}$  et  $I_{-1}$  modifiant l'apport d'un nœud enfant dans l'extrapolation du nœud parent ; cf. équation 3.5 ;
- $\text{simu\_dm}(L, H, (e_x, e_y))$  recherche, en simulant la subdivision, le nœud enfant en position  $(e_x, e_y)$  et renvoie la liste des positions de ses deux, trois ou quatre parents ;
- $X$  un nombre pseudo-aléatoire compris dans l'intervalle  $[-1,1]$  ;
- Nous utilisons une file  $\mathcal{FPA}$  et au besoin des files  $\mathcal{Le}_{(x,y)}$  (i.e. la liste des nœuds enfants de  $(x,y)$ ) pour stocker des coordonnées de cellules. L'implémentation de la gestion de la file est standard [102] :
  - $\text{non\_vide}(\mathcal{F})$  retourne *vrai* si  $\mathcal{F}$  est non vide, sinon *faux* ;

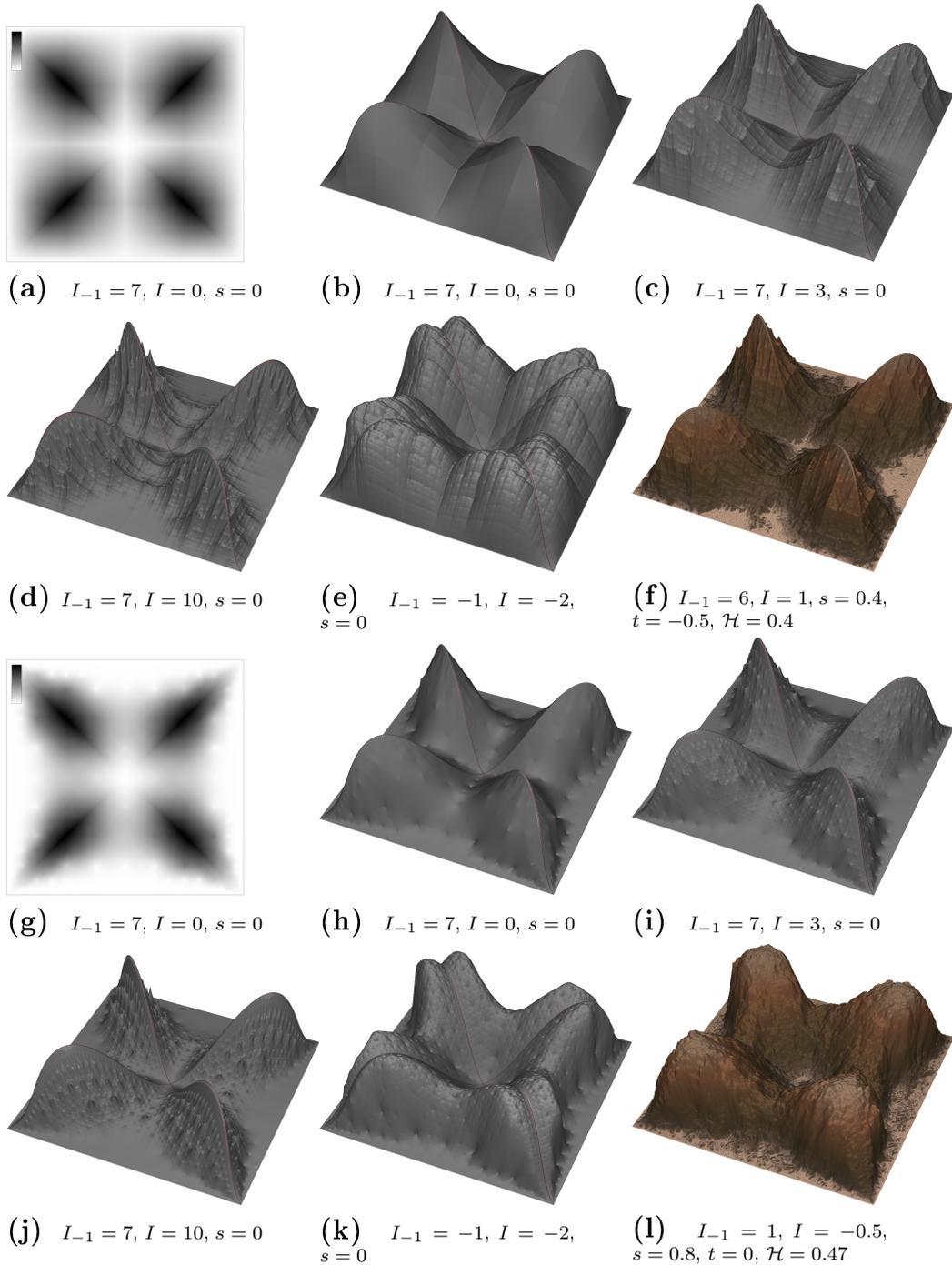


FIG. 3.9 – Modèles obtenus en modifiant les pondérations du processus d'extrapolation (le paramètre  $I_{-1}$ ) ainsi que les pondérations du processus d'interpolation (le paramètre  $I$ ). Ces modèles sont générés à partir du MNT-squelette en croix avec contre-diagonale décalée et bords à zéro. Les images (a) à (f) et (g) à (l) sont respectivement produites par un DMMC en Triangle-Edge et un DMMC en Diamond-Square.

- $\text{enfiler}(\mathcal{F}, (x, y))$  ajoute la coordonnée  $(x, y)$  en fin de file  $\mathcal{F}$ .
- $\text{defiler}(\mathcal{F})$  défile la coordonnée en tête de file  $\mathcal{F}$ .
- Nous utilisons un arbre binaire  $\mathcal{AB}$  permettant de stocker pour chaque coordonnée  $(x, y)$  un lien vers une liste de coordonnées  $\mathcal{L}e_{(x, y)}$ . Nous définissons deux fonctions de gestion de l'arbre :
  - $\text{position}(\mathcal{AB}, (x, y))$  cherche une entrée liée à la position  $(x, y)$  : retourne un lien vers la liste  $\mathcal{L}e_{(x, y)}$  si l'entrée est dans l'arbre  $\mathcal{AB}$ . Sinon ajoute l'entrée et crée une liste  $\mathcal{L}e_{(x, y)}$  vide et retourne le lien vers cette liste. L'insertion est faite de manière à trier les entrées selon leur abscisse puis leur ordonnée;
  - $\text{supprimer}(\mathcal{AB}, \mathcal{L}e_{(x, y)})$  supprime l'entrée liée à la position  $(x, y)$  de l'arbre  $\mathcal{AB}$ .

Le processus bas-haut de l'algorithme DMMC est décrit en détail dans l'**Algorithme 11**. Le processus haut-bas est réalisé par la procédure de déplacement des milieux  $\text{dmm}$  prenant en compte des interpolations non-linéaires telles que décrites dans la section 3.2.1.

---

**Algorithme 11** L'algorithme du Déplacement des Milieux Morphologiquement Contraint — DMMC.

---

**ENTRÉES:** Les éléments définis en début de section (i.e. section 3.2.3).

**SORTIES:** le MNT  $\mathcal{T}$  complété.

```

pour tout  $y \in \{0,1,2,\dots,H-1\}$  faire
  pour tout  $x \in \{0,1,2,\dots,L-1\}$  faire
    si  $T[y][x].e \neq \text{inconnu}$  alors  $\text{enfiler}(\mathcal{FPA}, (x,y))$ 
  fin pour
fin pour
tantque  $\text{non\_vide}(\mathcal{FPA})$  faire
  tantque  $\text{non\_vide}(\mathcal{FPA})$  faire
     $(e_x, e_y) \leftarrow \text{defiler}(\mathcal{FPA})$ 
    pour tout  $(p_x, p_y) \in \text{simu\_dm}(L, H, (e_x, e_y))$  faire
      si  $T[p_y][p_x].e \neq \text{inconnu}$  alors CONTINUER
       $\mathcal{L}e_{(p_x, p_y)} \leftarrow \text{position}(\mathcal{AB}, (p_x, p_y))$ 
       $\text{enfiler}(\mathcal{L}e_{(p_x, p_y)}, (e_x, e_y))$ 
    fin pour
  fin tantque
  pour tout nœud  $\mathcal{L}e_{(x,y)}$  dans  $\mathcal{AB}$  (*) faire
     $a \leftarrow D \leftarrow 0$ 
    tantque  $\text{non\_vide}(\mathcal{L}e_{(x,y)})$  faire
       $(e_x, e_y) \leftarrow \text{defiler}(\mathcal{L}e_{(x,y)})$ 
       $d \leftarrow \sqrt{(x - e_x)^2 + (y - e_y)^2}$ 
       $a \leftarrow a + d \times f_{-1}(T[e_y][e_x].a, d)$ 
       $D \leftarrow D + d$ 
    fin tantque
     $T[y][x].a \leftarrow \frac{T[y][x].a}{D}$ 
     $T[y][x].e \leftarrow \text{connu};$ 
     $\text{enfiler}(\mathcal{FPA}, (x,y))$ 
     $\text{supprimer}(\mathcal{AB}, \mathcal{L}e_{(x,y)})$ 
  fin pour
fin tantque
 $\text{dmm}(\mathcal{T}, I, s, t, \mathcal{H})$ 

```

(\*) effectuer un parcours préfixé

---

C'est le paradoxe de la modélisation de présenter sous la forme d'abstractions et d'algorithmes ce qu'il y a de plus concret dans la Nature, inhérent à toute organisation humaine, animale, ou végétale : le terrain.

En y regardant à deux fois, et sachant que, sur Terre en tout cas, un terrain doit se présenter comme un paysage vu par un oeil humain, ou par son représentant (tel que le Mars Rover), il est à son tour une abstraction. Le terrain, défini par ses contraintes, décide de la première ressource d'un peuplement d'êtres vivants : l'irrigation. Bien sûr, l'irrigation modifie également le terrain et en devient un tel déterminant que nous avons été tenté d'en faire un élément de contrainte : une *rivière abstraite* devient, convenablement utilisée, le descriptif formel permettant d'obtenir un générateur de terrain plausible tel que notre oeil s'y est habitué au cours des siècles.

A cette étape, il devenait inévitable de revenir à ce qu'une rivière peut faire de plus utile : c'est à dire relier les unités de peuplement et favoriser leur croissance, en bref, l'urbanisation du paysage.

## Chapitre 4

# Du terrain naturel au terrain urbanisé

Les chapitres précédents nous ont familiarisé avec une vision de reliefs contraints, en quelque sorte de haut en bas : des contraintes sont émises, ou reçues venant de l'utilisateur<sup>1</sup>, le relief s'en déduit, et tout un jeu mutuel entre reliefs et contraintes se déroule. C'est une évolution de type processus : des programmes qui ont un début mais pas une fin logique, car c'est en général à la suite d'un signal extérieur qu'un processus se termine. L'actualité scientifique, en 2007, évoque des hypothèses de pire cas, où le terrain n'évolue plus mais se retrouve aux prises de cataclysmes : astéroïdes, éruptions ou séismes de grande ampleur etc. Si à présent, nous voulons traiter l'hypothèse de terrains *habités*, le caractère incrémental (et de bas en haut) de leur constitution devient inévitable. Rares dans l'histoire de la démographie, sont les cas où une population de grande ou petite taille se constitue instantanément. Les variations se mesurent à l'échelle humaine : la famille, les amis et les groupes. Évidemment, avec une famille par ci, une famille par là, puis les naissances, nous aboutissons rapidement à un village, une bourgade, une puis des villes, des métropoles ; toutes sont généralement reliées par un réseau fluvial s'il existe déjà ou des canaux aménagés à posteriori. C'est dans ce contexte qu'il faut considérer le présent chapitre, et c'est ce que nous avons voulu dire en parlant de *terrain urbanisé*.

Nous allons donc traiter de la génération automatique de plan d'occupation des sols. Soit, à partir d'un MNT donné en entrée, nous réalisons, un par un, tous les éléments de la chaîne produisant en sortie les agglomérations et le réseau routier les reliant entre-elles.

---

1. Nous dirons *utilisateur* au sens large : il peut s'agir d'opérateurs humains, mais aussi de modules embarqués transmettant des contraintes issues de données de capteurs.

Nous utilisons, pour illustrer notre modèle, un MNT échantillonné avec une précision d'un point tout les 500 mètres et représentant la Corse. Nous appliquons des contraintes sur la distribution des tailles des agglomérations afin de les faire correspondre aux données du recensement de la population française, effectué en mars 1999 par l'INSEE [55]. Le réseau connectant les agglomérations est calculé et les routes sont définies en fonction du relief. Nous obtenons une carte routière de la Corse où les agglomérations sont représentées selon leur importance et où les liaisons sont coloriées selon leur type : autoroutes, nationales ou départementales.

## 4.1 Nos motivations : un problème d'intégration

Dans [86], le modèle proposé par Parish et Müller génère la configuration d'une agglomération correspondant à une base statistique donnée sous la forme d'une carte de densité. La méthode ne traite pas le problème à une plus grande échelle ; elle se limite au traitement d'une seule agglomération mais détaille le modèle jusqu'à donner une description des bâtiments. L'approche est procédurale et basée sur des modèles de L-Systèmes où les règles de production correspondent aux modèles d'urbanisations souhaités. Le premier résultat obtenu est un découpage en blocs de l'espace occupé décrit par la carte de densité de population. Cette dernière est fournie.

Ainsi, à son échelle<sup>2</sup>, cette méthode [86] propose des solutions satisfaisantes ; elle permet d'obtenir plusieurs configurations possibles (des pavages proches de ceux de Paris, San Francisco ou New York) ; mais demeure dépendante de questions importantes : où placer les agglomérations ? quelle est la répartition de la population ? où se situent les principaux axes ? Nous choisissons de donner des solutions aux interrogations posées. Soit des solutions aux différents problèmes de modélisation automatique se posant à l'échelle d'une région. Nous considérons qu'au delà de la taille d'une ville (i.e. à l'échelle où nous ne décrivons plus les rues) une zone est assez grande pour représenter une région<sup>3</sup>. A cette échelle, le problème consiste à calculer des sites pour les différentes agglomérations ou villes puis de relier ces sites de manière à obtenir une infrastructure routière optimale. Ainsi, les résultats obtenus en sortie d'un modèle opérant à cette échelle serviraient comme entrée à la méthode citée plus haut [86].

---

2. Par ailleurs, d'autres méthodes proposent des solutions aux problèmes de modélisation automatique d'éléments urbains se posant à des échelles plus petites [137, 16].

3. L'Union Européenne avec ses pays, vue du ciel aujourd'hui, présente du point de vue du réseau d'infrastructure routière la même topologie que la France et ses régions.

Nous décrivons le processus utilisé pour générer une carte de densité de population. Cette carte doit être automatiquement générée à partir du MNT donné en entrée. A partir du modèle de densité, nous calculons les emplacement des sites d'agglomération de manière à faire correspondre le rayon de ces sites avec une distribution approximant une loi de Zipf [44]. Nous calculons un premier réseau connectant les sites d'agglomération puis nous le simplifions en fonction de l'apport de chaque connexion. Enfin, nous proposons une méthode, basée sur des courbes de Bézier, calculant les trajectoires des routes reliant les sites d'agglomération.

## 4.2 Densité de population

La densité de population est une mesure définissant un nombre d'individus par unité de surface. Dans ce travail, nous nous intéressons exclusivement à la densité d'individus humains pour une unité de surface variable ; cette variation est fonction de la résolution du modèle numérique de terrain donné en entrée de notre système de calcul de densité de population. Ainsi, nous définissons un *Modèle Numérique de Densité de Population Humaine* (MNDPH) comme un tableau de dimensions égales à celles du MNT et contenant une valeur comprise dans un intervalle borné  $[0,1]$ . Ce MNDPH est utilisé, dans la section suivante, pour déterminer quelles sont les meilleures positions de sites d'agglomération pour le MNT donné. Soit le MNDPH doit correspondre à la réalité<sup>4</sup> traduite par le MNT ; par exemple : il n'existe pas d'agglomération subaquatique ou bien d'agglomération au-delà de 4000 mètres d'altitudes<sup>5</sup>.

### 4.2.1 Les quatre caractéristiques des unités de terrain urbanisé : les sommets du graphe

Nous proposons de générer, à partir du MNT donné en entrée du système, une partie des données servant à construire un MNDPH plausible. Il est possible de mettre en évidence plusieurs types d'informations liées à un MNT. Nous choisissons, pour chaque position dans le MNT, d'extraire quatre caractéristiques distinctes :

1. Savoir si une position appartient à un littoral : cela peut-être une position proche d'une zone considérée en dessous du niveau de la mer ;

---

4. L'utilisateur du système peut avoir sa propre idée de la réalité et la transcrire sous forme de contraintes correspondantes.

5. Nous donnons le cas extrême de Potosi, une ville bolivienne de 140000 habitants se trouvant à 3967 mètres d'altitude.

si une mer, un lac ou toute autre bassin sédimentaire immergé n'est pas clairement explicité dans la donnée décrite par le MNT. Aussi, par proche, nous entendons toutes les positions se trouvant dans une fourchette de distances par rapport à une étendue d'eau. Cette fourchette est comprise entre le rayon minimal (i.e. le hameau) et le rayon maximal (i.e. la mégapole) correspondant à un site d'agglomération (i.e. ces valeurs sont fixées par rapport à l'échelle utilisée). Plus une position est entourée, à moyenne distance, d'étendues d'eau, plus grande sera sa valeur de position côtière. Ces valeurs sont ré-échelonnées dans l'intervalle  $[0,1]$  une fois le calcul effectué pour toutes les positions ;

2. L'hydrométrie calculée en chaque position : nous déterminons, dans la section 4.2.2, une carte d'hydrométrie pour un MNT donné. Un tableau de dimensions égales à celles du MNT est obtenu. Nous avons, en chaque position, une valeur comprise dans l'intervalle  $[0,1]$  informant de la quantité relative de flux passant en cette position.
3. Savoir si une position est un site appartenant à un plateau possible. Pour simplifier, l'utilisateur peut fixer une altitude à partir de laquelle un site est considéré comme appartenant à un plateau ; il fixera un seuil représentant un maxima à ne pas dépasser. Dans les exemples présentés dans cette section, nous considérons un minima à 400 mètres et un maxima à 4000 mètres pour une zone de type plateau. Pour améliorer la détection, nous opposons la définition des plaines (cf. le point suivant) à celles des plateaux. Puis, nous appliquons le même système utilisé lors du calcul des pondérations des positions appartenant au littoral ;
4. Savoir détecter les plaines : soit informer si une position appartient à une zone où le relief est relativement plat. Pour une zone centrée en cette position et délimitée par un rayon égal au rayon du plus petit site possible (valeur fixée par l'utilisateur et liée à l'échelle), étudier la variation des altitudes de cette zone. Plus la variation moyenne est proche de zéro, plus la position est considérée comme appartenant à une plaine. L'inverse est obtenu pour des valeurs proches à supérieurs au maxima fixé par l'utilisateur ; nous fixons ce maxima à 50 mètres pour les exemples présentés dans cette section. Enfin, les pondérations obtenues sont ré-échelonnées dans l'intervalle  $[0,1]$  ;

Ensuite, nous pondérons chacune des quatre informations extraites par une valeur négative, nulle ou positive. Cette valeur informe sur la nature respectivement *répulsive*, *neutre* ou *attractive* de la caractéristique en question. Soit, pour les  $P_i$  avec  $i \in \{1,2,3,4\}$ , les pondérations respectives des caractéristiques 1 à 4, nous calculons le MNDPH  $\mathcal{D}$  correspondant au MNT  $\mathcal{T}$  donné en entrée :

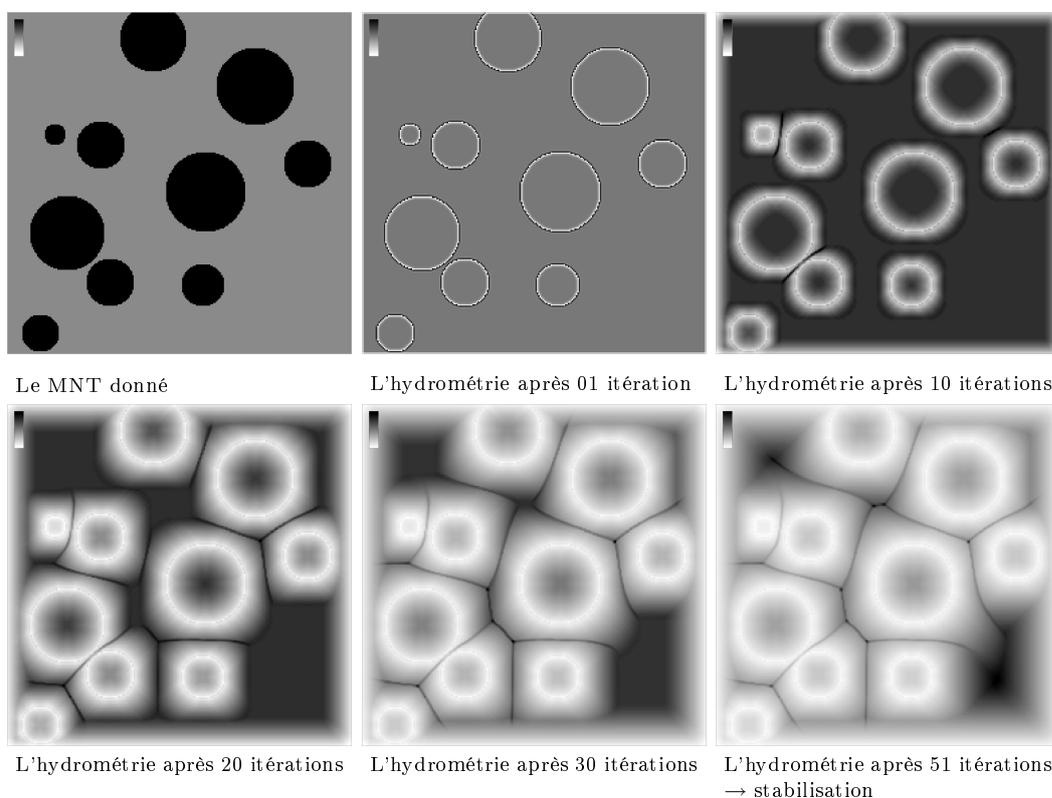


FIG. 4.1 – Une hydrométrie calculée pas à pas.

$$\mathcal{D}[y][x] = \frac{\sum_{i=1}^4 P_i \times \mathcal{D}_i[y][x]}{\sum_{i=1}^4 P_i}$$

où les  $\mathcal{D}_i[y][x]$  représentent le calcul de la caractéristique  $i$  pour la position  $(x,y)$ .

Ainsi, à partir d'un minimum de données (c.-à-d. le MNT) nous pouvons modifier le paramétrage du système afin d'obtenir différentes configurations possibles de cartes de densité de population. Nous pouvons, de la même manière, introduire des données extérieures (par exemple : cartes des forêts, cartes de zones industrielles ou cartes de zones protégées) et les recouper avec les informations générées afin de produire un nouveau modèle MNDPH.

### 4.2.2 Les arcs d'un graphe d'urbanisation : liaisons fluviales et hydrométrie

Dans [122, 121], l'auteur propose une méthode itérative de calcul de l'hydrologie d'un modèle numérique de terrain. Le but du travail était d'extraire le réseau fluvial sous-jacent à un MNT. Nous nous inspirons de ce travail pour résoudre un autre problème. Nous souhaitons simuler une averse et mesurer la quantité d'eau passant, sans exceptions, en chacune des positions d'un MNT.

Ainsi, nous proposons un modèle de calcul de carte d'hydrométrie pour un MNT donné. Une même quantité initiale d'eau est placée à l'initialisation *au dessus* de chacune des cellules du MNT. Puis, à chaque instant, nous mesurons et accumulons la quantité d'eau passant en chaque cellule. Ensuite, le flux se déplace d'une cellule donnée vers ses cellules voisines (dans [121] seul le meilleur voisin est choisi, i.e. le voisin ayant la plus forte pente descendante). Le voisinage est 8-connexe. Le fluide se déplace vers une cellule voisine si et seulement si le vecteur représentant la direction du fluide et le vecteur représentant la direction de la cellule voisine vont dans une même direction ; soit pour un produit scalaire positif. La quantité de fluide déplacé vers une cellule voisine est relative au scalaire obtenu. Le processus est répété itérativement. La quantité d'eau passant en chaque cellule est accumulée de proche en proche et pour un nombre d'itérations donné. Pour toute cellule, la valeur accumulée en fin de processus est ré-échelonnée dans l'intervalle  $[0, 1]$  et par rapport à l'ensemble des cellules ; cette valeur représente la mesure finale d'hydrométrie de la cellule.

#### 4.2.2.1 Notre implémentation du modèle hydrométrique

Pour réaliser ce calcul, nous utilisons un tableau, noté  $\mathcal{H}m$ , de mêmes dimensions que le MNT  $\mathcal{T}[H][L]$  donné (cf. section 1.1.1). Pour chaque position  $(x, y)$ , la structure sous-jacente à  $\mathcal{H}m[y][x]$  est composée de :

1. Huit vecteurs, notés  $\vec{\mathcal{V}}_i(\mathcal{H}m[y][x])$  avec  $0 \leq i < 8$ . Les  $\vec{\mathcal{V}}_i(\mathcal{H}m[y][x])$  sont les vecteurs obtenus en calculant la différence de coordonnées entre la position  $(x, y, \frac{\mathcal{T}[y][x].a}{2^{16}-1})$  et  $(x_i, y_i, \frac{\mathcal{T}[y_i][x_i].a}{2^{16}-1})$  où  $(x_i, y_i)$  est la  $i^{\text{ème}}$  cellule voisine de  $(x, y)$  pour un voisinage 8-connexe. Si la cellule voisine ne représente pas une coordonnée valide du MNT (i.e. la coordonnée est au delà des bords du MNT, soit  $\mathcal{T}[y_i][x_i].a$  n'existe pas) alors nous utilisons la coordonnée  $(x_i, y_i, -1)$  à la place ;
2. Un vecteur  $\vec{\mathcal{F}}(\mathcal{H}m[y][x])$ , sa norme représente la quantité et son orientation la direction du fluide présent au niveau de la cellule  $\mathcal{T}[y][x]$  à

l'instant donné ; par analogie avec une goutte de pluie en chute libre, nous posons, à l'initialisation, pour toute position  $(x,y)$  :  $\vec{\mathcal{F}}(\mathcal{H}m[y][x]) = (0, -1, 0)$  ;

3. Une valeur  $\mathcal{Q}(\mathcal{H}m[y][x])$  représentant la quantité de fluide passant à un instant donné par la cellule  $\mathcal{T}[y][x]$  ; cette valeur est initialisée à zéro.
4. Une valeur  $\mathcal{A}cc(\mathcal{H}m[y][x])$  représentant le total de quantité de fluide passée par la cellule  $\mathcal{T}[y][x]$  ; cette valeur est initialisée à zéro.

Un deuxième tableau, temporaire, noté  $\mathcal{H}'m[H][L]$ , est utilisé pour simuler un calcul effectué au même instant et pour toutes les positions. La structure nécessaire pour ce tableau comprend seulement les points 2, 3 et 4 cités plus haut. Ainsi, pour chaque position  $(x,y)$  dans  $\mathcal{H}'m[H][L]$ , nous stockons un vecteur  $\vec{\mathcal{F}}(\mathcal{H}'m[y][x])$  et deux quantités  $\mathcal{A}cc(\mathcal{H}'m[y][x])$  et  $\mathcal{Q}(\mathcal{H}'m[y][x])$ .

Soit, à chaque itération, et simultanément pour toutes les positions  $(x,y)$  dans  $\mathcal{H}m$ , nous calculons pour chaque vecteur entrant  $\vec{\mathcal{F}}(\mathcal{H}m[y][x])$  huit vecteurs  $\vec{R}_i$  sortants représentant la diffusion du fluide vers les cellules voisines. Ce calcul est détaillé dans l'**Algorithme 12** où nous donnons le développement de l'ensemble d'une itération de diffusion. La totalité du processus de diffusion converge vers un état stable ; nous le détectons en observant la moyenne des quantités de déplacement pour une itération donnée. Cette moyenne débute à 1 et converge vers la valeur presque nulle  $2^{-126}$  (la plus petite valeur flottante positive) ; ici, nous considérons la convergence du système à partir d'une valeur moyenne inférieure à  $2^{-20}$ . Ainsi, après le nombre d'itérations souhaité, ou la convergence du calcul, nous ré-échelonnons les quantités accumulées en chaque  $\mathcal{A}cc(\mathcal{H}m[y][x])$ . Ces valeurs représentent la carte d'hydrométrie du MNT donné.

Nous montrons, figure 4.1, différents stades du calcul de l'hydrométrie pour un MNT représentant une surface binaire : soit une surface composée de seulement deux niveaux distincts ; un plan à altitude médiane sur lequel nous dessinons des cercles à altitudes maximales ; soit un modèle de terrain représentant une plaine parsemée de cylindres à hauteur pseudo-infinie. Les images données figure 4.1 représentent, de gauche à droite et du haut vers le bas, le MNT utilisé puis son hydrométrie après respectivement une, dix, vingt, trente et cinquante et une itérations ; cette dernière est considérée comme l'état stable du système de calcul de l'hydrométrie.

#### 4.2.2.2 Application de l'hydrométrie dans le calcul du MNDPH

Nous utilisons les données créées via le modèle d'hydrométrie pour produire des cartes de densité de population. Nous montrons, pour quatre différents MNTs, quelques résultats obtenus. Pour tous ces exemples, nous uti-

---

**Algorithme 12** Une itération du calcul de l'hydrométrie d'un MNT.

---

**ENTRÉES:** Un tableau  $\mathcal{H}m[H][L]$  initialisé selon le MNT  $\mathcal{T}[H][L]$  donné et comme énuméré plus haut (cf. section 4.2.2.1). Un tableau  $\mathcal{H}'m[H][L]$  dont tous les champs sont initialisés à zéro.

**SORTIES:** Un pas du processus de calcul de l'hydrométrie d'un MNT.

```

pour tout  $0 \leq y < H$  faire
  pour tout  $0 \leq x < L$  faire
     $\Sigma n \leftarrow 0$ 
    pour tout  $0 \leq i < 8$  faire
       $\vec{R}_i \leftarrow \vec{\mathcal{F}}(\mathcal{H}m[y][x]) + \vec{\mathcal{V}}_i(\mathcal{H}m[y][x])$ 
       $S_i \leftarrow \vec{R}_i \odot \vec{\mathcal{V}}_i(\mathcal{H}m[y][x])$ 
       $N_i \leftarrow \|\vec{R}_i\|$ 
       $\Sigma n \leftarrow \Sigma n + N_i$ 
    fin pour
    pour tout ( $0 \leq i < 8$  avec  $(x_i, y_i)$  coordonnées du voisin  $i$ ) faire
      si ( $S_i > 0$  et  $(x_i, y_i) \in \{0, L-1\} \times \{0, H-1\}$ ) alors
         $\vec{\mathcal{F}}(\mathcal{H}'m[y_i][x_i]) \leftarrow \vec{\mathcal{F}}(\mathcal{H}'m[y_i][x_i]) + \vec{R}_i$ 
         $\mathcal{Q}(\mathcal{H}'m[y_i][x_i]) \leftarrow \mathcal{Q}(\mathcal{H}'m[y_i][x_i]) + \frac{N_i \times \mathcal{Q}(\mathcal{H}m[y][x])}{\Sigma n}$ 
      finsi
    fin pour
  fin pour
   $\Sigma \mathcal{Q} \leftarrow 0$ 
  pour tout  $0 \leq y < H$  faire
    pour tout  $0 \leq x < L$  faire
       $\text{Acc}(\mathcal{H}'m[y][x]) \leftarrow \text{Acc}(\mathcal{H}m[y][x]) + \mathcal{Q}(\mathcal{H}m[y][x])$ 
       $\Sigma \mathcal{Q} \leftarrow \Sigma \mathcal{Q} + \mathcal{Q}(\mathcal{H}m[y][x])$ 
    fin pour
  fin pour
   $\Sigma \mathcal{Q} \leftarrow \frac{\Sigma \mathcal{Q}}{L \times H}$ 
  si ( $\Sigma \mathcal{Q} \approx 0$ ) alors
    STABILISATION DU SYSTÈME : FIN DU CALCUL.
  finsi

```

---

lisons les quatre caractéristiques décrites en début de section. Chacune est utilisée en fonction d'une pondération. Soit pour le quadruplé de caractéristiques (littoral, hydrométrie, plateau, plaine) nous noterons les pondérations respectivement affectées sous la forme  $(P_1, P_2, P_3, P_4)$ .

La figure 4.2, illustre les modèles obtenus pour trois MNTs faiblement différents. Les zones complètement blanches dénotent l'absence totale de population. Il s'agit des zones au dessus (ici 4000 mètres) et en dessous (ici le niveau de la mer, soit 0 mètres) des seuils limites. Nous fixons un troisième seuil au niveau de la quantité d'hydrométrie enregistrée en une position. Ce seuil est atteint seulement pour les valeurs maximales. Les figures 4.2(a) et 4.2(b) montrent des MNDPH calculés pour le MNT et son hydrométrie illustrés en figure 4.1. Les pondérations (littoral, hydrométrie, plateau, plaine) utilisées sont respectivement  $(0, 1, -1.5, 0.3)$  et  $(0, 1, 1.5, -0.3)$ . Soit le littoral est neutre (il n'y en a pas dans cet exemple) et le taux d'hydrométrie est attractif dans les deux cas. Dans cet exemple, nous avons mis en opposition les plateaux et les plaines. Sur l'image figure 4.2(a), la proximité des plateaux (des falaises) est répulsive et les plaines sont légèrement attractives : une large bande rouge (dense) se forme au niveau des plaines et spécialement sur les positions à forte hydrométrie. Sur l'image figure 4.2(b), la proximité des plateaux est attractive et opposée aux plaines légèrement répulsives : un halo rouge se forme autour des falaises et, au niveau des plaines, seules les positions à forte hydrométrie ont une forte densité.

L'image figure 4.2(c) est obtenue avec les mêmes pondérations utilisées en (b) et pour une variante du modèle de terrain présenté en figure 4.1, déjà utilisé pour calculer (a) ou (b). Nous remplaçons les cylindres, soit les cercles noirs et uniformes en vue  $2D$ , par des cônes : soit des cercles dessinés en dégradé en vue  $2D$ , du noir au gris, de l'intérieur vers l'extérieur. Les zones dépassant les seuils sont réduites ; la coupe au niveau des cônes donne des cercles, à densités nulles, plus petits. Maintenant il existe une zone intermédiaire représentant les plateaux et le halo rouge produit autour des montagnes (i.e. des cônes) est plus large et dénote une plus grande densité.

Une troisième variante du MNT montré sur la figure 4.1 est obtenue en remplaçant les cylindres par des cônes traversant le plan médian (i.e. la plaine) jusqu'au point le plus bas ; soit des cercles dessinés en dégradé en vue  $2D$ , du noir au blanc, de l'intérieur vers l'extérieur. Il en résulte des bassins sédimentaires immergés autour des parties émergées des cônes. L'image figure 4.2(d) montre le MNT obtenu ; ici le code couleur est modifié, il représente : du bleu foncé au bleu clair les positions immergées de la plus profonde à la moins profonde ; du vert clair au marron en passant par un vert foncé les positions émergées de la plus basse à la plus haute. L'image figure 4.2(e) montre en superposition de (d) : la carte hydrométrique obtenue est superposée par

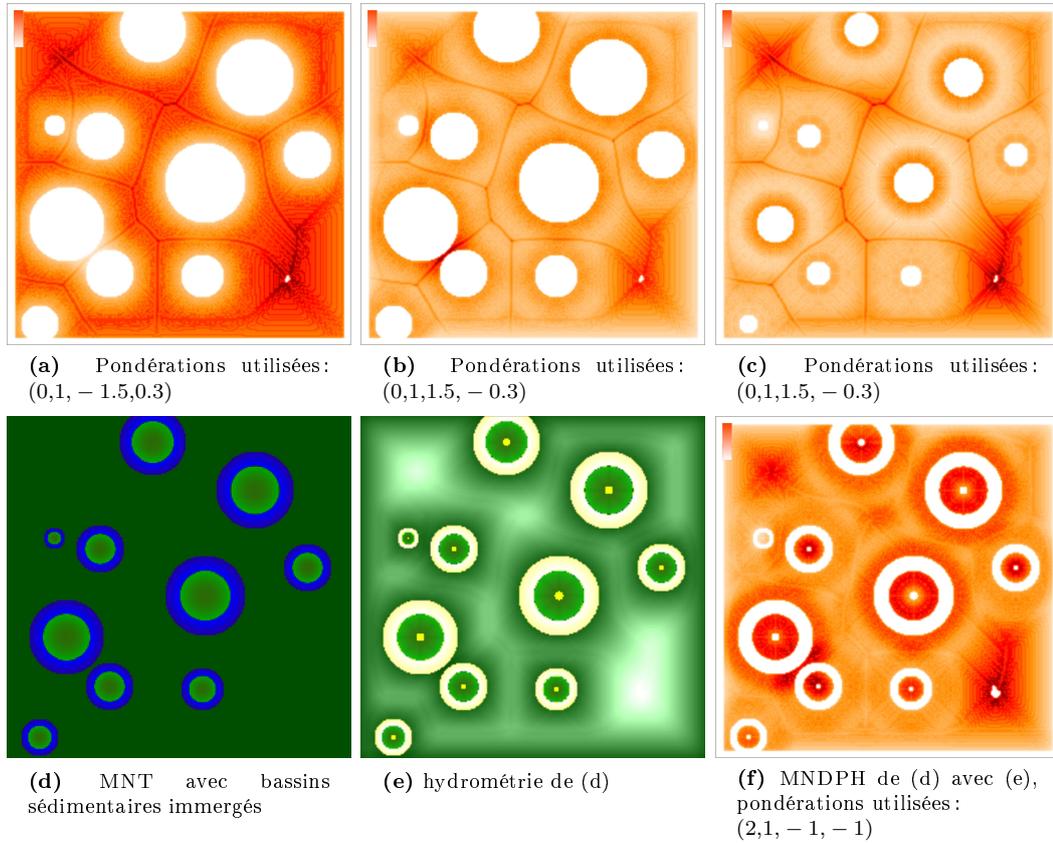
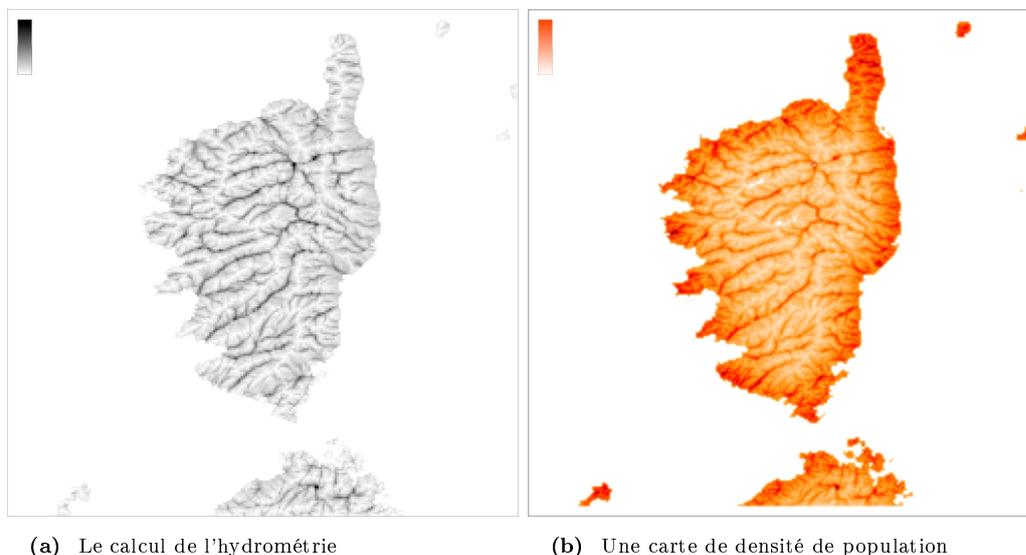


FIG. 4.2 – *Modèles numériques de densité de population humaine obtenus pour trois MNTs faiblement différents.*

éclaircissement ; les zones atteignant les seuils sont superposées en jaune. Finalement, la densité de population est calculée pour des pondérations (littoral, hydrométrie, plateau, plaine) respectivement égales à  $(2,1, -1, -1)$ , l'image figure 4.2(f) en montre un rendu : le littoral est ici plus attractif, l'hydrométrie est deux fois moins attractive et les plateaux et les plaines sont répulsifs ; ici l'effet répulsif des plateaux n'est pas assez important pour annuler l'effet attractif du littoral se trouvant à proximité mais l'effet répulsif des plaines réduit l'effet faiblement attractif de certaines régions à moyenne hydrométrie.



MNT de la Corse : latitude nord 40.983333 à 43.116667 avec un pas de 0.008333 degrés — longitude ouest 8.000000 à 10.133333 avec un pas de 0.008333 degrés.

FIG. 4.3 – *Un calcul d'hydrométrie puis d'une densité de population que nous avons réalisé pour un modèle numérique de la Corse.*

Enfin, un exemple de calcul de modèle de densité de population est réalisé pour un MNT de dimensions  $256 \times 256$  représentant une partie du globe terrestre : l'île de la Corse et le nord de l'île de Sardaigne. Ce type de modèles est disponible au *National Geophysical Data Center* [81] dans un format brut 16 ou 32 bits auquel est associé à un fichier d'en-tête décrivant les latitudes, les longitudes, la fréquence d'échantillonnage et des données statistiques. La figure 4.3(a) montre l'hydrométrie obtenue après une vingtaine d'itérations et 227 millisecondes de temps de calcul sur notre architecture de référence spécifiée en section 1.1.1.2. La figure 4.3(b) montre le MNDPH calculé pour cet MNT et pour des pondérations (littoral, hydrométrie, plateau, plaine) égales à  $(1,1,1.5,0)$  ; ce modèle est obtenu en 923 millisecondes. Plus tard, nous utiliserons cet MNDPH pour les calculs des positions des sites d'agglomération liés à ce modèle numérique de terrain.

## 4.3 Le réseau d'agglomérations

### 4.3.1 Position et configuration des sites d'agglomérations

Nous définissons un site d'agglomération comme la composante unique ou plurielle, principale ou secondaire, d'une agglomération ou unité urbaine ; soit l'unité urbaine est le résultat d'un assemblage urbain composé d'un ou de plusieurs sites d'agglomération. A notre échelle, et pour un MNT donné, nous considérons un site d'agglomération comme un couple de coordonnées associé à un rayon. Un site est classé dans une catégorie, ou rang, uniquement selon son rayon. Cette catégorisation ne prend pas en compte le type<sup>6</sup> du site. Cette information apparaîtra plus tard, elle sera décrite par le réseau obtenue. Ainsi, en fonction d'une fourchette de dimensions (i.e. le rayon minimal et le rayon maximal) un quota est fixé pour chaque catégorie et selon une distribution prédéfinie. Cette distribution peut être donnée par l'utilisateur et pour des distances discrètes liées à l'échelle du MNT.

#### 4.3.1.1 Distribution utilisée : une approximation de Zipf

Pour les exemples présentés ici, nous utilisons une fonction approximant la distribution des unités urbaines par nombre d'individus en France métropolitaine. Cette distribution a été calculée à l'aide des données du recensement de la population française de mars 1999 [55]. La figure 4.4 donne le nombre d'unités urbaines de taille associée à une fourchette de nombre d'individus. Nous pourrions approximer cette distribution, notée  $\mathcal{U}$ , par une loi de Zipf [44]. Dans une loi de Zipf la population est inversement proportionnelle au rang de l'unité urbaine ; ainsi la plus importante unité urbaine comprend le plus grand nombre d'individus et appartient au rang 1, la deuxième comprend deux fois moins d'individus et appartient au rang 2, et ainsi de suite jusqu'à un rang  $n$  donné ; ici nous souhaitons confondre la notion de rang avec le nombre d'unités appartenant à ce même rang, ainsi au rang  $n = 1072$  nous aurions 1072 unités urbaines de ce même rang et cela correspondra aux données de la distribution  $\mathcal{U}$ . Comme nous pouvons le constater sur la figure 4.4, la loi originelle de Zipf (ici nous utilisons  $y = rang = \frac{x_{max}}{x}$  où  $x_{max} = 10^6$ ) approxime mal la distribution  $\mathcal{U}$ . La loi modifiée améliore le résultat mais reste difficilement paramétrable ; ici  $y = rang = \frac{x_{max}}{x^{0.8}}$ . Nous proposons une meilleure approximation en utilisant une fonction exponentielle inverse  $f$

---

6. Par exemple, selon des données de l'INSEE, nous pouvons distinguer plusieurs catégories dans le cas des communes françaises : les lieux dits, les villages ou les villes isolées, les banlieues et leur centre ville.

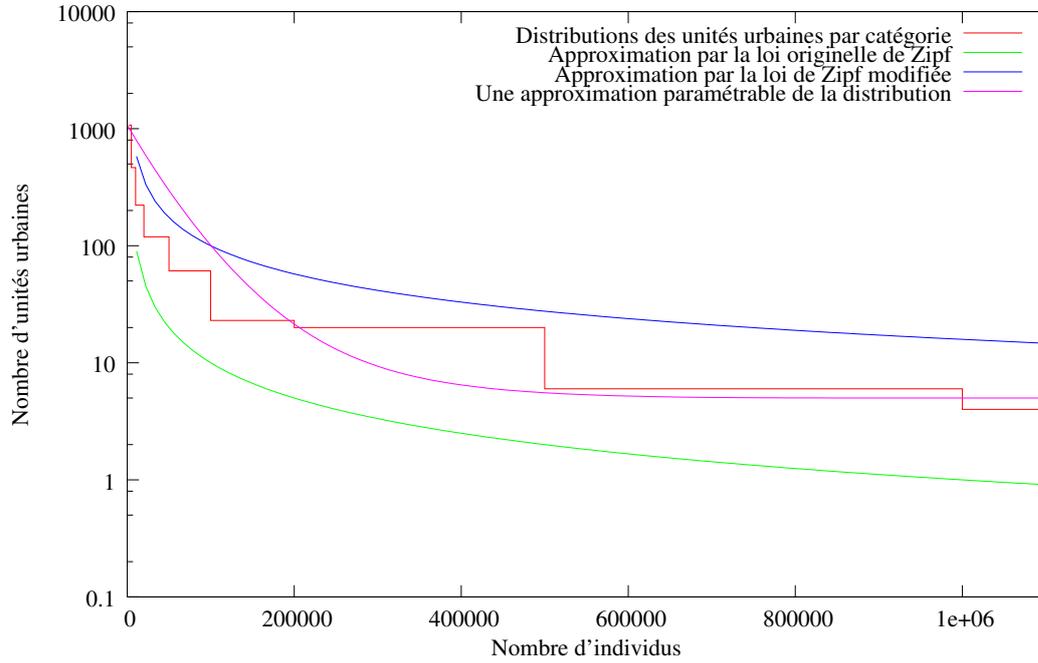


FIG. 4.4 – Approximation de la distribution des unités urbaines en France métropolitaine. Cette distribution associe un nombre d'unités urbaines pour le nombre d'individus la composant.

continue et dérivable dans  $[0, x_{max}]$  où pour chaque  $x \in [0, x_{max}]$  individus,  $y = rang = f(x)$  donne le nombre d'unités urbaines du rang (ou catégorie) correspondant. Soit, pour un nombre d'unités urbaines décroissant en fonction du nombre d'individus les composant, nous posons :

$$f(x) = \mathcal{U}_{min} + (\mathcal{U}_{max} - \mathcal{U}_{min}) \left( \frac{x_{max} - x}{x_{max}} \right)^p \quad (4.1)$$

où  $\mathcal{U}_{min}$  et  $\mathcal{U}_{max}$  sont respectivement la valeur minimale et la valeur maximale données par la distribution approximée  $\mathcal{U}$  et  $p > 1$  contrôle la vitesse de décroissance de la courbe. Dans cet exemple, nous avons :

- $\mathcal{U}_{min} = 4$  pour 4 unités urbaines de plus d'un million d'habitants ;
- $\mathcal{U}_{max} = 1072$  pour 1072 unités urbaines de moins de cinq mille habitants ;
- $x_{max} = 10^6$ , est la borne inférieure de la fourchette supérieure ;
- $p = 4$ , choisi de manière empirique, donne une approximation de la distribution  $\mathcal{U}$ .

Par analogie, nous transposons cette distribution liée au nombre d'individus [55] pour l'utiliser dans le calcul des rayons  $r$  des sites d'agglomération ;

---

**Algorithme 13** Calcul d'un quota de catégories d'agglomération.

---

**ENTRÉES:**  $\mathcal{U}_{min}$ ,  $\mathcal{U}_{max}$ ,  $r_{min}$ ,  $r_{max}$ ,  $dr_{min}$  entiers strictement positifs et  $e$  l'échelle du MNT utilisé. Les fonctions  $f$  et  $g$  sont respectivement définies dans (4.2) et (4.2).

**SORTIES:**  $\mathcal{Q}$  le tableau de quotas.

$N_{sites} \leftarrow 0$

$dr \leftarrow \max(dr_{min}, \lfloor \frac{1}{e} \rfloor)$

$n \leftarrow \lfloor \frac{r_{max} - r_{min}}{dr} \rfloor$

**pour tout** ( $i \in \{0, 1, \dots, n-1\}$ ) **et** ( $r \in \{r_{min}, r_{min} + dr, \dots, r_{max} - dr\}$ ) **faire**

$\mathcal{Q}[i] \leftarrow \lfloor f(g(r)) \rfloor$

$N_{sites} \leftarrow N_{sites} + \mathcal{Q}[i]$

**fin pour**

---

nous posons le changement de variable transformant le nombre d'individus en rayon de site d'agglomération :

$$x = g(r) = r \times 4 \times 10^2 \quad (4.2)$$

où  $x$  est le nombre d'individus utilisé dans (4.1) pour le calcul d'une approximation de  $\mathcal{U}$  et  $r$  est le rayon, en mètres, des sites d'agglomération ; ce changement de variable fait correspondre une unité urbaine d'au minimum un million d'habitants à un site d'agglomération d'un rayon minimum de 2500 mètres.

Ainsi nous pouvons calculer une distribution prenant la forme d'un tableau de quotas, noté  $\mathcal{Q}[n]$ , pour  $\mathcal{U}_{min}$ ,  $\mathcal{U}_{max}$  fixés et  $r_{min} \geq dr$ ,  $r_{max} \geq r_{min}$  donnés tels que  $r \in \{r_{min}, \dots, r_{max} - dr\}$  ; avec  $r_{min}$  et  $r_{max}$  des multiples de  $dr$ , l'incrément de  $r$  ; tous sont donnés en mètres et sont liés à l'échelle  $e$  du MNT par  $dr = \max(dr_{min}, \lfloor \frac{1}{e} \rfloor)$  avec  $dr_{min} \gg 1$  ; nous posons ce minima, d'une part, afin d'éviter un pas inférieur à 1 unité de mesure, soit le mètre, d'autre part, pour éviter une catégorisation trop détaillée. En pratique, nous posons  $dr_{min} = 50$  ; il n'est pas utile dans l'exemple réalisé sur l'île de la Corse, ici, l'échelle est égale à 0.002 soit nous avons une mesure tout les 500 mètres (i.e.  $dr = 500$ ). **Algorithme 13** développe le calcul des quotas stockés dans  $\mathcal{Q}$ . Nous en profitons pour calculer un total de nombre de sites noté  $N_{sites}$  ; il sera utilisé lors du calcul de positionnement des sites.

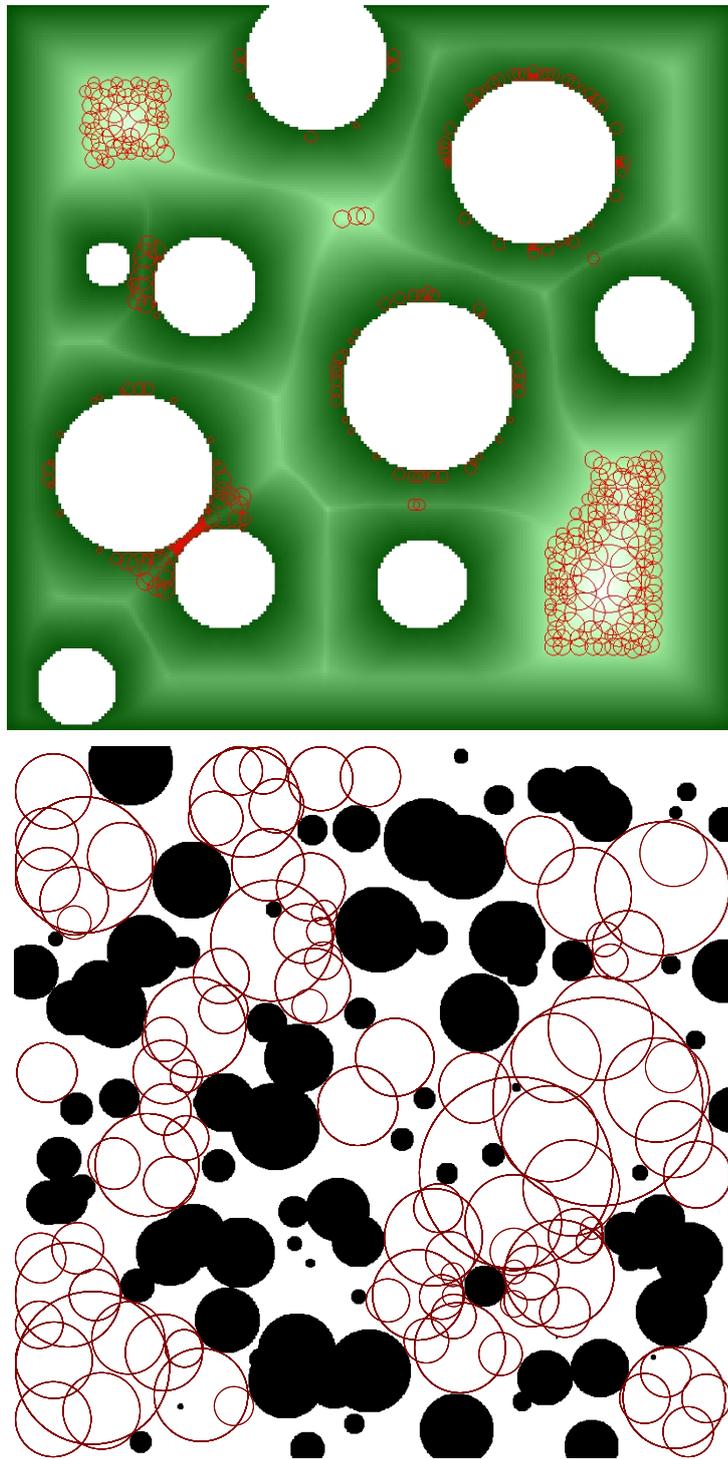


FIG. 4.5 – Configuration et positionnement automatique de sites d'agglomération pour un MNT donné.

### 4.3.1.2 Positionnement des sites

Notre approche consiste à placer les sites d'agglomération de manière à correspondre aux données décrites dans un modèle de densité de population.

D'une part, il faudrait pouvoir limiter le nombre et la catégorie des sites créés. Nous proposons une méthode stochastique basée sur les *Random Neighborhood Graphs* [139,138]. La méthode est modifiée afin de garder seulement un nombre limité de sites obtenus. Cette restriction prend la forme d'un tableau de quotas  $\mathcal{Q}$  (donné ou calculé comme détaillé ci-dessus) fixant un nombre limite de sites par catégorie (i.e. rayon du site).

D'autre part, la dimension des sites obtenus doit correspondre à une densité exprimée au niveau de chacune des positions sur le terrain. Nous utilisons les résultats de la méthode présentée dans la précédente section (cf. section 4.2) afin de générer un MNDPH pour un MNT donné. La valeur présente en chaque position du MNDPH est assimilée à un potentiel de densité relative ; chacune est définie dans l'intervalle  $[0,1]$ .

Soit un MNDPH  $\mathcal{D}$  où  $\mathcal{D}[y][x]$  donne le potentiel de densité relative d'une position  $(x,y)$ . Nous plaçons  $M$  positions dans une liste  $\mathcal{L}_{sites}$  de sites probables et nous posons :  $M \gg N_{site}$  ;  $N_{site}$  est un majorant du nombre de sites à placer, cette valeur est calculée dans **Algorithme 13**. Ces  $M$  positions dans  $\mathcal{L}_{sites}$  peuvent représenter toutes les positions constructibles  $N_{cons}$  ; soient les positions n'ayant pas atteint les seuils limites, par exemple : au dessus ou en dessous d'un certain niveau d'altitude, au dessus de la limite hydrométrique ou appartenant à un relief trop accidenté ; cette dernière caractéristique dépend fortement de l'échelle de mesure, elle devient accessoire à partir de résolutions supérieures à la dizaine de mètres par point. Nous proposons une heuristique pour éviter de traiter une liste trop importante de positions probables. Nous ajoutons, dans  $\mathcal{L}_{sites}$  et au hasard, les  $M$  positions parmi toutes les positions constructibles ; ces dernières sont équiprobables et  $M \ll N_{cons}$ . Nous trions, de manière décroissante, la liste  $\mathcal{L}_{sites}$  des sites  $S$ . Le tri est effectué en fonction du potentiel des positions correspondantes aux sites ; soit, si  $S \in \mathcal{L}_{sites}$  est un site centré en  $S.c$  et de rayon  $S.r$  à définir, son potentiel est donné par  $\mathcal{D}[S.c.y][S.c.x]$ . Nous parcourons la liste des sites ainsi triée. Pour chaque élément  $S$  dans  $\mathcal{L}_{sites}$ , un processus de croissance du site correspondant est réalisé. Nous essayons de faire correspondre ce site à une catégorie de plus en plus grande ; une catégorie est donnée par un rayon  $r$  et un quota correspondant à ce rayon est stocké dans  $\mathcal{Q}[i]$  avec  $r = r_{min} + i \times dr$  (nous rappelons :  $r_{min}$  est le plus petit rayon de site et  $dr$  est l'incrément constant du rayon). Soit, pour un rayon  $r$ , nous simulons, à l'aide d'une fonction `intersecte_seuil`, le dessin d'un cercle centré en  $S.c$  et de rayon  $r$ . Tous les points  $(x,y)$  du cercle discret résultant sont testés afin de détec-

ter une intersection avec une zone non constructible du MNT ; soit une zone dont les positions ont atteint des seuils limites. Dans ce cas, deux situations se présentent :

- Si une intersection est détectée, le processus de croissance est arrêté pour ce site et nous utilisons, s'il existe, le dernier (soit le plus grand) rayon valide obtenu. Un rayon est valide si aucune intersection n'est détectée et pour une catégorie sous-jacente disponible selon les quotas donnés dans  $\mathcal{Q}$ .
- Si aucune intersection n'est détectée pour  $S.c$  et  $r$ , le rayon  $r$  est considéré valide si les quotas le permettent ; dans tous les cas nous continuons le processus de croissance jusqu'à atteindre la plus grande catégorie.

Ainsi, en fin de croissance d'un site, nous regardons s'il peut correspondre à une catégorie donnée, notée  $j$ , et disponible selon  $\mathcal{Q}[j]$ . Si la condition est satisfaite, alors le rayon du site est fixé selon la catégorie et le quota dans cette catégorie décroît  $\mathcal{Q}[j] = \mathcal{Q}[j] - 1$ . Dans le cas contraire, le site est supprimé, via la procédure `supprimer`, de la liste  $\mathcal{L}_{sites}$ .

Ce processus est réalisé pour tous les éléments de  $\mathcal{L}_{sites}$  mais arrêté une fois la quantité  $N_{sites}$  atteinte ; c.-à-d. au moment où tous les quotas sont épuisés. Soit à cet instant, nous avons au maximum nos  $N_{sites}$  premiers éléments de  $\mathcal{L}_{sites}$  respectant les quotas décrits à l'initialisation dans  $\mathcal{Q}$ . Nous choisissons de supprimer les sites non traités par le processus de croissance.

La liste  $\mathcal{L}_{sites}$  est maintenant triée par rayon de site décroissant. Il est probable d'obtenir des sites d'agglomération inclus dans d'autres. Dans certains cas, nous supprimons les sites considérés inclus. Pour un couple de sites  $S$  et  $T$  avec  $S.r \geq T.r$ , nous distinguons deux cas possibles d'inclusion :

- $T$  est complètement inclus dans  $S$  ; le cercle  $((T.c.x, T.c.y), T.r)$  est inclus dans le cercle  $((S.c.x, S.c.y), S.r)$  ; soit le centre  $T.c$  est dans le cercle  $((S.c.x, S.c.y), S.r)$  et les deux cercles ne s'intersectent pas : la distance entre les deux centres plus le rayon  $T.r$  est inférieur ou égale au rayon  $S.r$  ;
- $T$  est partiellement inclus dans  $S$  et de rang équivalent ; soit le centre  $T.c$  est dans le cercle  $((S.c.x, S.c.y), S.r)$  et les deux cercles s'intersectent : la distance entre les deux centres plus le rayon  $T.r$  est supérieure au rayon  $S.r$ . Sachant  $T.r$  est inférieur ou égal à  $S.r$  nous considérons  $T$  et  $S$  de rang équivalent si  $2 \times T.r > S.r$ .

Dans le premier cas, nous supprimons  $T$  de la liste en considérant sa contribution nulle à l'augmentation de la superficie de l'agglomération. Dans le deuxième cas, nous supprimons seulement les sites  $T$  partiellement inclus dans  $S$  si le rayon  $T.r$  est supérieur à la moitié du rayon du  $S.r$  ; l'idée sous-jacente est de considérer le site secondaire  $T$  (la banlieue) comme une

continuité au minimum deux fois plus petite par rapport au site principal  $S$  (le centre). Ce modèle donne des résultats proches des configurations de grandes villes à plan radio-concentrique comme Paris. Pour obtenir des configurations proches d'un modèle à pavage régulier, comme la partie centrale de l'île de Manhattan, il suffit d'omettre cette condition supplémentaire et supprimer tous les cas d'inclusion partielle.

Ces suppressions de sites peuvent produire un déséquilibre de proportion par rapport aux quotas imposés dans  $\mathcal{Q}$ . Cette proportion peut être rétablie par ajout ou suppression de sites dans  $\mathcal{L}_{sites}$ ; une suppression est réalisée en un parcours sur une liste préalablement triée par densités croissantes et rétablie rapidement une proportion équivalente à celle donnée par les quotas  $\mathcal{Q}$ . Par ailleurs, plusieurs solutions peuvent être proposées pour rétablir les quotas par ajout de sites. Dans ce cas, la résolution du problème de convergence vers une configuration respectant les quotas imposés est équivalente à la résolution du problème du sac à dos. En pratique, ce déséquilibre étant minoritaire et statistiquement probable, nous considérons la configuration obtenue satisfaisante et nous la laissons inchangée.

Le détail de la méthode utilisée est donné par l'**Algorithme 14**. L'image en haut de la figure 4.5 montre un résultat obtenu pour le modèle de densité de population affiché en figure 4.2(b) et utilisant la distribution donnée en section 4.3.1.1. Les sites d'agglomération apparaissent en rouge. L'hydrométrie apparaît par mélange de rendu; la quantité ré-échelonnée d'hydrométrie augmente l'intensité de couleur de la position correspondante. Le MNT est représenté en deux couleurs, les plaines sont en vert et les montagnes en blanc; ces cylindres à hauteur pseudo-infinie représentent des zones non constructibles. Nous utilisons un deuxième modèle dont l'échelle est deux fois plus petite afin d'obtenir l'image en bas de la figure 4.5. Les plaines sont en blanc et les cylindres à hauteur pseudo-infinie sont en noir et en plus grand nombre; ils représenteront les obstacles à contourner au moment de la création des routes reliant les sites d'agglomération. Ces derniers, occupent l'espace de manière homogène, le MNDPH utilisé pour cet exemple contient, pour toute position constructible, la même valeur de densité 1. Enfin, nous remarquons la présence de zones non constructibles au sein de certains sites (les petits disques noirs dans les cercles rouges); cela est dû à l'utilisation d'un grand incrément de rayon  $dr$ , la zone se trouve exactement entre deux états consécutifs du processus de croissance du site. Ainsi, il est probable de ne pas détecter des zones non constructibles de diamètre, au maximum, égal à l'incrément de rayon  $dr$ . Cette configuration peut correspondre à une réalité, nous la laissons donc inchangée.

---

**Algorithme 14** Création et positionnement automatique des sites d'agglomération.

---

**ENTRÉES:** Un tableau de quotas  $Q[n]$  pour les  $N_{sites}$  sites cibles de rayons allant de  $r_{min}$  à  $(r_{max} - dr)$  avec un pas  $dr$ .  $\mathcal{L}_{sites}$  est une liste de  $M$  sites  $S$  probables ;  $M \gg N_{site}$ .

**SORTIES:** Une liste de sites d'agglomération correspondant approximativement aux quotas donnés dans  $Q[n]$ .

Trier  $\mathcal{L}_{sites}$  par densité relative décroissante (\*)

**pour tout** ( $(S \in \mathcal{L}_{sites})$  **et tant que**  $N_{sites} > 0$ ) **faire**

$i \leftarrow 0$  **et**  $j \leftarrow -1$

**pour tout** ( $S.r \in \{r_{min}, r_{min} + dr, \dots, r_{max} - dr\}$ ) **faire**

**si** ( $\text{intersecte\_seuil}(S.c.x, S.c.y), S.r) \leftarrow \text{vrai}$ ) **alors**

ARRÊTER L'ITÉRATION

**sinon**

**si** ( $Q[i] > 0$ ) **alors**  $j \leftarrow i$

**finsi**

**fin pour**

**si** ( $j \neq -1$ ) **alors**

$N_{sites} \leftarrow N_{sites} - 1$  **et**  $Q[j] \leftarrow Q[j] - 1$  **et**  $S.r \leftarrow r_{min} + j \times dr$

**sinon**

supprimer( $\mathcal{L}_{sites}, S$ )

**finsi**

**fin pour**

supprimer les sites jamais atteints

Trier  $\mathcal{L}_{sites}$  par rayon de site décroissant (\*)

**pour tout** ( $S \in \mathcal{L}_{sites}$ ) **et pour tout** ( $T \in \text{suivant}(S)$ ) **faire**

**si** ( $\|\overrightarrow{T.c \ S.c}\| < S.r$ ) **alors**

**si** ( $(\|\overrightarrow{T.c \ S.c}\| + t.r \leq S.r)$  **ou** ( $2 \times T.r > S.r$ )) **alors**

supprimer( $\mathcal{L}_{sites}, T$ ) (\*\*)

**finsi**

**finsi**

$i \leftarrow 0$  **et**  $j \leftarrow -1$

**fin pour**

(\*) Aucun ordre n'est ajouté aux valeurs égales contiguës.

(\*\*) Dans ce cas les quotas données par  $Q[n]$  ne sont plus respectés ; l'équilibre (la proportionnelle) peut être rétabli par ajout ou suppression de sites dans  $\mathcal{L}_{sites}$ . Nous choisissons de laisser une part de déséquilibre et laissons la configuration telle quelle.

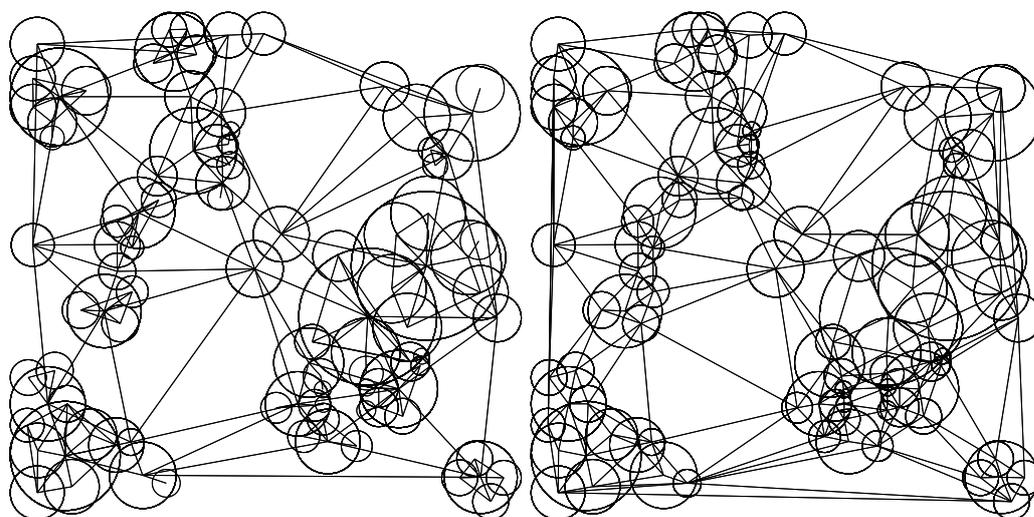
---

### 4.3.2 Connecter les sites d'urbanisation

Une description de l'espace sous-jacent à un MNT est obtenue en positionnant des sites d'agglomération puis en les reliant entre-eux. Ici, nous connaissons les positions et rayons des sites d'agglomération. Nous utilisons plusieurs triangulations de Delaunay [59] et un algorithme de parcours de graphes afin de produire une première définition du réseau routier reliant tous les sites. Le graphe souhaité est non dirigé et possède une seule composante connexe. Chaque site, considéré comme un nœud du graphe, est atteignable à partir de tout autre site via une combinaison de routes, considérées comme les arêtes du graphe. Ce graphe peut être cyclique.

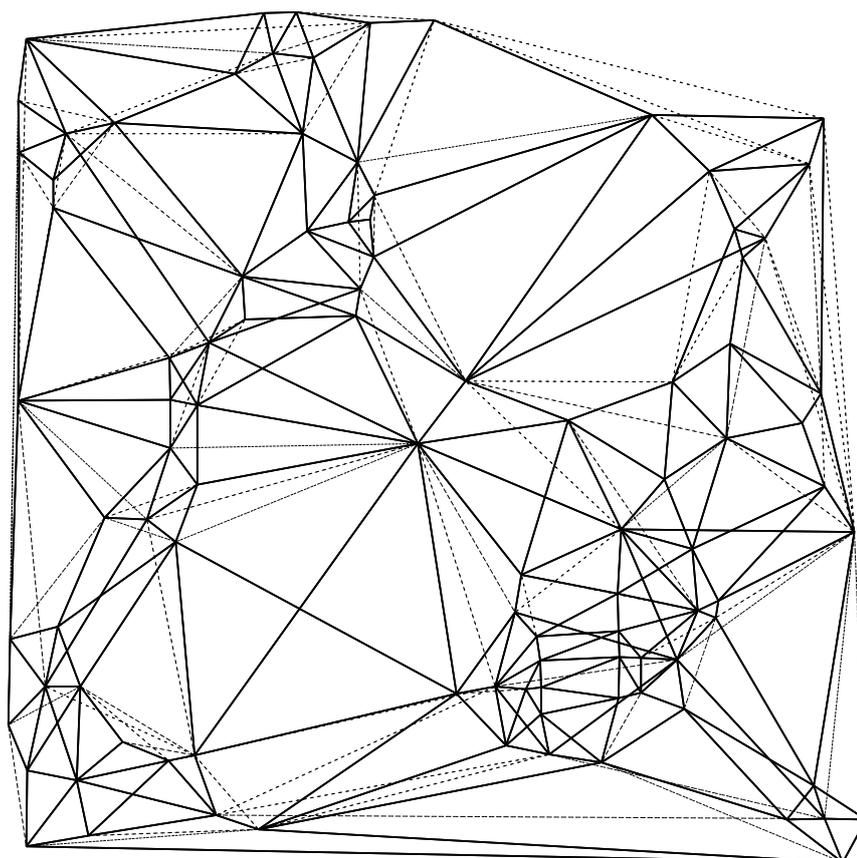
Nous utilisons une configuration initiale composée d'une superposition de plusieurs sous graphes planaires. Ces derniers sont obtenus par des triangulations de Delaunay réalisées séparément sur plusieurs sous ensembles de sites d'agglomération. Cette configuration initiale est simplifiée ; nous supprimons les redondances d'arêtes et la question de garder certaines liaisons directes entre deux sites donnés est posée. Ainsi, pour un ensemble de sites d'agglomération où chacun est défini par une position dans le plan et un rayon, nous donnons la configuration du graphe initial comme la superposition de :

- plusieurs triangulations de Delaunay, chacune est réalisée sur un sous ensemble de sites. Les sites d'agglomération sont réorganisés en arbre  $n$ -aire décrivant la hiérarchisation entre sites ; un critère d'inclusion sert à insérer un site à la position adéquat dans l'arbre  $n$ -aire. Nous reprenons les mêmes conditions utilisées dans la section 4.3.1.2 précédente et nous considérons les sites partiellement inclus et de rang inférieur comme nœuds enfants ; la deuxième partie de l'**Algorithme 14** est modifiée afin de réaliser cette hiérarchisation dans le même processus supprimant les sites complètement inclus ou les sites partiellement inclus et de même rang. L'**Algorithme 15** donne le détail de la modification apportée. Soit pour l'arbre  $n$ -aire ainsi obtenu, nous calculons, d'une part, une triangulation sur l'ensemble des sites présents à la racine de l'arbre  $n$ -aire ; cette triangulation est assimilée au réseau décrivant exclusivement les liaisons inter unités urbaines. Nous calculons, d'autre part, une triangulation pour chaque sous arbre  $n$ -aire de profondeur 1 appartenant l'arbre principal. Soit, une triangulation de Delaunay est réalisée pour tout ensemble de sites composé d'un nœud parent et de l'ensemble de ses nœuds enfants obtenus par descendance direct. La figure 4.6(a) montre les triangulations obtenues (i.e. les sous graphes planaires) pour la configuration de sites illustrée sur l'image du bas de la figure 4.5.
- une triangulation de Delaunay calculée, sans discrimination, sur l'en-



(a) Graphe composé d'une superposition de plusieurs sous graphes planaires créés par triangulation de Delaunay sur des sous ensembles de sites obtenus par hiérarchisation.

(b) Graphe obtenu par triangulation de Delaunay sur l'ensemble des sites pris sans discrimination.



(c) Réseau final obtenu à partir de (a) et (b) : le graphe est simplifié ; des liaisons directes sont supprimées (les arêtes hachurées) pour un faible apport comparé à la meilleure liaison indirecte correspondante.

FIG. 4.6 – Calcul du réseau connectant l'ensemble des sites d'agglomération.

semble des sites donnés. La figure 4.6(b) montre le graphe obtenu pour la configuration de sites illustrée sur l'image du bas de la figure 4.5.

Notre configuration initiale étant une superposition de plusieurs graphes planaires nous unifions l'ensemble en un unique graphe non-planaire et supprimons les doublons d'arêtes. Ce graphe est par la suite simplifié et les arêtes correspondant aux critères suivants sont supprimées :

- Une arête peut être supprimée seulement si elle ne correspond pas à une liaison parent-enfant. Soient les routes reliant les centres d'agglomérations aux banlieues ne peuvent être supprimées ; ces routes représentent les grands axes des grandes agglomérations ;
- Une arête est supprimée s'il existe un deuxième<sup>7</sup> plus court chemin reliant les deux nœuds extrémités de l'arête, ce plus court chemin doit représenter une alternative satisfaisante par rapport à la liaison directe offerte par cette arête. Un chemin indirect est dit satisfaisant si la distance, notée  $D_i$ , parcourue en empruntant ce chemin est inférieure à  $k \times D_d$  où  $k$  est une constante strictement supérieure à 1 et  $D_d$  est la distance euclidienne entre les deux nœuds extrémités de l'arête.

Les arêtes sont prises par ordre décroissant de dimension de sites d'agglomération ; le traitement d'une arête est effectué une seule fois. Ainsi, pour toute arête ne représentant pas de liaison enfant-parent, nous utilisons l'algorithme de Dijkstra [37] déterminant si un chemin indirect satisfait la condition  $D_i < k \times D_d$  posée. Le processus de recherche de plus court chemin étant effectué en profondeur d'abord, des coupes sont réalisées au moment où la valeur  $k \times D_d$  est atteinte ou dépassée. L'ensemble du processus de recherche est arrêté et l'arête est supprimée pour un plus court chemin indirect satisfaisant  $D_i < k \times D_d$ .

La figure 4.6(c) montre le réseau simplifié obtenu à partir des graphes illustrés en figure 4.6(a) et (b). La valeur utilisée pour la constante  $k$  est 1.8 ; i.e. un chemin indirect avec un surcoût inférieur à 80% du chemin direct implique la suppression de ce dernier. Les liaisons directes supprimées apparaissent en lignes hachurées.

## 4.4 Finalisation du réseau : le calcul des routes

Nous présentons une approche permettant de calculer les routes de sinuosité arbitraire reliant chaque paire de sites d'agglomération liés par une arête.

---

<sup>7</sup> Le graphe étant décrit sur un plan, le réseau sous-jacent est dit euclidien [103]. Soit, pour un couple de nœuds  $(A,B)$ , s'il existe une arête  $AB$ , le plus court chemin entre  $A$  et  $B$  est cette arête  $AB$ .

---

**Algorithme 15** Hiérarchisation des sites d'agglomération.

---

**ENTRÉES:** Une liste  $\mathcal{L}_{sites}$  de sites d'agglomération.

**SORTIES:** Supprime les sites complètement inclus ou les sites partiellement inclus et de même rang. Créé une arbre  $n$ -aire  $\mathcal{A}$  décrivant la hiérarchisation des sites.

Trier  $\mathcal{L}_{sites}$  par rayon de site décroissant (\*)

insérer( $\text{racine}(\mathcal{A})$ ,  $\text{tête\_de}(\mathcal{L}_{sites})$ )

**pour tout** ( $S \in \mathcal{L}_{sites}$ ) **et pour tout** ( $T \in \text{suivant}(S)$ ) **faire**

**si** ( $\|\overrightarrow{T.c \ S.c}\| < S.r$ ) **alors**

**si** ( $(\|\overrightarrow{T.c \ S.c}\| + t.r \leq S.r)$  **ou** ( $2 \times T.r > S.r$ )) **alors**

      supprimer( $\mathcal{L}_{sites}$ ,  $T$ ) (\*\*)

**sinon**

      insérer( $\text{fils\_de}(\mathcal{A}, S)$ ,  $T$ )

**finsi**

**sinon**

    insérer( $\text{racine}(\mathcal{A})$ ,  $T$ )

**finsi**

$i \leftarrow 0$  **et**  $j \leftarrow -1$

**fin pour**

(\*) Aucun ordre n'est ajouté aux valeurs égales contiguës.

(\*\*) Dans ce cas les quotas données par  $\mathcal{Q}[n]$  (cf. **Algorithme 14**) ne sont plus respectés ; l'équilibre (la proportionnelle) peut être rétabli par ajout ou suppression de sites dans  $\mathcal{L}_{sites}$ . Nous choisissons de laisser une part de déséquilibre et laissons la configuration telle quelle.

---

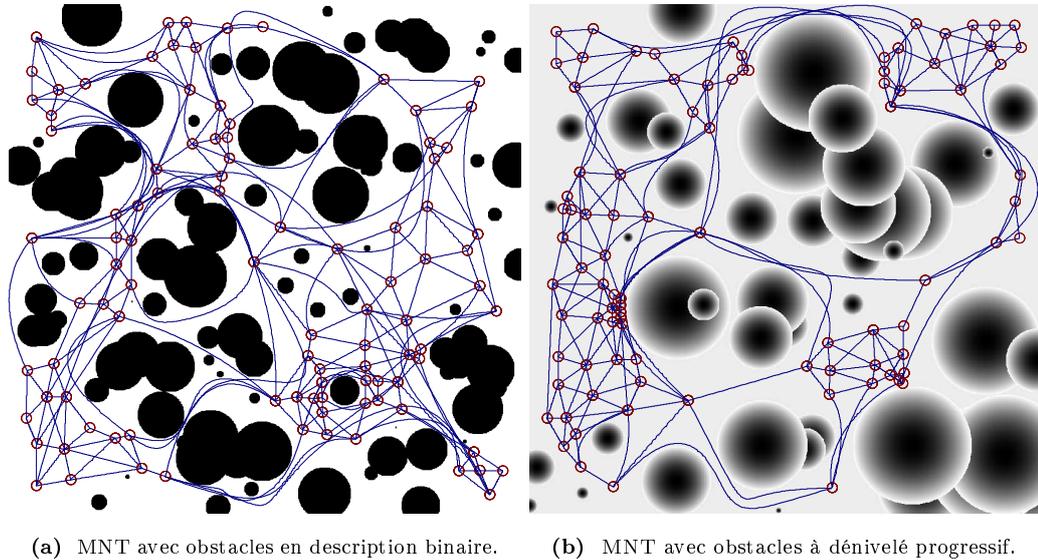


FIG. 4.7 – Le calcul des routes à partir du réseau d'agglomérations.

Nous nous imposons une description des routes indépendante de l'échelle du modèle numérique de terrain utilisé. Ainsi, pour un exemple de simulation automobile, avant ou pendant la phase de rendu, nous pourrions affiner, à l'aide de notre algorithme DMMC, le MNT au niveau de la jonction entre le relief et la route.

L'échelle à laquelle est réalisé l'ensemble du processus d'urbanisation décrit dans ce chapitre commence au niveau d'une région composée de plusieurs unités urbaines. Les routes décrites ici représentent au minimum les grands axes des sites d'agglomération et s'étendent en voiries communales, routes départementales, nationales ou autoroutes. Il est important, pour ces catégories de voiries, d'avoir une représentation décrivant une courbe  $C^1$  continue.

#### 4.4.1 Quelles solutions disponibles ?

Nous nous intéressons aux travaux utilisant des courbes de Bézier ( $C^\infty$ ) ou des  $B - Splines$  ( $C^1$ ) afin de planifier le mouvement de mobiles non holonomes (i.e. un mobile soumis à des contraintes cinématiques). La plupart de ces travaux traitent de problèmes de planification de mouvement dans des espaces de configuration où les obstacles sont identifiés par une représentation binaire [79, 38, 111, 93] ou par couches [62].

Zvi Shiller propose, dans plusieurs de ses travaux, l'utilisation de  $B - Splines$  pour la planification de trajectoires sur des surfaces avec dénivelés. L'approche consiste à minimiser le temps de parcours d'un mobile, avec

comme contraintes supplémentaires, des dénivelés de terrain pénalisant les trajectoires calculées. Une recherche globale et hiérarchique est effectuée sur un ensemble initial de chemin possibles [107] ; elle est raffinée par modification des points de contrôle décrivant la *B – Spline*. Ainsi des planifications de trajectoires sont réalisées dans [106, 108] pour des exemples de surfaces décrites par des patches de Bézier, et dans [104, 105], où la planification de mouvement du *Mars Rover* est réalisée et testée sur la *JPL MarsYard* [70] une simulation de la surface de Mars.

Par ailleurs, il existe d’autres approches traitant de la planification de trajectoire de mobiles évoluant sur divers terrains accidentés. Ainsi, une planification de mouvement pour un mobile de type chenille à  $n$  roues est proposée dans [110]. Les auteurs utilisent un espace de configuration tridimensionnel où le terrain est décrit à plusieurs niveaux de détails par des *quadtrees* correspondant à des cuboïdes dans cet espace de configuration. La trajectoire est calculée par un algorithme de type  $A^*$  ; les liaisons entre les configurations discrètes sont réalisées par arcs de cercle où le coût est donné selon les contraintes du mobile. Une autre approche, proposée dans [84], raffine la trajectoire en fonction d’une représentation multirésolution du terrain obtenue par ondelettes.

#### 4.4.2 Notre approche de la problématique des routes

Les méthodes présentées ont une complexité trop importante ou ne correspondent pas exactement à notre problématique. Ces méthodes sont souvent appliquées aux modèles de terrain basse résolution ou basse définition (i.e. peu de valeurs possibles en altitude de point) et pour le calcul d’une trajectoire entre seulement deux emplacements. D’une part, nous construisons plusieurs routes entre plusieurs couples de sites d’agglomération. D’autre part, le coût calculé ici n’est pas lié à la sûreté du mobile empruntant une trajectoire mais au coût lié à la sécurisation de la route empruntée par ce même mobile. Ainsi, pour un parcours ne respectant pas les contraintes de sûreté imposées, la route correspondante est aménagée de manière à garantir la contrainte de sûreté imposée. Soit l’approche requise ici tend à minimiser le nombre et le coût des aménagements réalisés ; les aménagements ayant un coût proportionnel à la non satisfaction de la contrainte liée. Comme exemple d’aménagements coûteux, nous pouvons citer la construction de ponts, tunnels, viaducs ou liaisons maritimes. Cette dernière, à la différence des autres, a un coût seulement lié au flux de véhicules empruntant la liaison. Pour les autres aménagements, le coût est en plus calculé en fonction de la distance parcourue.

Néanmoins, l’approche proposée par Zvi Shiller nous semble la plus adap-

table à notre problématique. Les contraintes de notre système sont plus lâches comparées aux contraintes d'un robot mobile explorant son espace. Par exemple, il est possible de limiter le temps de calcul d'une trajectoire donnée et de se contenter de la meilleure solution trouvée parmi les configurations explorées. Les portions infranchissables des trajectoires trouvées seront aménagées.

### 4.4.3 Une première solution au problème des routes

Nous proposons une nouvelle méthode de recherche et de raffinement des trajectoires. Notre algorithme réalise, pour une durée de temps imparti et de manière stochastique, un raffinement incrémental d'une courbe de Bézier<sup>8</sup> décrivant la route reliant deux sites d'agglomération. Le processus est répété jusqu'à obtenir : soit une trajectoire satisfaisante où la pire position discrète possède un coût inférieur à un seuil donné ; soit aucune trajectoire satisfaisant les contraintes imposées n'est trouvée au-delà de la durée de temps imparti.

Ce processus décrit ici est indépendant pour chaque arête reliant des sites d'agglomération. Il est initialisé par une courbe de Bézier d'ordre 2 comprenant les deux extrémités de l'arête. Un algorithme de De Casteljaou [42] est utilisé pour la représentation de la courbe sur le plan discret décrit par le MNT. Nous parcourons la courbe discrète et calculons un coût de passage entre la position précédente et la position en cours (les deux positions extrêmes ont par défaut un coût nul). Les coûts sont calculés en fonction de la pente entre les deux positions et en fonction de l'altitude de la position ; les deux sont ré-échelonnées dans l'intervalle  $[-1,1]$  puis la valeur absolue est utilisée comme coût en cette position. Nous utilisons, pour les exemples obtenus ici, un filtre  $f(x) = \sqrt{x}$  modifiant les valeurs initiales de coûts et accentuant la différence (i.e. une augmentation de contraste) entre les situations favorables les situations défavorables. Soit pour une suite de points discrets, où chaque position possède un coût dans l'intervalle  $[0,1]$ , nous trions ces positions par coût décroissant. La méthode tente de trouver une solution au pire cas en améliorant le coût global de la courbe. Pour un pire cas, nous calculons la normale à la courbe en position correspondante. Nous tirons au hasard un point le long de l'axe décrit par cette normale. Le tirage suit une loi normale centrée au pire cas et d'écart-type proportionnel à la longueur de la courbe discrète. La position obtenue est testée comme nouveau point de contrôle de la courbe de Bézier. Son coût total est calculé et comparé au coût de la courbe initiale. Nous posons un nombre de tentatives limité par pire cas

---

8. La fonction paramétrique d'une courbe de Bézier permet de déduire, en tout point de la courbe, une vitesse. Cette vitesse est continue et dérivable et pourrait servir à poser directement des limitations de vitesse sur les routes créées.

traité : si une solution est trouvée avant cette limite, i.e. la nouvelle courbe possède un coût global inférieur, la nouvelle courbe remplace l'ancienne et l'ensemble du processus est répété ; sinon, i.e. aucune meilleure solution n'est trouvée pour ce pire cas, le pire cas suivant est traité à son tour. Enfin, nous nous fixons une limite de temps pour la recherche d'une seule trajectoire entre deux extrémités d'arêtes.

Les définitions suivantes sont utilisées dans le développement de la méthode donnée en **Algorithme 16**. Ainsi, nous posons :

- `init_Bézier( $P, Q$ )` initialise une structure pour une courbe de Bézier dont les extrémités sont  $P$  et  $Q$  ;
- `discrétiser( $\mathcal{T}[H][L], B_{PQ}$ )` discrétise la courbe de Bézier  $B_{PQ}$  par rapport au plan discret décrit par le MNT  $\mathcal{T}[H][L]$ . Cette procédure renvoie un tableau  $t$  à  $n$  éléments où  $t[i].p$  est la  $i^{\text{ème}}$  position discrète à partir de  $P$ ,  $t[i].i$  son indice et  $t[i].c$  son coût ;
- `normal( $B_{PQ}, DB_{PQ}[i].p$ )` calcule le vecteur normal à la courbe  $B_{PQ}$  au point discret  $DB_{PQ}[i].p$  ;
- `ajouter( $B_{PQ}, R, t$ )` insère un nouveau point de contrôle  $R$  à la courbe  $B_{PQ}$  au temps  $t \in [0,1]$ .

Nous testons notre **Algorithme 16** pour différents types de modèles numériques de terrain. Le seuil est fixé à 0.1 et la limite de temps imposée pour le calcul d'une trajectoire est de 1000 millisecondes.

Un premier test est effectué pour un terrain où les obstacles sont décrits de manière binaire. La figure 4.7(a) montre les trajectoires générées pour le MNT illustré en bas de la figure 4.5 et pour la configuration des sites d'agglomération et le réseau illustrés en figure 4.6. L'algorithme se comporte bien, les routes évitent le cylindre à altitude *pseudo-infinie*, soit une solution en dessous du seuil est trouvée pour chaque arête. L'ensemble du réseau, composé de 257 routes, est calculé cent fois. Nous obtenons une moyenne de temps de calcul de 15.1 secondes avec un écart-type de 4.5 secondes sur notre architecture de référence (cf. section 1.1.1.2).

Dans figure 4.7(b) le MNT décrit des variations moins brutales d'altitudes. Ici aussi la méthode réagit bien, toutes les connexions aboutissent à des routes où le seuil minimal est atteint. L'ensemble du réseau, composé de 226 routes, est aussi calculé cent fois. Nous obtenons une moyenne de temps de calcul de 27.2 secondes et un écart-type de 16.1 secondes. La moyenne est deux fois plus élevée pour un écart-type quatre fois plus grand ; i.e. le temps de résolution est plus contrasté.

---

**Algorithme 16** Modèle stochastique utilisant des courbes de Bézier pour calculer des trajectoires entre deux positions sur un modèle numérique de terrain.

---

**ENTRÉES:** Un modèle numérique de terrain  $\mathcal{T}[H][L]$  et deux points  $P$  et  $Q$  dans le plan décrit par  $\mathcal{T}$ .

**SORTIES:** Une trajectoire, sur un terrain  $\mathcal{T}$ , calculée entre deux points  $P$  et  $Q$ .

$B_{PQ} \leftarrow \text{init\_Bézier}(P, Q)$

etiquette :

$DB_{PQ}[n] \leftarrow \text{discrétiser}(\mathcal{T}[H][L], B_{PQ})$

$B'_{PQ} \leftarrow B_{PQ}$

Trier  $DB_{PQ}[n]$  par coût décroissant

si  $DB_{PQ}[0].c < \text{seuil}$  alors

ARRÊTER

**finsi**

$\sum_{\text{couts}} \leftarrow \sum_{i=0}^{n-1} DB_{PQ}[i].c$  (\*)

**pour tout**  $i \in \{0, 1, \dots, n-1\}$  **faire**

$\vec{N} \leftarrow \text{normal}(B_{PQ}, DB_{PQ}[i].p)$

$\text{tentatives} \leftarrow \text{tentatives}_{\text{max}}$

**répéter**

$d \leftarrow \pm 4 \times n \times X_{\text{ncr}}$

$P' \leftarrow DB_{PQ}[i].p + d \times \vec{N}^t$

ajouter( $B'_{PQ}, P', \frac{DB_{PQ}[i].i}{n}$ )

$DB'_{PQ}[m] \leftarrow \text{discrétiser}(\mathcal{T}[H][L], B'_{PQ})$

si  $(\sum_{i=0}^{m-1} DB'_{PQ}[i].c < \sum_{\text{couts}})$  alors

$B_{PQ} \leftarrow B'_{PQ}$

ALLER À *etiquette*

**finsi**

si (la limite de temps est atteinte) alors

ARRÊTER

**finsi**

$\text{tentatives} \leftarrow \text{tentatives} + 1$

**jusqu'à**  $\text{tentatives} = \text{tentatives}_{\text{max}}$

(\*) Nous utilisons cette notation car la somme des coûts peut être calculée au cours de la discrétisation de la courbe.

**fin pour**

---

#### 4.4.4 Résultats et discussion

La figure 4.8 illustre l'application de l'ensemble des méthodes proposées dans ce chapitre au modèle numérique représentant la Corse. Cette donnée provient du *National Geophysical Data Center* [81], les élévations sont échantillonnées tout les 500 mètres avec une précision, en altitude, d'1 mètre<sup>9</sup>. Ainsi, utilisant seulement cet MNT comme entrée de notre système, nous calculons un modèle d'hydrométrie puis de densité de population tels que décrits dans la section 4.2 et illustrés par les images (a) puis (b) de la figure 4.3. Le modèle de densité est alors utilisé pour générer une configuration de sites d'agglomérations selon une distribution approximant celle des unités urbaines de France métropolitaine calculée selon les données du recensement de la population française de mars 1999. Soit, l'image en haut à gauche de la figure 4.8 montre un mélange de rendu superposant l'hydrométrie et la densité de population calculées pour le MNT de l'île de la Corse. L'image en haut à droite montre les positionnements et dimensions des sites d'agglomération générés ; le modèle décrit une partie suffisamment large du nord de l'île de Sardaigne, ainsi quelques sites y sont placés. Le réseau d'agglomérations est calculé puis simplifié comme décrit en section 4.3.2, l'image en bas à gauche montre ce réseau où les segments hachurés représentent les connexions gardées et les segments en jaune l'ensemble des connexions initiales avant simplification. Enfin, l'image en bas à droite de la figure 4.8 représente le résultat final de l'ensemble du système décrit ici. Nous obtenons la description des routes sous forme de courbes de Bézier. Nous avons ajouté un étiquetage sommaire liée à la moyenne des rangs (soit cette valeur est liée au rayons des sites d'agglomération) des plus importantes extrémités d'une route ; cet moyenne permet d'étiqueter une route en autoroute, coloriée en jaune et rouge, en nationale, coloriée en blanc et rouge ou en départementale, coloriée en rouge. Les liaisons maritimes sont coloriées en bleu.

---

9. Pour cet MNT représentant la Corse, la plus basse altitude enregistrée est de 1 mètre et la plus haute altitude est de 2463 mètres. La réalité de l'île de la Corse est autre, les profondeurs sous-marines sont variables et le relief culmine au *Monte Cinto* à 2706 mètres.

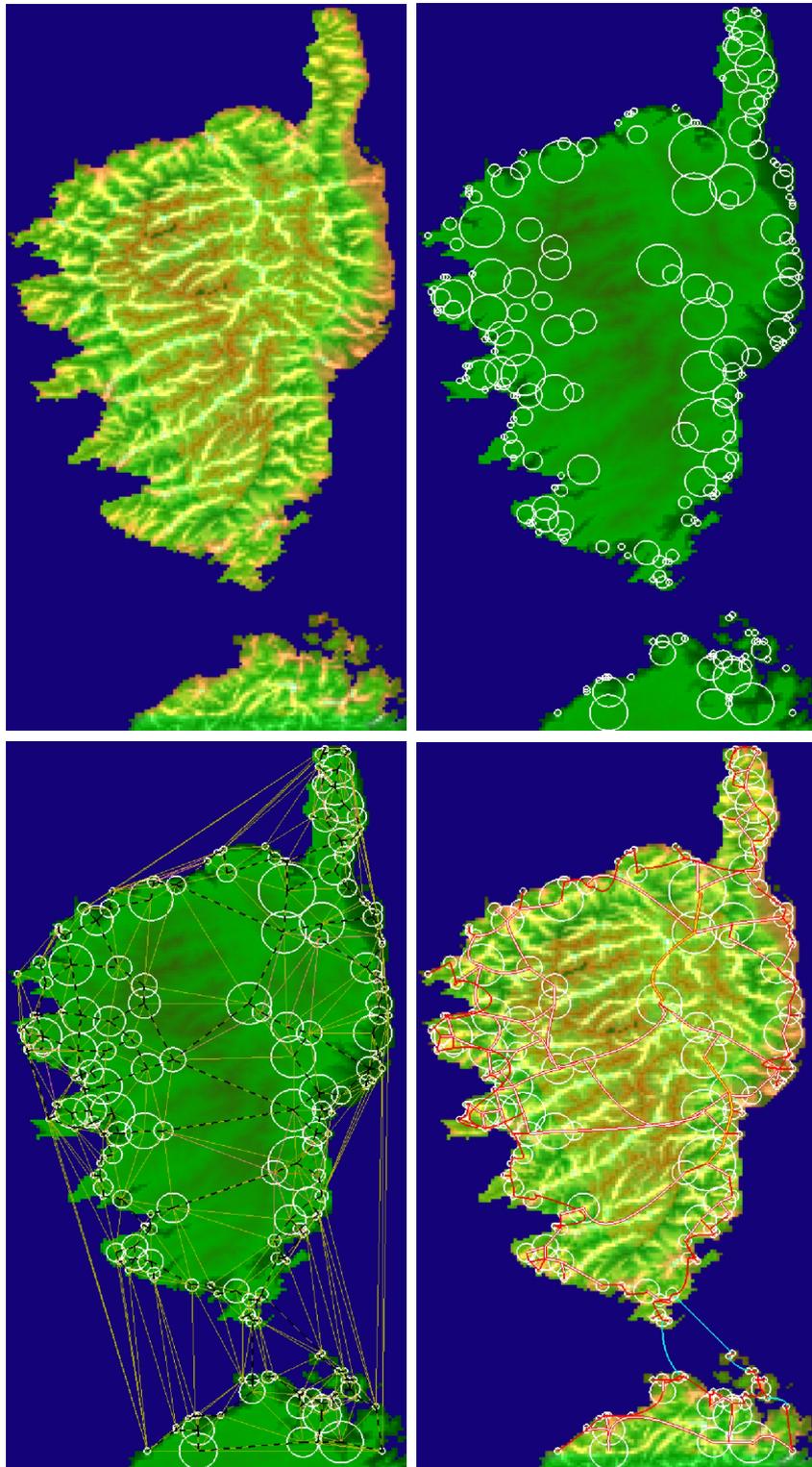


FIG. 4.8 – Génération du modèle d'urbanisation de la Corse.

Enfin, des aspects du travail réalisé dans cette dernière partie restent à explorer et quelques améliorations peuvent être proposées. D'une part, concernant la méthode de planification de trajectoire choisie, l'utilisation des *B-Splines* afin de calculer les routes aurait posé moins de difficultés ; i.e. une modification locale d'un point de contrôle de courbe de Bézier produit des variations dans l'ensemble de la courbe, ces variations sont inversement proportionnelles à la distance temporelle entre le point de contrôle et les points de la courbe. Des *B-Splines* permettraient un raffinement aisé après changement d'échelles. En contre partie, ces courbes garantissent seulement une continuité de la vitesse mais pas de sa dérivée ; il faudra donc réaliser un processus de vérification des trajectoires produites et ainsi les valider puis les étiqueter si elles sont considérées valides. D'autre part, le flux correspondant au trafic dans l'ensemble du réseau routier obtenu doit être étudié afin d'optimiser la catégorisation des routes composant le réseau. Nous pourrions aussi remodeler légèrement le réseau de routes en calculant des jonctions communes ou en dédoublant des axes surchargés.

## 4.5 État de l'art

Depuis moins d'une décennie, la modélisation automatique d'environnements urbains est devenue une des disciplines phares de l'informatique graphique [52, 135, 132, 136].

Les villes sont des structures complexes obtenues par un long processus dans lequel l'homme individuel n'est plus en mesure d'interagir avec son environnement. Les facteurs historiques, sociologiques et culturels sont autant d'acteurs intervenant dans le processus et augmentant arbitrairement sa complexité.

Ainsi, une approche de la modélisation des villes existantes consiste à produire les modèles à partir d'un maximum de données disponibles. La grande majorité des méthodes adoptant cette approche utilisent les techniques de restitution *photogrammétriques*<sup>10</sup> mais les méthodes proposées ne sont généralement que semi-automatiques<sup>11</sup> [53, 51, 126, 120, 18] et dépendent d'un opérateur humain éditant manuellement les modèles ou validant des configurations proposées. Le **CC-Modeler**<sup>TM</sup> (*CyberCity Modeler*) [31, 51, 50, 128] est la

---

10. La photogrammétrie permet de proposer des modèles numériques tridimensionnels d'un objet ou d'une scène à partir de plusieurs images photographiques prises sous différents angles et positions.

11. Certains éléments tels que les toitures [53], les façades d'immeubles [141, 76] ou les routes peuvent être extraits de manière entièrement automatique à partir de données photographiques simples ou stéréoscopiques.

suite logicielle la plus utilisée dans ce type d'approches, elle permet d'extraire des modèles d'objets ou de structures faisant habituellement partie des environnements urbains, par exemple : les immeubles, les résidences, les routes, les arbres, etc. Cet outil est initialement conçu pour traiter des photographies aériennes stéréoscopiques mais des extensions sont proposées afin d'extraire des modèles à partir d'images satellite ou traiter directement des données tridimensionnelles obtenues par un *Scanner 3D*<sup>12</sup> [127, 128]. Dans [49], Gruen et al. proposent d'interfacer le **CC-Modeler** avec des données produites par des senseurs *multi-CCD* (i.e. plusieurs capteurs *Charge-Coupled Device* alignés) et obtiennent des modèles de zones urbaines japonaises et turques : des modèles texturés en haute résolutions de Yokohama et Ginza, un quartier de Tokyo, et un modèle texturé en basse résolutions de la ville d'Izmir.

Une approche légèrement différente est proposée par Sazuki et Chikatsu dans [117] où les auteurs font renaître l'ancien site historique de la ville de Kawagoe<sup>13</sup> à partir des cartes antiques disponibles. Cette approche est très utile pour réaliser des études sur l'évolution des agglomérations selon la disposition des cartes les décrivant et représenterait un outil formidable pour les historiens, néanmoins, une grande partie du traitement à effectuer pour réaliser le modèle reste manuelle.

Dans le système appelé **CityEngine**, Parish et Müller [86] développent leur approche qui consiste à générer des modèles correspondant à des *nouvelles villes* : l'objectif visé par ces travaux n'est pas de reproduire à l'exact le modèle d'une ville existante mais de s'en inspirer<sup>14</sup> pour générer une nouvelle ville. Leur modèle est entièrement basé sur des méthodes procédurales, chacune produit respectivement le plan des rues, le découpage en blocs, les modèles de bâtiments et les textures des façades. Une extension de L-Système est utilisée pour générer les plans des rue et la géométrie des bâtiments selon plusieurs modèles de développement : New York, San-Francisco ou Paris (des combinaisons peuvent être réalisées). Cet L-Système est aidé de données statistiques et de données de relief.

D'autres travaux basés sur des approches procédurales sont spécialisés dans la génération de modèles détaillés de bâtiments à l'architecture moderne [137], ancienne [16] ou les deux à la fois [75]. Il existe aussi, à l'échelle des villes, des approches procédurales spécifiques comme la génération de pavages correspondant aux zones d'habitats précaires (i.e. les bidonvilles)

---

12. Un *Scanner 3D* est un périphérique de numérisation permettant une acquisition de données tridimensionnelles par balayage.

13. Kawagoe est une ville japonaise fondée en 1457 et surnommée la *petite Edo*, i.e. le nom que portait Tokyo à l'époque.

14. Extraire à partir des modèles de villes les spécificités structurelles permettant de reproduire un modèle équivalent.

d'Afrique du Sud [45], ou simplement la génération de villes composées d'immeubles et ayant des pavages relativement réguliers : un projet appelé *Another Manhattan Project* [140] présente un modèle pour la génération automatique de zones urbaines proches du modèle de Manhattan ; dans [47], les auteurs proposent d'explorer une réalité virtuelle où des villes pseudo-infinies sont générées en temps réel, une graine<sup>15</sup>, calculée en fonction d'une position, détermine le modèle du bâtiment implanté en cette position.

Dans [48], les auteurs exposent une approche dans laquelle le modèle urbain généré sert dans la perspective d'un système d'aide à la conduite automobile. Le modèle proposé utilise un système expert générant à partir d'un ensemble de règles la structure des ruelles, les blocs, la signalisation routière et le marquage au sol.

Enfin, dans [116], les auteurs proposent une méthode produisant interactivement des modèles de réseaux routiers à grande échelle (l'échelle d'une région) utilisant des motifs réguliers, concentriques ou basés sur des cartes de population.

---

15. Une graine est une valeur initiale déterminant la séquence de nombres obtenue dans une série dite de nombres pseudo-aléatoires.



## Chapitre 5

# Conclusion et projets

Nous avons proposé des méthodes rapides pour la génération de modèles de terrain naturel et urbain. Notre modèle fractal, le déplacement des milieux morphologiquement contraint, gère aussi bien les contraintes locales serrées, données comme altitudes imposées au relief à générer, que les contraintes globales modifiant l'allure de ce même relief. Notre méthode des écoulements permet de produire des modèles-squelettes décrivant des lignes de crête et l'hydrologie de surface correspondante : le DMMC crée, à partir de ces modèles-squelettes, des modèles de terrain érodé. Nous avons conçu une chaîne de traitement entièrement automatique urbanisant le relief naturel à partir du seul MNT. Différentes configurations multi-urbaines sont obtenues en modifiant quatre pondérations<sup>1</sup> et une fonction de distribution<sup>2</sup> : nous avons proposé un nouvel algorithme rapide, basé sur les courbes de Bézier, pour calculer, en prenant en compte les contraintes du relief, les trajectoires (i.e. les routes) entre les unités urbaines.

Nous développons, dans ce chapitre, le champ d'application de nos méthodes. Nous présentons : des rendus réalistes de modèles de terrain entièrement générés et de modèles construits à partir de squelettes de crêtes obtenus à l'aide d'une description de L-Système ou en utilisant un croquis dessiné par l'utilisateur. Nous présentons des extensions dans la synthèse de textures : un coloriage du squelette du précédent L-Système et un effet de relief de type crépi à partir du précédent croquis sont réalisés. Nous présentons nos résul-

---

1. Pour obtenir différentes cartes de densité de population, nous pondérons différemment les quatre caractéristiques des unités de terrain urbanisé précédemment présentées en section 4.2.1.

2. Nous avons approximé la courbe statistique du recensement de la population française, effectué par l'INSEE en mars 1999 [55] (c'est le dernier recensement concernant toute la population en même temps), pour calculer une distribution de rayons de sites agglomérations (cf. section 4.3.1.1).

tats dans le domaine de la reconstruction, à partir de données partielles, de reliefs naturels relevés par satellite : nous utilisons un sous-échantillonnage aléatoire ou une détection de creux et de crêtes pour appauvrir la donnée initiale dans l'objectif de la reconstruire puis de comparer le modèle original au modèle reconstruit. Une extension est proposée dans le domaine de la reconstruction de photographies où un effet non-photoréaliste est obtenu. Nous terminons par l'urbanisation automatique d'un relief généré et nous discutons sur les perspectives de nos travaux.

## 5.1 Le générateur de modèles

Les cartes de hauteur générées peuvent être utilisées comme une donnée servant à créer des maillages réguliers représentant les reliefs des paysages ou bien servir à créer des textures en les utilisant comme des indices dans une table d'indirection de couleurs (i.e. une *Color Look Up Table*) ou encore donner du relief à une surface en utilisant ces valeurs comme une carte de relief (i.e. *Bump mapping*).

### 5.1.1 Le générateur de terrains

Nous présentons, par l'intermédiaire des trois prochaines figures (les figures 5.1, 5.2 et 5.3), des rendus en lancer de rayon de modèles générés par notre algorithme DMMC selon trois différentes contraintes initiales :

1. Le modèle rendu<sup>3</sup> sur la figure 5.1 est généré à partir d'un modèle-squelette obtenu par la méthode des écoulements.
2. Le modèle dont le MNT est donné par l'image en haut à droite de la figure 5.2 est généré à partir du modèle squelette de l'image en haut à gauche de la même figure : nous utilisons un L-Système<sup>4</sup> [99] pour générer ce squelette, les branches décrivent les crêtes dont l'altitude est décroissante en fonction de la profondeur de la pile dans l'interprète réalisant le rendu i.e. la tortue. Le rendu en lancer de rayon est donné sur l'image du bas de la figure 5.2.

---

3. Nous avons exporté notre modèle au format `ter` puis utilisé le logiciel *Terragen*<sup>TM</sup> [124] pour l'importer et effectuer son rendu.

4. Le L-Système utilisé pour créer ce squelette est décrit par :

```

 $\theta_0$  ←  $\frac{\pi}{2}$ 
 $\theta$  ←  $\frac{\pi}{25}$ 
 $\omega$  ← ++++++X
X ← F[@.5+++++X] -F[@.4-----!X]@.6X

```

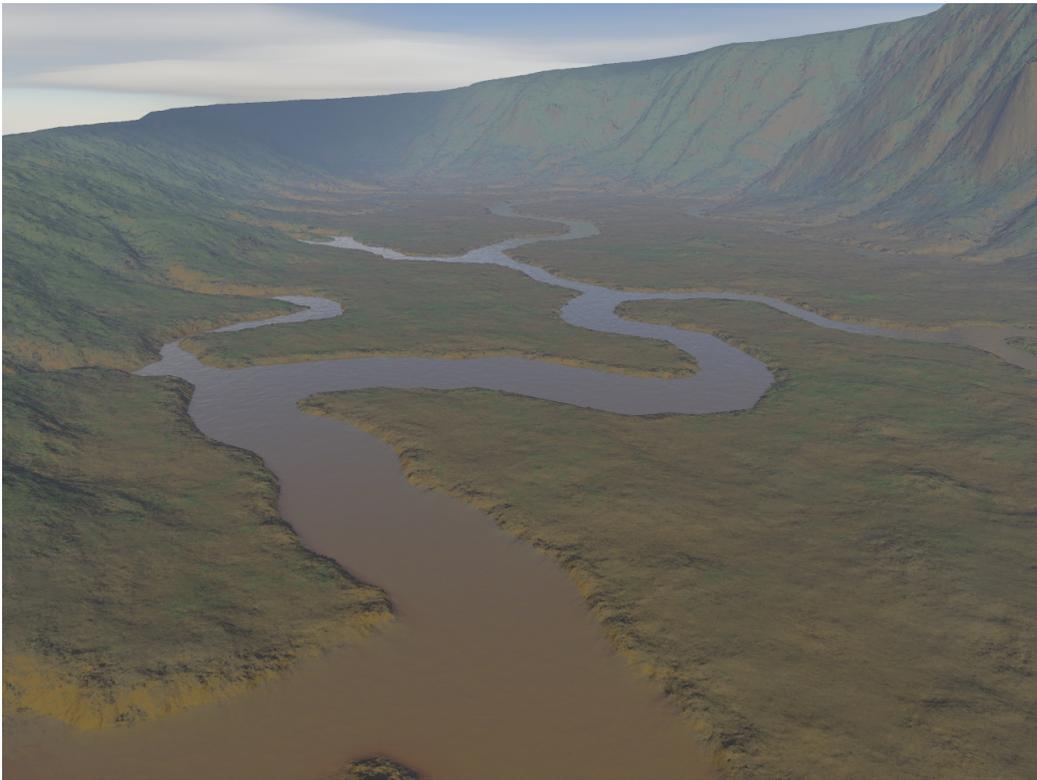


FIG. 5.1 – *Modèle de terrain entièrement généré : notre algorithme DMMC génère le terrain à partir d'un modèle-squelette produit par notre méthode des écoulements.*

3. Les modèles rendus<sup>5</sup> en figure 5.3 sont générés à partir d'un même croquis réalisé avec un éditeur d'images : une image, en niveaux de gris (8 ou 16 bits), décrit des contraintes d'altitudes imposées aux pixels dont l'intensité lumineuse de la couleur est non nulle ; le contour du continent africain est dessiné en plus du texte *Afrigraph'07* en référence à la conférence internationale du même nom<sup>6</sup> ; l'image du haut est obtenue en étiquetant *riviere* (i.e. extrapolation vers le haut) les cellules du MNT-squelette ; l'image du bas est obtenue en étiquetant *crete* (i.e. extrapolation vers le bas) les cellules du MNT-squelette.

---

5. Nous avons exporté nos modèles au format png (*Portable Network Graphics, iso-15948*) et utilisé le logiciel *Povray* [97] pour réaliser le rendu.

6. Ces images sont tirées de notre dernière publication [7], deux de nos paysages ont été également choisis pour être présents sur la couverture des actes de la conférence.

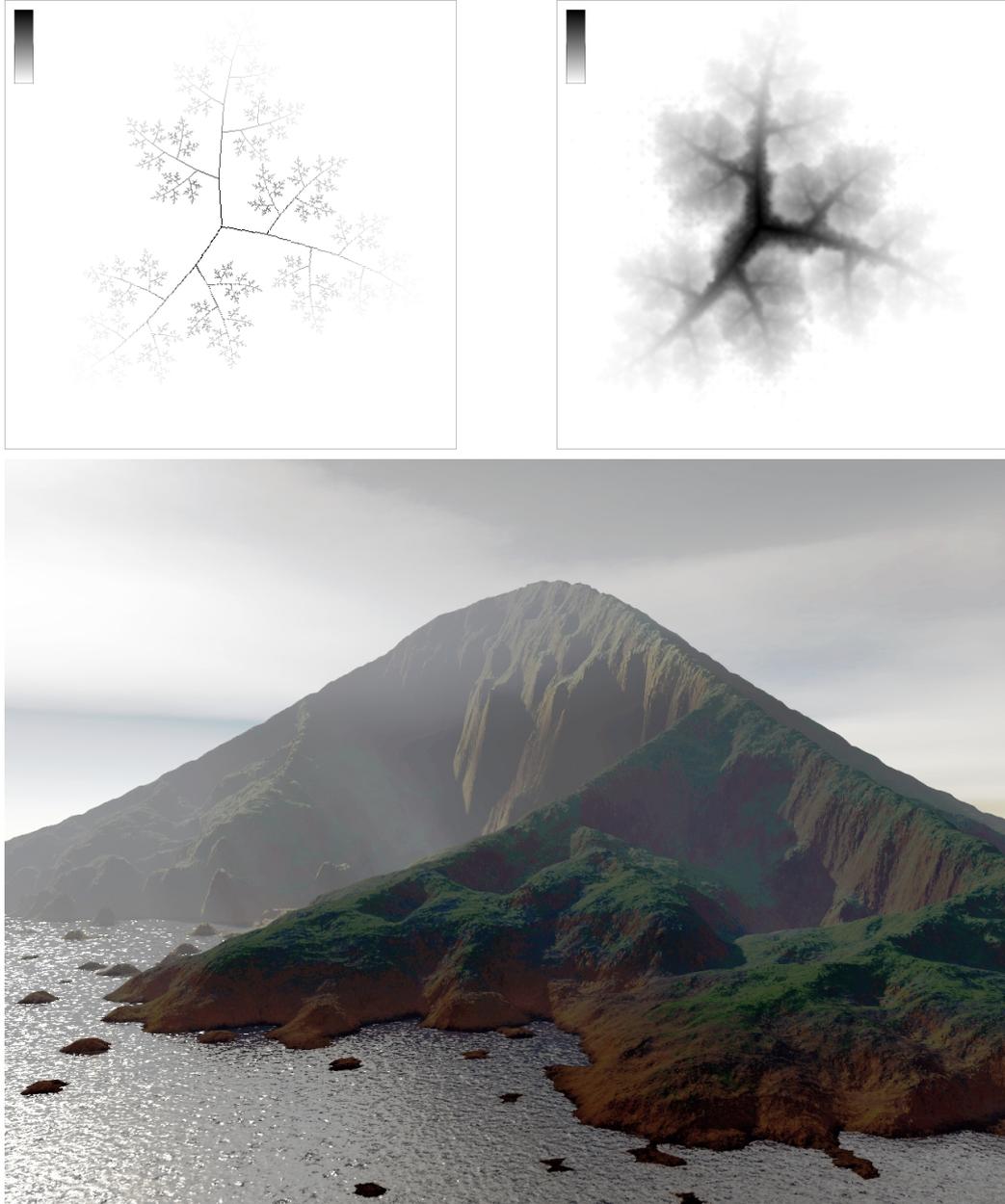


FIG. 5.2 – *Modèle de terrain entièrement généré: notre algorithme DMMC génère le terrain à partir d'un squelette de lignes de crêtes décrit par un L-Système.*



Nous rappelons au lecteur qu'il s'agit là de paysages imaginaires générés par notre algorithme DMMC, et non une image obtenue par photo aérienne d'un paysage existant en Afrique.

FIG. 5.3 – Modèles de terrain générés par l'algorithme DMMC à partir d'un croquis donné dans une image.

### 5.1.2 Le générateur de matériels

Nous reprenons les deux derniers modèles-squelettes, vus en section 5.1.1, et les modifions afin de générer du *matériel* pour le coloriage ou le rendu de surfaces. Les valeurs d'altitudes du squelette généré à l'aide du L-Système sont inversées, nous sélectionnons, à l'aide d'un éditeur d'images, une zone à distance fixe du squelette et contraignons le complémentaire de cette zone à l'altitude la plus élevée : l'image à gauche de la figure 5.4 est obtenue<sup>7</sup>. Nous colorions, à l'aide de notre algorithme DMMC et d'une *Color LUT* la zone autour du squelette afin d'obtenir l'image à droite de la figure 5.4.

Pour notre second exemple de matériel, nous utilisons le précédent croquis auquel nous ajoutons le logo d'*Afrigraph* [2] et un effet *poivre et sel*<sup>8</sup> à 1%, cf. l'image à gauche de la figure 5.5, pour générer en utilisant l'algorithme DMMC aidé d'un filtre médian<sup>9</sup> [22] une carte de relief dont le MNT est donné sur l'image à droite de la figure 5.5. Un effet de peinture au crépi est obtenu par le rendu<sup>10</sup> en tracer de rayon donné en figure 5.6.

---

7. Nous rappelons au lecteur que nous avons choisi de représenter un MNT par son négatif.

8. Le nom *poivre et sel* est donné au bruit apparaissant de manière uniforme au niveau d'une image en niveaux de gris.

9. Le filtre médian est généralement utilisé pour supprimer l'effet poivre et sel.

10. Nous avons exporté la carte au format `png` (*Portable Network Graphics, iso-15948*) et utilisé le logiciel *Povray* [97] pour réaliser le rendu du mur dont le relief est décrit par cette carte.

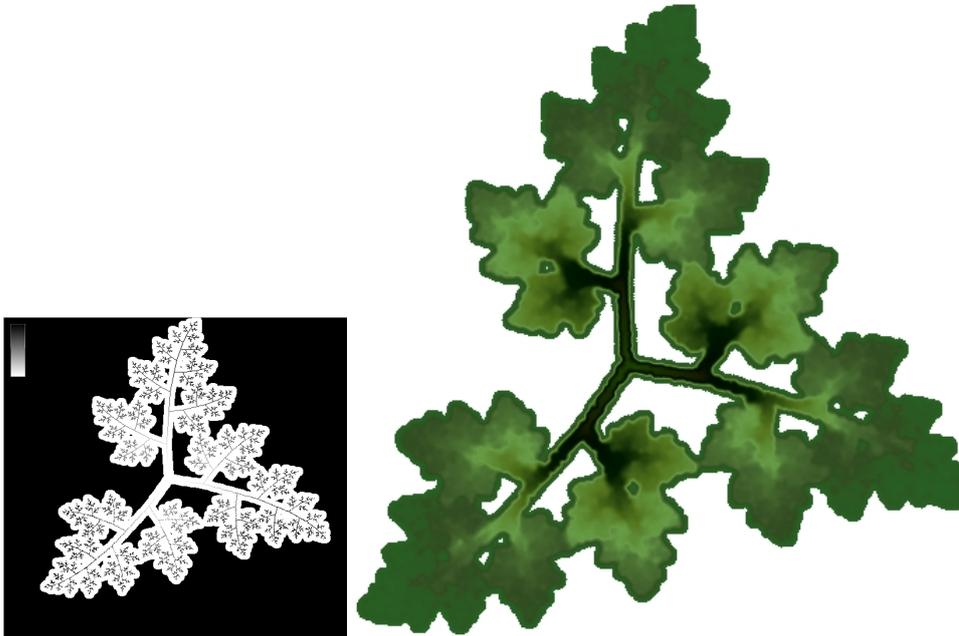


FIG. 5.4 – Une texture de feuille.

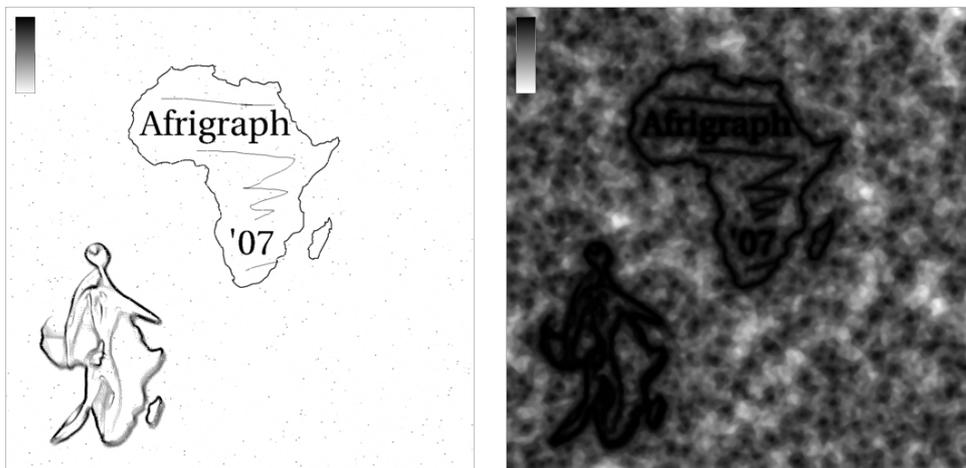


FIG. 5.5 – Un effet de peinture au crépi: le MNT de la surface est obtenu en appliquant l'algorithme DMMC à partir d'un croquis. Le rendu est donné en figure 5.6.

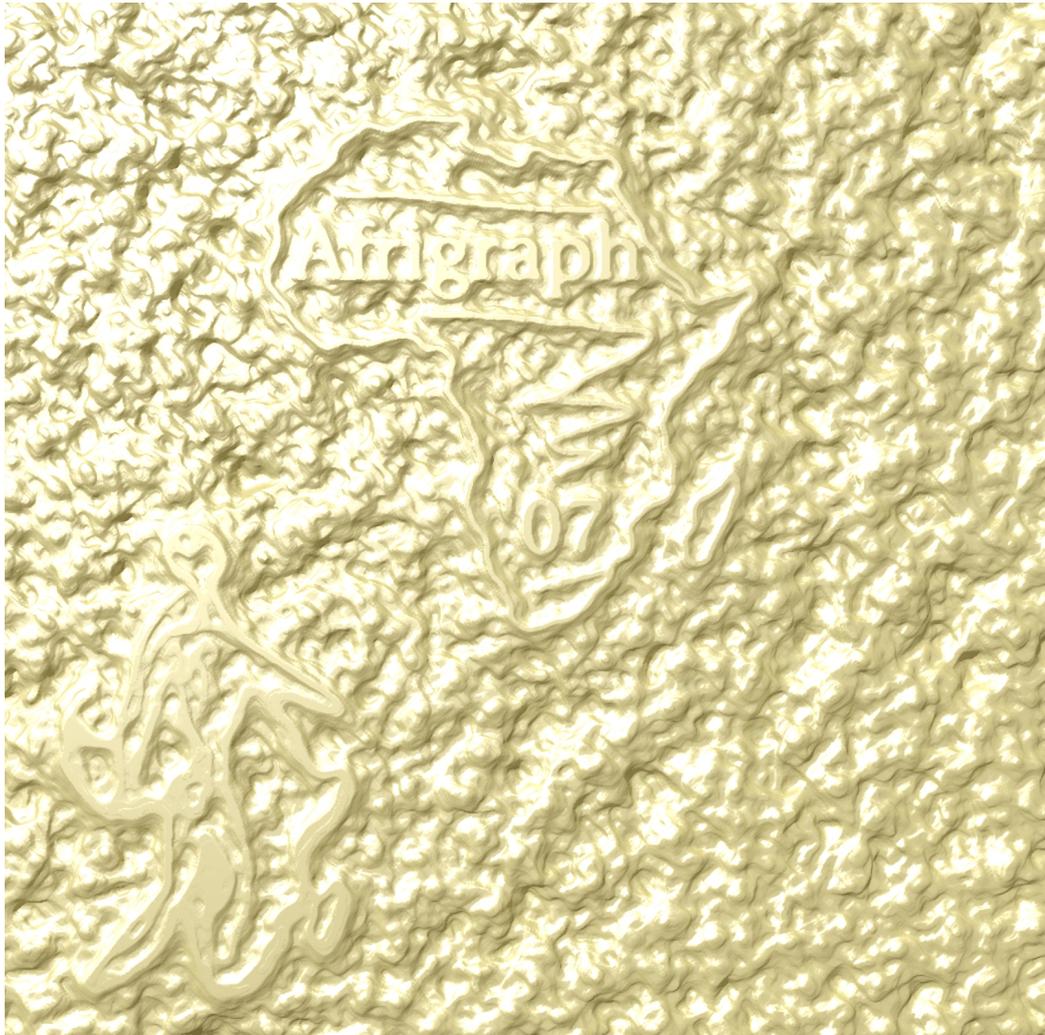


FIG. 5.6 – *Un effet de peinture au crépi: la surface est obtenue en appliquant l'algorithme DMMC à partir d'un croquis.*

## 5.2 La reconstruction des surfaces

Nous appliquons notre algorithme DMMC à la reconstruction de données satellites partielles.

### 5.2.1 La reconstruction à partir d'un sous échantillonnage aléatoire

Ici, nous comparons nos résultats à ceux obtenus par J. Pouderoux et al. dans [95] où une approche par réseaux de neurones est utilisée pour approximer une donnée satellite aléatoirement sous-échantillonnée à hauteur de 3.3% de l'ensemble initial. La donnée satellite utilisée est un MNT de l'USGS [129] représentant le Mont Washington (état du New Hampshire, USA). Elle comporte 1310813 ( $941 \times 1393$ ) altitudes, chacune est exprimée sur 2 octets, le pas vertical (i.e. une unité verticale) est de 0.019836 mètres et nous avons 58405 échantillons différents ; cette donnée est alors sous-échantillonnée à 43256 altitudes et nous reconstruisons l'ensemble en utilisant notre algorithme DMMC.

Le MNT reconstruit est comparé à l'original en utilisant un calcul d'écart-type : nous appliquons les mêmes conditions de test que celles utilisées dans [95] et nous obtenons<sup>11</sup> nos résultats en deux étapes. Le premier modèle reconstruit par l'algorithme DMMC est généré en 1949 millisecondes (1735 millisecondes pour le processus bas-haut et 214 millisecondes pour le processus haut-bas) et donne un écart-type<sup>12</sup> de 7.65 mètres. La variance (donc l'écart-type) étant proportionnelle aux carrés des distances (i.e. les différences), un lissage de la surface obtenue peut nous permettre de réduire cette quantité<sup>13</sup>. Nous appliquons en boucle, tant que l'écart-type est décroissant (13 itérations pour cet exemple), un filtre passe-bas  $3 \times 3$  modifiant les altitudes générées, nous obtenons une amélioration du résultat final donnant un nouvel écart-type de 5.92 mètres pour un surcoût en temps de 1159 millisecondes. Ainsi, avec notre approche, nous reconstruisons la surface originale en 3109 millisecondes donnant 5.92 mètres d'écart-type contre, en comparaison avec

---

11. Seulement pour cet exemple, nous utilisons une architecture à base de Pentium IV 4 Ghz avec 1 Go de mémoire : il était nécessaire, pour effectuer la comparaison de notre méthode avec celle présentée dans [95], de normaliser les mesures en faisant tourner nos algorithmes sur des unités centrales identiques ou comparables.

12. Pour comparaison, la donnée sous-échantillonnée à 3.3%, où toutes les autres altitudes sont nulles, donne un écart-type de 535.39 mètres par rapport à la donnée originale.

13. L'application seule du filtre passe-bas à partir de la donnée sous-échantillonnée, et modifiant seulement les altitudes non connues, converge très lentement (après 2 minutes de calcul et 1409 itérations) vers une surface donnant un écart-type de 7.79 mètres par rapport à la surface originale.

la méthode proposée dans [95], une surface reconstruite en 531 secondes et donnant un écart-type de 5.04 mètres. Enfin, la figure 5.8 illustre le résultat obtenu par notre méthode en donnant successivement le MNT original, le MNT sous-échantillonné et le MNT reconstruit. Une comparaison visuelle entre une vue tridimensionnelle du modèle reconstruit dans [95], du modèle original et du modèle reconstruit par notre approche est proposée en figure 5.8.

Du point de vue de la complexité, nous devons séparer les deux processus mis en œuvre dans le cas de la méthode DMMC ; i.e. le processus bas-haut (extrapolations) et le processus haut-bas (interpolations par subdivisions récursives). Ainsi, dans le cas du processus haut-bas nous avons déjà, en section 1.1.1.1, établi sa complexité à un ordre  $\mathcal{O}(n)$  où  $n = L \times H$  soit les dimensions du MNT ; cela implique, pour des MNTs de même résolution, que la variation de la complexité est uniquement liée au processus bas-haut. Cette variation est illustrée par la figure 5.9 : le processus bas-haut possède une complexité croissante en fonction des contraintes d'altitudes imposées, cette complexité commence à diminuer quand le nombre de contraintes devient supérieur au nombre d'altitudes à calculer (i.e. les altitudes non imposées). Ainsi, un pire cas peut être obtenu quand toutes les feuilles (i.e. les nœuds terminaux se trouvent en position à coordonnées impaires pour le cas particulier où  $W = 2^n + 1$  et  $H = 2^m + 1$ ) de l'arbre de subdivision ont une altitude imposée (i.e. une cellule à état non-connu) à l'inverse de tous les autres nœuds de l'arbre ; dans ce cas la complexité est approximativement égale à  $\mathcal{O}(C \times (\log_4 N)^2)$  où  $C \approx \frac{N}{4}$  dénote le nombre d'altitudes imposées.

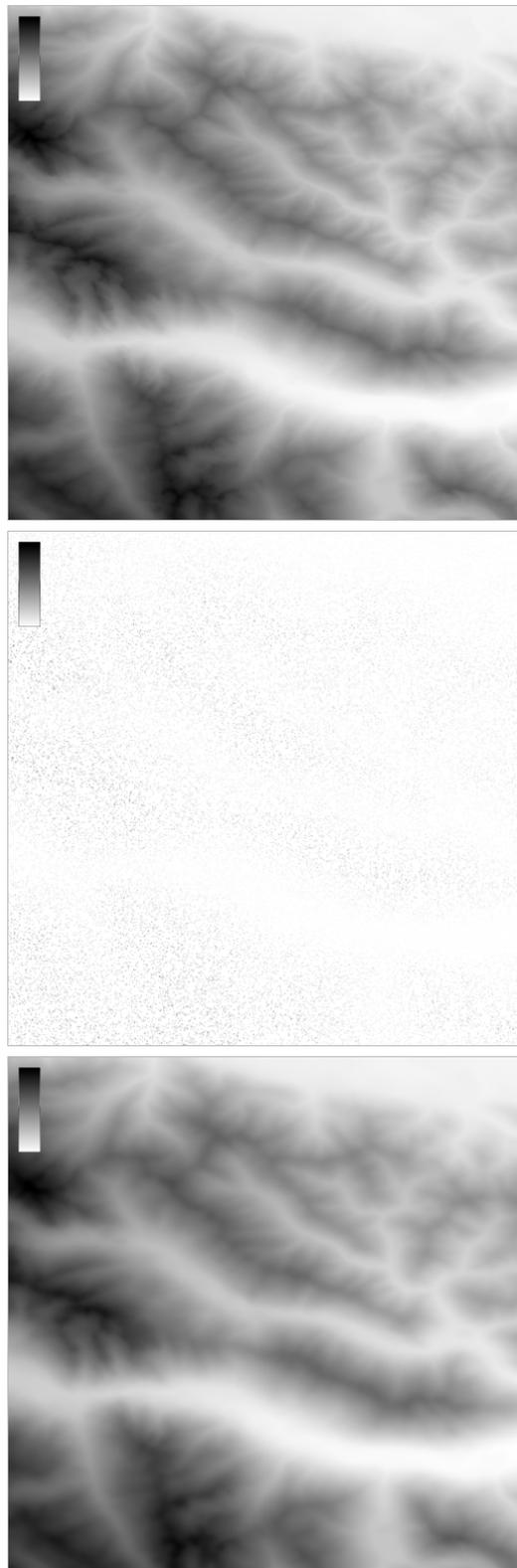


FIG. 5.7 – *Sous-échantillonnage puis reconstruction de la donnée satellite représentant le Mont Washington, de haut en bas: le MNT original; son sous-échantillonnage aléatoire à 3.3%; le MNT reconstruit à l'aide de notre algorithme DMMA à partir de la donnée sous-échantillonnée.*

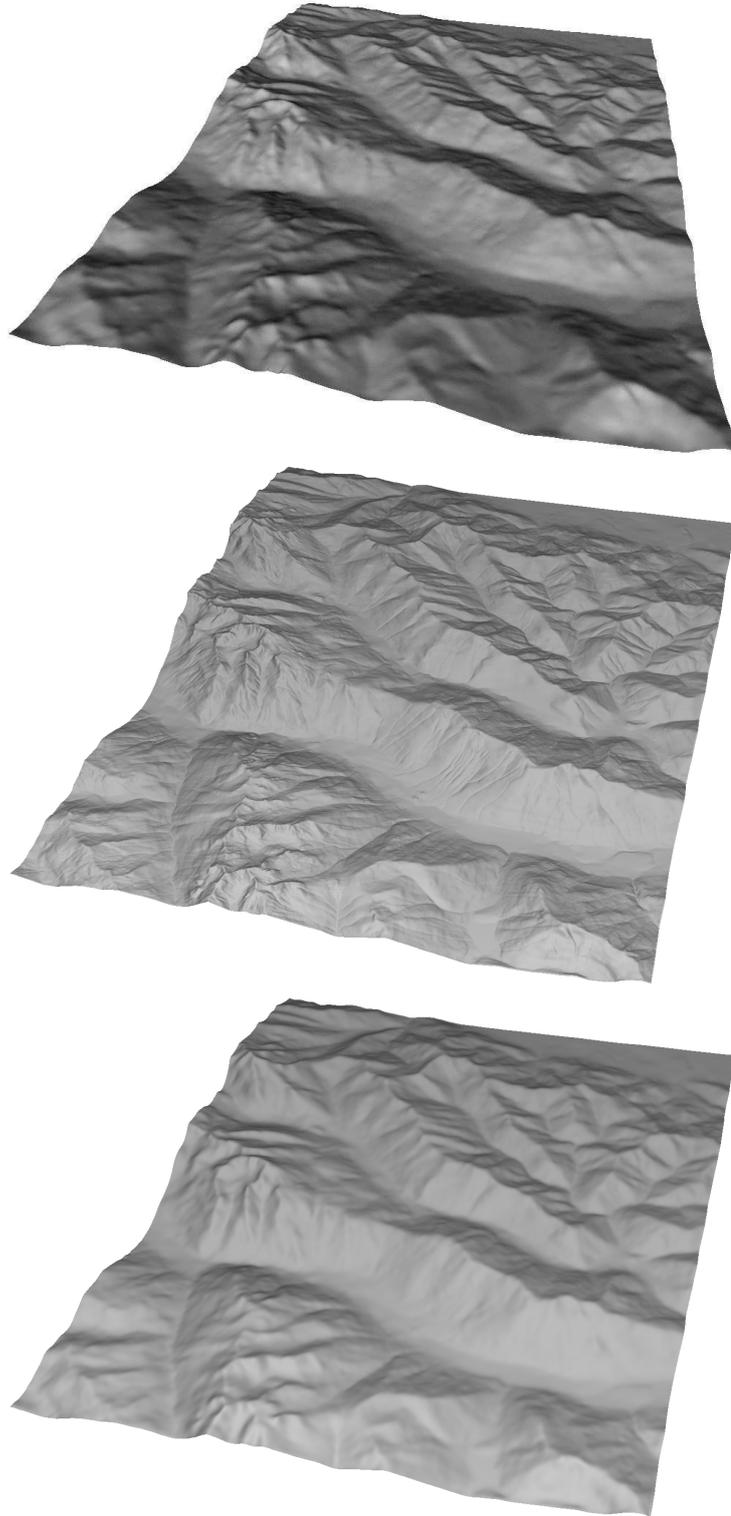


FIG. 5.8 – *Comparaison visuelle des modèles du Mont Washington. Sont illustrés de haut en bas : le modèle reconstruit dans [95] à partir du sous-échantillonnage aléatoire à 3.3% du modèle original ; le modèle original ; le modèle reconstruit à l'aide de notre algorithme DMMC à partir de la donnée aléatoirement sous-échantillonnée à 3.3%.*

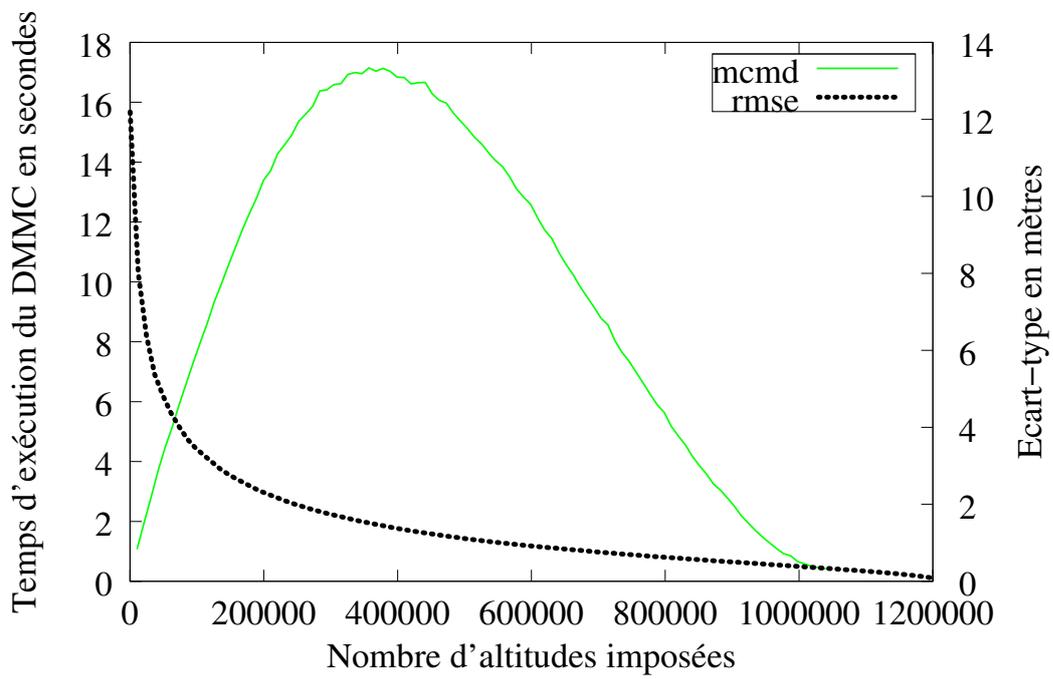


FIG. 5.9 – Variation du temps et de l'écart-type en fonction du nombre d'échantillons initiaux.

### 5.2.2 La reconstruction à partir des contours

Nous proposons à présent une application permettant de reconstruire à partir des contours d'une surface donnée la surface originale. Nous empruntons, pour détecter les contours, une partie des calculs réalisés dans la section 4.2.2.1. Nous utilisons les huit vecteurs associés à chaque position et notés  $\vec{V}_i$  pour calculer une pondération<sup>14</sup>, positive ou négative, exprimant la forme du relief en cette position. Ainsi, cette dernière peut représenter à plusieurs niveaux d'intensité : un pic, une crête, un plan, un creux ou un trou et il est évident que ramener ces états à une seule valeur numérique comprise dans l'intervalle  $[-1,1]$  supprime une grande partie de l'information. Nous proposons un processus permettant de suivre à partir d'une position pondérée un chemin correspondant aux pondérations équivalentes à  $\pm 10\%$  de la précédente position et dans lequel nous nous interdisons le déplacement d'un cavalier aux échecs. Nous illustrons cette méthode sur le MNT<sup>15</sup> de l'USGS [129] du Mont Saint Helen (état de Washington), les résultats sont obtenus sur notre architecture de référence (cf. section 1.1.1.2), les deux images de droite en figure 5.10 expriment respectivement de haut en bas une détection de creux (i.e. sont utilisées les positions ayant les plus petites pondérations) ainsi qu'une détection de creux et de crêtes (i.e. sont utilisées les positions ayant les plus grandes valeurs absolues de pondérations); chacune est approximativement réalisée en 3.7 secondes à partir du MNT original se trouvant en haut à gauche de la figure. Nous obtenons l'image en bas à gauche à partir des creux et des crêtes et par reconstruction à l'aide de l'algorithme DMMC : l'entrée représente approximativement 18% de la donnée initiale et le MNT reconstruit, en 6483 millisecondes, possède un écart-type de 3.19 mètres par rapport à l'original. Enfin, la figure 5.11 montre une vue tridimensionnelle du modèle original ainsi que le modèle reconstruit.

---

14. Nous calculons la moyenne des produits scalaires de chaque  $\vec{V}_i$  avec le vecteur unitaire de la composante d'altitude (0,0,1).

15. Cet MNT est de dimensions  $975 \times 1399$  et chaque altitude est encodée sur 2 octets pour un total de 10474 échantillons différents.

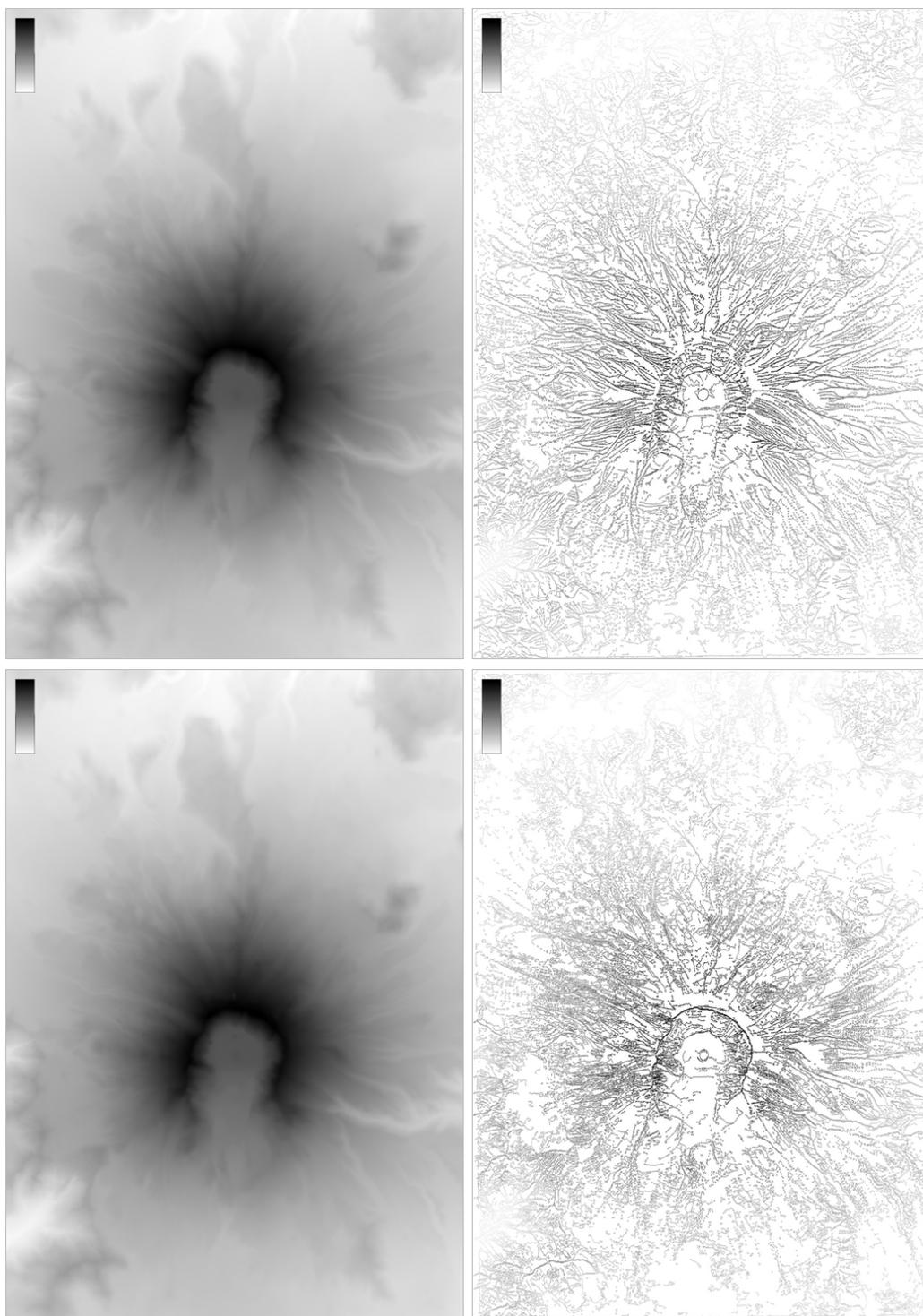


FIG. 5.10 – Reconstruction du MNT du Mont Saint Helen (état de Washington) à partir du squelette du modèle. Sont illustrés, de l'image en haut à gauche et dans le sens trigonométrique inverse: le MNT original; une détection de creux; une détection de crêtes et de creux; le MNT reconstruit à partir des crêtes et des creux à l'aide de notre algorithme DMMC.

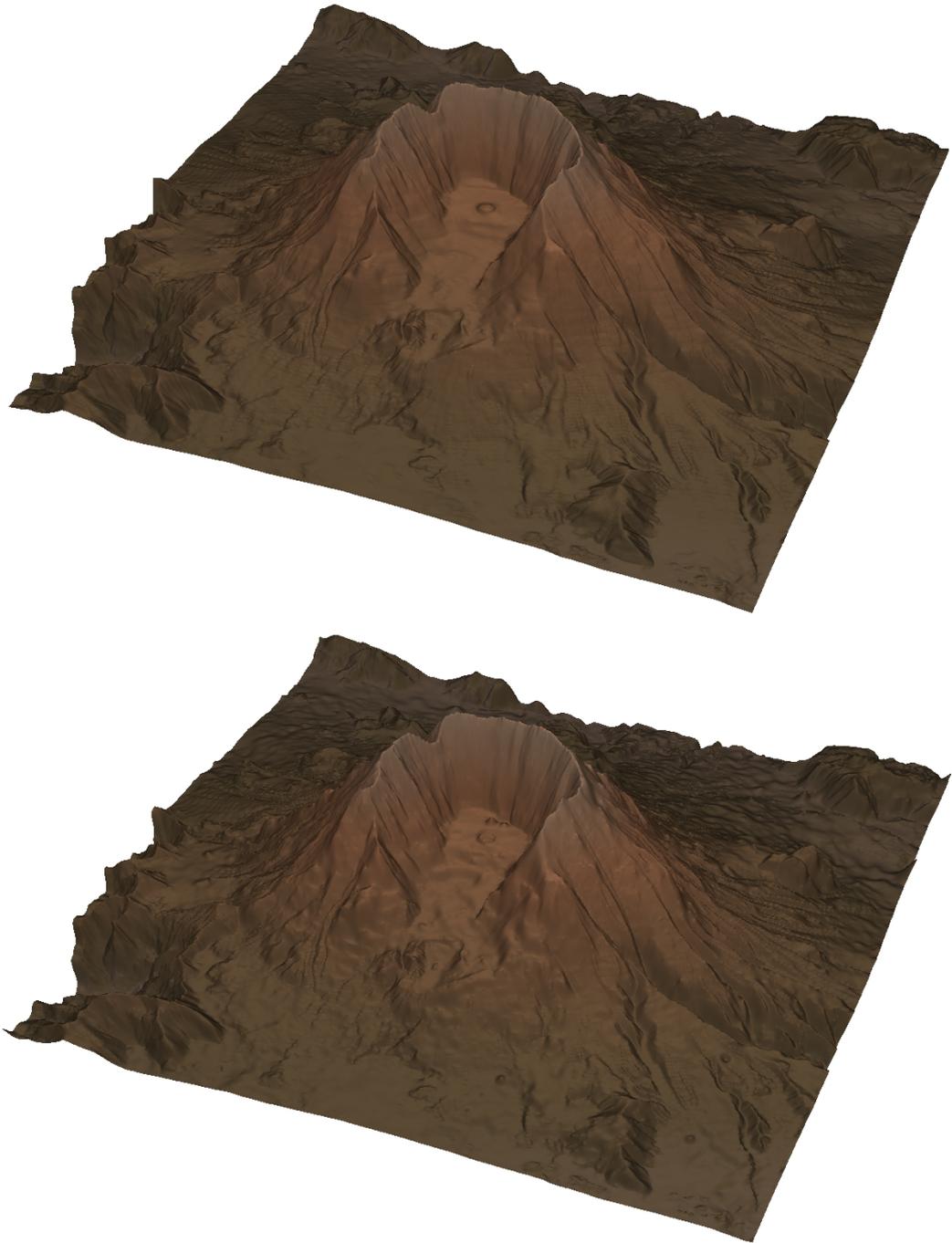


FIG. 5.11 – *Comparaison visuelle entre le modèle original du Mont Saint Helén et le modèle reconstruit (cf. figure 5.10) à l'aide de l'algorithme DMMC.*

### 5.3 De la peinture impressionniste aux planètes artificielles : l'univers dans une graine

Nous utilisons les résultats obtenus en section 5.2.2 pour proposer des rendus non-photoréalistes de photographies<sup>16</sup>. En effet, nous pouvons considérer une photographie, même en couleur, comme une surface ou bien une composition de surfaces. Deux possibilités sont proposées : soit nous décomposons<sup>17</sup> l'images en *HSV* et réalisons le traitement en utilisant la composante d'intensité puis recomposons l'image, soit nous décomposons l'image en *RGB* et réalisons le traitement sur chacune des composantes avant de recomposer l'image. La figure 5.12 illustre ces deux démarches réalisées à partir de la photographie originale présentée en haut à gauche de la figure, nous avons, dans le sens trigonométrique inverse : une détection de crêtes sur la composante rouge de l'image, une reconstruction à partir des crêtes de chacune des composantes puis recombinaison (i.e. dans ce cas l'algorithme DMMC est utilisé 3 fois), une détection de creux et de crêtes sur la composante intensité de l'image, et enfin deux différentes reconstructions ( $\mathcal{H} = 1$  et  $\mathcal{H} = 0.45$ ) réalisée à l'aide de l'algorithme DMMC à partir de la composante intensité.

Nous obtenons les résultats illustrés par les figures 5.13 et 5.14 en réalisant une reconstruction à partir des contours, creux et crêtes, détectés sur la composante intensité de chacune des images originales (i.e. en haut à gauche en figure 5.13 et en haut pour la figure 5.14). Nous utilisons, dans les paramètres de l'algorithme DMMC, une petite valeur pour le paramètre de Hurst,  $\mathcal{H} \approx 0.35$ , notre algorithme est suivi d'un post-traitement (car dans ce cas la surface générée est très bruitée) effectuant une passe de filtre médian  $7 \times 7$  afin d'obtenir les résultats définitifs.

Pour finir, nous illustrons notre modèle d'urbanisation automatique obtenu pour un terrain généré par l'algorithme DMMC. Le terrain est contraint par un MNT-squelette créé dans un éditeur d'images et représentant deux hauts plateaux formant l'acronyme IA<sup>18</sup> et l'allure de la surface générée est contrainte vers le bas ( $I_{-1} = 6$ ) lors de l'extrapolation : la figure 5.15 montre le modèle-squelette ainsi que le modèle généré. Pour ce dernier, les figures 5.16, 5.17 et 5.18 illustrent l'ensemble de son processus d'urbanisation.

---

16. Notre première démarche était d'essayer d'appliquer l'algorithme DMMC à la reconstruction d'images à partir de données partielles dans une perspective de compression de données.

17. Une décomposition revient à décrire une couleur dans un espace de représentation à plusieurs dimensions, il existe plusieurs modèles de couleurs [17, 42] et nous utiliserons, dans nos traitements, les modèles *RGB* (*rouge, vert et bleu*) ou *HSV* (*teinte, saturation et intensité*).

18. IA est l'acronyme de Intelligence Artificielle.



FIG. 5.12 – Application de la reconstruction des surfaces à partir des contours aux photographies : les composantes RGB ou l'intensité lumineuse de la couleur sont considérées comme des reliefs.



FIG. 5.13 – *Un coucher de soleil à l'Escalade (Catalogne) : reconstruction de la photographie à partir des contours puis redimensionnement de l'image et égalisation des couleurs.*

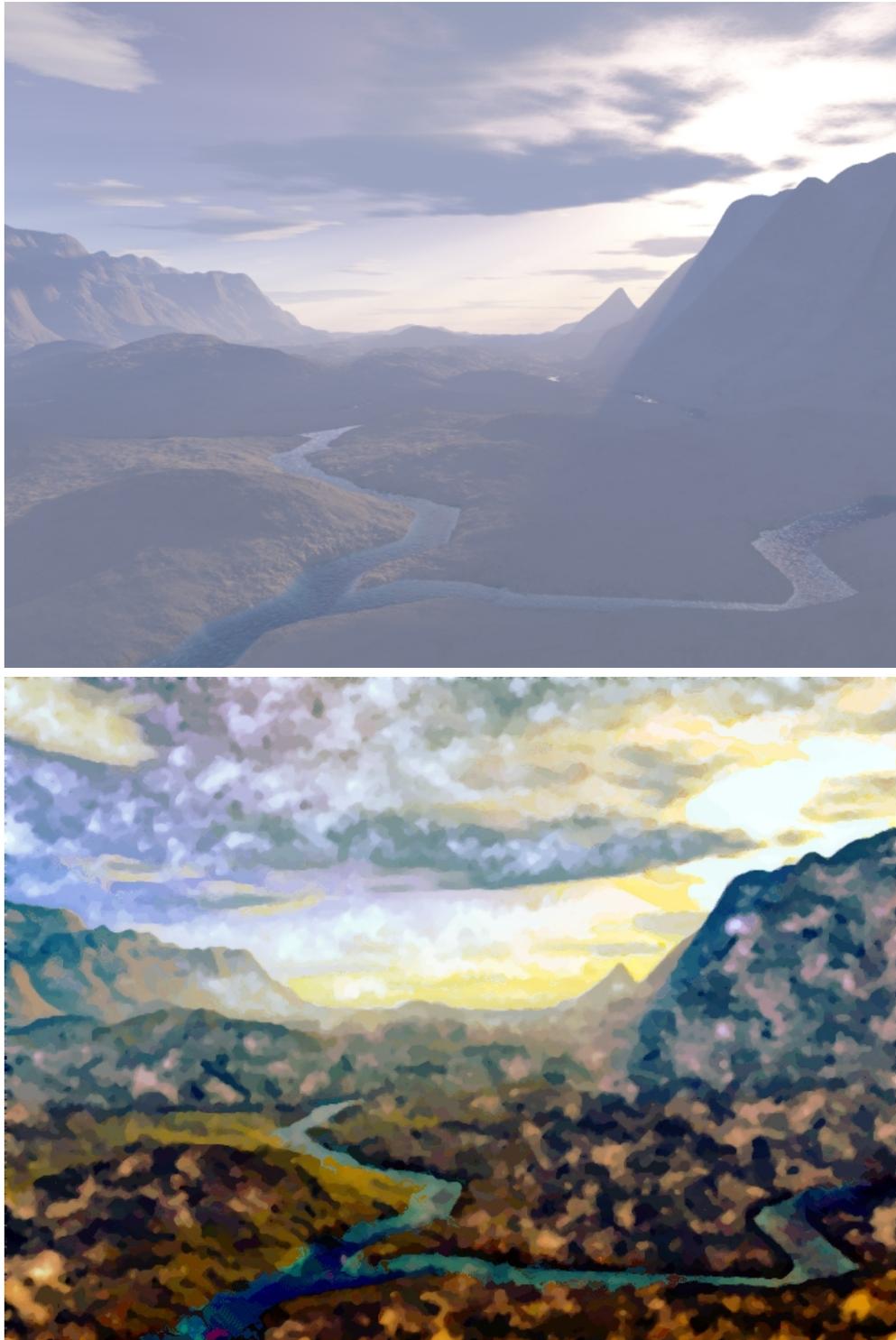


FIG. 5.14 – *Peinture entièrement procédurale.*



FIG. 5.15 – *Contrainte et modèle de terrain correspondant : le Mont IA.*

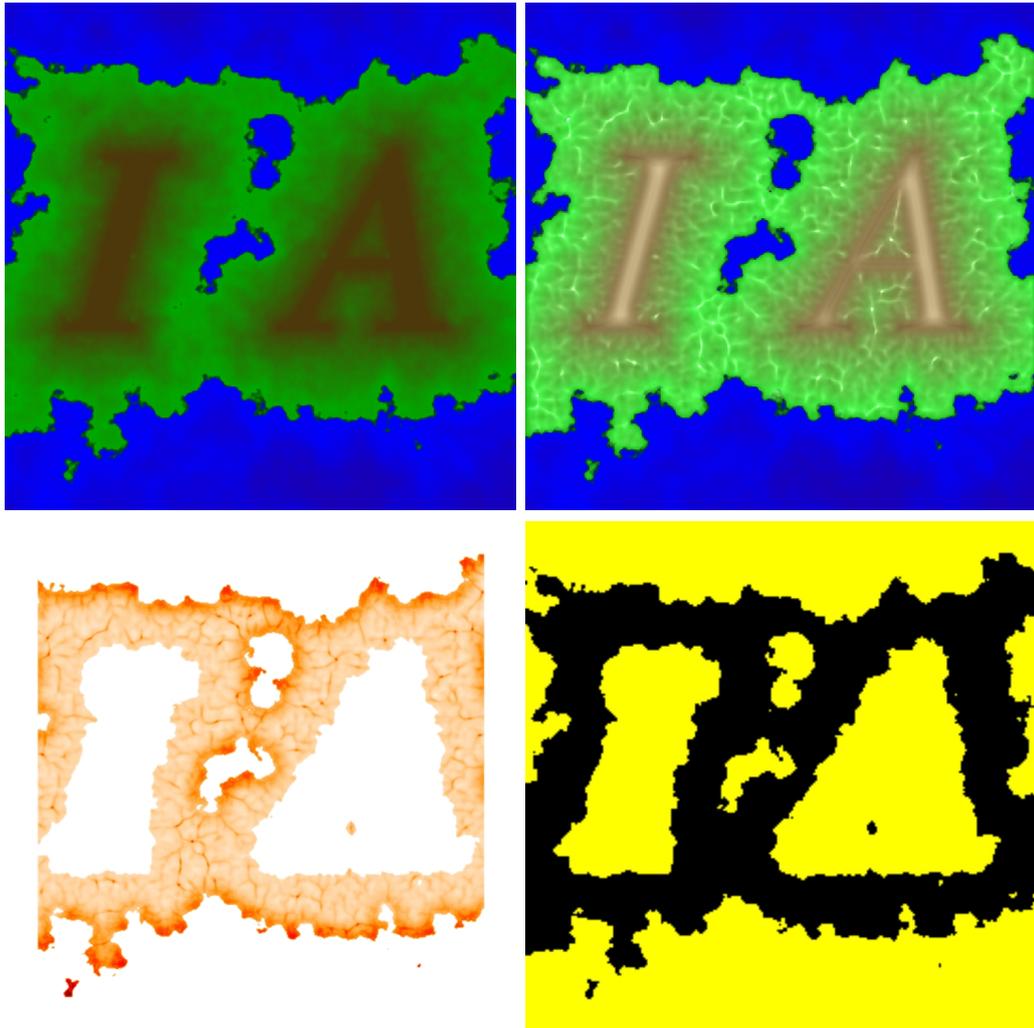


FIG. 5.16 – Urbanisation du Mont IA, sont illustrés, d'en haut à gauche et dans le sens trigonométrique inverse: le MNT; son hydrométrie; les zones inhabitables; la carte de densité de population.

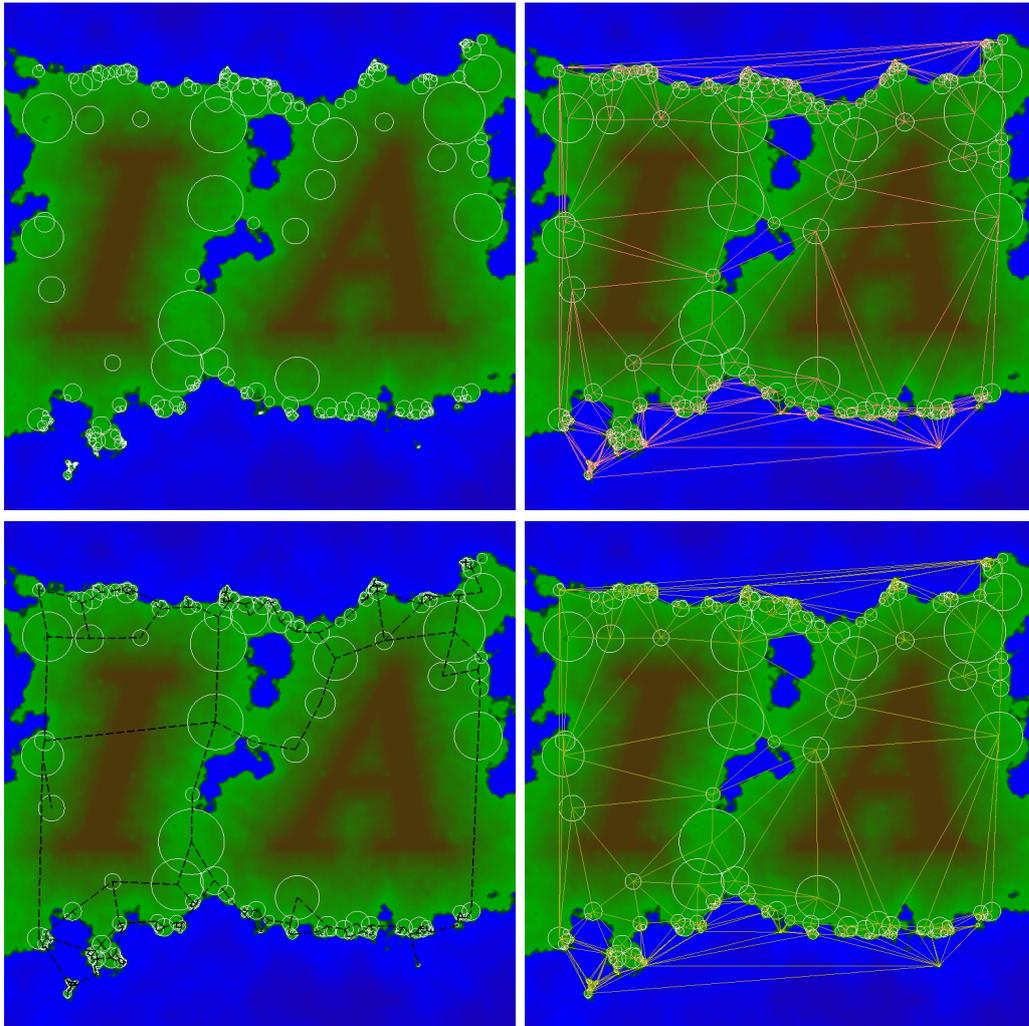


FIG. 5.17 – Urbanisation du Mont IA, sont illustrés, d'en haut à gauche et dans le sens trigonométrique inverse : les emplacements des sites d'agglomération ; le réseau hiérarchisé ; le réseau planaire ; le réseau simplifié.

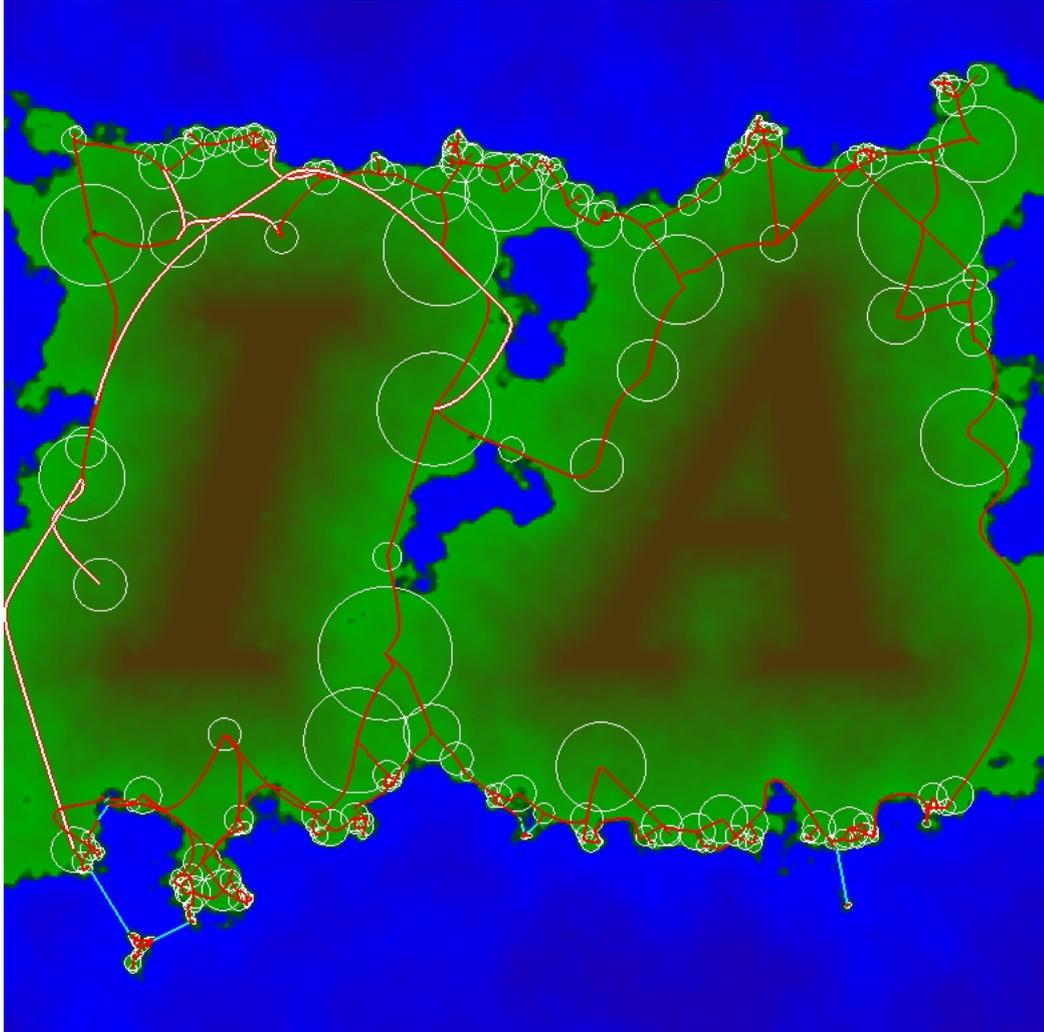


FIG. 5.18 – *Urbanisation du Mont IA : le calcul des routes.*

## 5.4 Projets

Nous proposons, en quatre points, des perspectives à court puis à moyen terme :

- L’algorithme DMMC interpole mal à partir des modèles décrivant le relief par ses lignes de niveau ; il faudrait modifier le calcul d’extrapolation en fonction du sens de description des lignes de niveau : par exemple, en faisant tourner une ligne de niveau dans le sens trigonométrique, extrapoler *plus haut* les positions à gauche de la ligne de niveau et *plus bas* les positions se trouvant à sa droite.
- Application du générateur de routes à la planification de trajectoires de robots mobiles : remplacer, si nécessaire, les courbes de Bézier par des B-Splines.
- Étendre l’approche bas-haut à d’autres méthodes de subdivision. Qu’en est-il de la  $n$ -dimensionnalité de la méthode ?
- Implémenter l’algorithme fractal DMMC dans le *Graphic Processor Unit* afin de générer plus rapidement les terrains, reliefs et autres textures sous contraintes. Dans ce cas, il serait possible de proposer des rendus 2D non-photoréalistes de scènes tridimensionnelles.

Nos travaux ont principalement porté sur la modélisation automatique au niveau macroscopique. Nous avons montré, en section 5.1.2, que notre méthode fractale DMMC était capable de descendre à des échelles plus basses : la méthode est par exemple utilisée pour générer un effet de peinture au crépi. Nos modèles de terrain urbanisé peuvent être complétés, pour des échelles décroissantes, par des solutions générant : le quadrillage de l’agglomération [86] ainsi que ses bâtiments et leurs façades [60, 75] et aller jusqu’à simuler des effets de rouille [72], de fissures [35] ou de propagation de lichens [34]. Par ailleurs, tout le processus de génération du relief érodé peut être implanté dans le *GPU* et la méthode est largement assez rapide pour le faire en temps réel sans affecter le rendu : il n’y aura plus besoin de transférer une grande quantité de données entre l’architecture centrale (i.e. *CPU* et sa mémoire vive) et la carte graphique. Dans ce contexte, une perspective intéressante consiste à proposer des rendus très réalistes (i.e. proches de ceux obtenus par un lancer de rayon) d’environnements composés d’agencement de générateurs hiérarchisés, tous implémentés dans le *GPU*.



## Liste des algorithmes

1	Déplacement des Milieux par Fenêtrage Croissant. . . . .	39
2	Processus de génération de lignes de crête. . . . .	45
3	Calcul des bassins versants d'une ligne de crête ; la procédure correspond à l'appel de <code>fp calculer_ versants</code> effectué dans <b>Algorithme 2</b> . . . . .	49
4	Procédure de calcul de l'accélération d'une particule simulant l'écoulement d'un cours d'eau. . . . .	54
5	Procédure utilisée pour perturber la trajectoire d'une particule simulant l'écoulement d'un cours d'eau. . . . .	55
6	Processus réalisé lors du déplacement d'une particule simulant l'écoulement d'un cours d'eau. . . . .	56
7	Retour sur trace : procédure pour défaire une partie de la trajectoire décrite par une particule simulant l'écoulement d'un cours d'eau. . . . .	59
8	Validation de la trajectoire d'une particule simulant l'écoulement d'un cours d'eau. . . . .	61
9	Génération d'un réseau fluvial pour un MNT décrivant des bassins hydrographiques (processus principal de simulation d'écoulement). . . . .	62
10	Déplacement des milieux, l'approche bas-haut-bas. . . . .	74
11	L'algorithme du Déplacement des Milieux Morphologiquement Contraint — DMMC. . . . .	85
12	Une itération du calcul de l'hydrométrie d'un MNT. . . . .	94
13	Calcul d'un quota de catégories d'agglomération. . . . .	100
14	Création et positionnement automatique des sites d'agglomération. . . . .	105
15	Hierarchisation des sites d'agglomération. . . . .	109
16	Modèle stochastique utilisant des courbes de Bézier pour calculer des trajectoires entre deux positions sur un modèle numérique de terrain. . . . .	114



## Bibliographie

- [1] *Eingana: The First Live Atlas in 3D*. Focus Multimedia Ltd, février 2004.
- [2] AFRIGRAPH. « The African Graphics Association ». <http://www.afrigraph.org>, 2007.
- [3] Kenichi ARAKAWA et Eric KROTKOV. « Fractal Modeling of Natural Terrain: Analysis and Surface Reconstruction with Range Data ». *Graphical Models and Image Processing*, 58(5): pp. 413–436, septembre 1996.
- [4] Didier ARQUÈS et Olivier GRANGE. « A fast scan-line algorithm for topological filling of well-nested objects in 2.5D digital pictures ». *Theor. Comput. Sci.*, 147(1-2): pp. 211–248, 1995.
- [5] Pierre AUDIBERT. *Algorithmes et Programmation*. Notes de cours, Université Paris 8, 1997.
- [6] Pierre AUDIBERT. *Ordre et Chaos, Fractales, Pavages*. Notes de cours, Université Paris 8, 2000.
- [7] Farès BELHADJ. « Terrain modeling: a constrained fractal model ». Dans *AFRIGRAPH '07: Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pp. 197–204, New York, NY, USA, 2007. ACM.
- [8] Farès BELHADJ et Pierre AUDIBERT. « Modeling landscapes with ridges and rivers ». Dans *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 151–154, New York, NY, USA, 2005. ACM Press.
- [9] Farès BELHADJ et Pierre AUDIBERT. « Modeling landscapes with ridges and rivers: bottom up approach ». Dans *GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pp. 447–450, New York, NY, USA, 2005. ACM Press.
- [10] S. BELIN, R. BELLEFON, L. BIANCO, O. COUSINOU, Y. DUPONT, P. MARIÉ, P. MERMINOD et J. NÉGRONI. *Paysages 3D*. Les cahiers du designer – 9. Eyrolles, 1ère édition, avril 2003.

- [11] Bedrich BENES et Rafael FORSBACH. « Parallel implementation of terrain erosion applied to the surface of Mars ». Dans *AFRIGRAPH '01: Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation*, pp. 53–57, New York, NY, USA, 2001. ACM Press.
- [12] B. BENEŠ. « Hydraulic Erosion by Shallow Water Simulation ». Dans *Vriphys'07: The 4th Workshop on Virtual Reality Interaction and Physical Simulation*, 2007.
- [13] B. BENEŠ et X. ARRIAGA. « Table Mountains by Virtual Erosion ». Dans *Eurographics Workshop on Natural Phenomena*, pp. 33–39, 2005.
- [14] Bedrich BENEŠ et Rafael FORSBACH. « Layered Data Representation for Visual Simulation of Terrain Erosion ». Dans *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics*, pp. 80–86, Washington, DC, USA, 2001. IEEE Computer Society.
- [15] Fabio BETTIO, Enrico GOBBETTI, Fabio MARTON et Giovanni PINTORE. « High-quality networked terrain rendering from compressed bitstreams ». Dans *Web3D '07: Proceedings of the twelfth international conference on 3D web technology*, pp. 37–44, New York, NY, USA, 2007. ACM.
- [16] P. J. BIRCH, S. P. BROWNE, V. J. JENNINGS, A. M. DAY et D. B. ARNOLD. « Rapid procedural-modelling of architectural structures ». Dans *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pp. 187–196, New York, NY, USA, 2001. ACM.
- [17] Vincent BOYER. « Pour une palette graphique performante : accélération d'algorithmes fondamentaux ». Thèse en informatique, Université Paris 8, janvier 2001.
- [18] Claus BRENNER. « Towards Fully Automatic Generation of City Models ». Dans *Proceedings of the 19th ISPRS Conference — International Society for Photogrammetry and Remote Sensing*, volume XXXIII, pp. 85–92, 2000.
- [19] John BROSZ, Faramarz F. SAMAVATI et Mario Costa SOUSA. « Terrain synthesis by-example ». Dans *GRAPP*, pp. 122–133, 2006.
- [20] L.T. BRUTON et N.R. BARTLEY. « Simulation of fractal multidimensional images using multidimensional recursive filters ». *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 41(3): pp. 181–188, mars 1994.
- [21] BRYCE™. <http://www.daz3d.com>, 2007.
- [22] Gilles BUREL. *Introduction au traitement d'images : simulation sous Matlab*. Lavoisier, 2001.

- [23] Loren C. CARPENTER. « Computer rendering of fractal curves and surfaces ». Dans *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pp. 109, New York, NY, USA, 1980. ACM.
- [24] E. CATMULL et J. CLARK. « Recursively generated B-spline surfaces on arbitrary topological meshes ». *Computer Aided Design*, 10(6): pp. 350–355, 1978.
- [25] G.M. CHAIKIN. « An algorithm for high speed curve generation ». *Computer Vision, Graphics and Image Processing*, 3(4): pp. 346–349, 1974.
- [26] Min-Yu CHIANG, Shih-Chun TU, Jun-Yan HUANG, Wen-Kai TAI, Cheng-Duo LIU et Chin-Chen CHANG. « Terrain Synthesis: An Interactive Approach ». Dans *International Workshop on Advanced Image Technology*, 2005.
- [27] N. CHIBA, K. MURAOKA et K. FUJITA. « An Erosion Model Based on Velocity Fields for the Visual Simulation of Mountain Scenery ». *The Journal of Visualization and Computer Animation*, 9(4): pp. 185–194, 1998.
- [28] Malte CLASEN et Hans-Christian HEGE. « Terrain Rendering using Spherical Clipmaps ». Dans *IEEE VGTC Symposium on Visualization*, pp. 91–98, 2006.
- [29] Comité National Français des Sciences HYDROLOGIQUES. « Dictionnaire français d'hydrologie ». Définition de Monique Dacharry et Jean Margat, novembre 1995.
- [30] Robert L. COOK et Tony DEROSE. « Wavelet noise ». *ACM Trans. Graph.*, 24(3): pp. 803–811, 2005.
- [31] Cybercity MODELER. « CC-MODELER<sup>TM</sup>: Generation of reality-based virtual 3D city and facility models ». <http://www.cybercity.tv>, 2006.
- [32] Carsten DACHSBACHER et Marc STAMMINGER. « Rendering Procedural Terrain by Geometry Image Warping ». Dans *Rendering Techniques 2004 (Proceedings of Eurographics Symposium on Rendering)*, pp. 103–110, 2004.
- [33] Emanuele DANOVARO, Leila De FLORIANI, Paola MAGILLO, Mohamed Mostefa MESMOUDI et Enrico PUPPO. « Morphology-driven simplification and multiresolution modeling of terrains ». Dans *GIS '03: Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, pp. 63–70, New York, NY, USA, 2003. ACM Press.

- [34] Brett DESBENOIT, Eric GALIN et Samir AKKOUCHE. « Simulating and Modeling Lichen Growth ». *Computer Graphics Forum*, 23, Number 3, September 2004(3): pp. 341–350, 2004.
- [35] Brett DESBENOIT, Eric GALIN et Samir AKKOUCHE. « Modeling cracks and fractures ». *The Visual Computer*, 21(8-10): pp. 717–726, 2005.
- [36] Oliver DEUSEN, David S. EBERT, Ron FEDKIW, F. Kenton MUSGRAVE, Przemyslaw PRUSINKIEWICZ, Doug ROBLE, Jos STAM et Jerry TESSENDORF. « The elements of nature: interactive and realistic techniques ». Dans *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, pp. 32, New York, NY, USA, 2004. ACM.
- [37] Edsger Wybe DIJKSTRA. « A Note on Two Problems in Connexion with Graphs ». *Numerische Mathematik*, 1: pp. 269–271, 1959.
- [38] C. DOMINGUEZ, H. HASSAN et A. CRESPO. « Mobile robotic environment representational method based on Bézier curves ». Dans *CMMSE 2002: Proceedings of International Conference on Computational and Mathematical Methods in Science and Engineering*, pp. 75–84, Alicante, Spain, septembre 2002.
- [39] D. DOO et M. SABIN. « Analysis of the behavior of recursive division surfaces near extraordinary points ». *Computer Aided Design*, 10(6): pp. 356–360, 1978.
- [40] Mark DUCHAINEAU, Murray WOLINSKY, David E. SIGETI, Mark C. MILLER, Charles ALDRICH et Mark B. MINEEV-WEINSTEIN. « ROAMing terrain: real-time optimally adapting meshes ». Dans *VIS '97: Proceedings of the 8th conference on Visualization '97*, pp. 81–88, Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.
- [41] David S. EBERT, F. Kenton MUSGRAVE, Darwyn PEACHEY, Ken PERLIN et Steven WORLEY. « *Texturing & Modeling: A Procedural Approach* », Chapitre 20 – MOJOWORLD: Building Procedural Planets, pp. 565–615. Morgan Kaufmann, 2003.
- [42] James D. FOLEY, Andries van DOM, Steven K. FEINER et John F. HUGHES. *Computer Graphics: principles and practice*. Addison Wesley, second édition, 1996.
- [43] Alain FOURNIER, Don FUSSELL et Loren CARPENTER. « Computer rendering of stochastic models ». *Commun. ACM*, 25(6): pp. 371–384, 1982.
- [44] Murray GELL-MANN. « *Le Quark et le Jaguar* », Chapitre La puissance de la théorie. Flammarion, 1997.
- [45] Kevin R. GLASS, Chantelle MORKEL et Shaun D. BANGAY. « Duplicating road patterns in south african informal settlements using procedu-

- ral techniques ». Dans *Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pp. 161–169, New York, NY, USA, 2006. ACM Press.
- [46] Michael B. GOUSIE et Wm. Randolph FRANKLIN. « Constructing a dem from grid-based data by computing intermediate contours ». Dans *GIS '03: Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, pp. 71–77, New York, NY, USA, 2003. ACM.
- [47] Stefan GREUTER, Jeremy PARKER, Nigel STEWART et Geoff LEACH. « Real-time Procedural Generation of 'Pseudo Infinite' Cities ». Dans *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pp. 87–FF. ACM Press, 2003.
- [48] R. L. GRIMSDALE et S. M. LAMBOURN. « Generation of Virtual Reality Environments using Expert Systems ». Dans *Proceedings of the 5th International Conference in Central Europe on Computer Graphics and Visualization (WSCG '97)*, pp. 153–162, 1997.
- [49] A. GRUEN, Z. LI et X. WANG. « Generation of 3D City Models with Linear Array CCD-Sensors ». Dans *Proceedings of the 6th Conference on Optical 3-D Measurement Techniques*, pp. 21–31, 2003.
- [50] A. GRUEN et X. WANG. « News from CyberCity-Modeler ». Dans *Proceedings of the 3th International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pp. 93–101, 2001.
- [51] Armin GRUEN et Xinhua WANG. « CC-Modeler: a topology generator for 3-D city models ». *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5): pp. 286–295, octobre 1998.
- [52] Eric HANSON. « Urban modeling in visual effects ». Dans *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pp. 84–136, New York, NY, USA, 2006. ACM Press.
- [53] Olof HENRICSSON, André STREILEIN et Armin GRUEN. « Automated 3-D Reconstruction of Buildings and Visualisation of City Models ». Rapport Technique, Institute of Geodesy and Photogrammetry, Federal Institute of Technology, Zürich, Switzerland, 1995.
- [54] Hugues HOPPE. « Progressive meshes ». Dans *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 99–108, New York, NY, USA, 1996. ACM.
- [55] INSEE. « Recensement de la population française — mars 1999 ». <http://www.recensement.insee.fr>, 2007.

- [56] Michael KASS et Gavin MILLER. « Rapid, stable fluid dynamics for computer graphics ». Dans *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 49–57, New York, NY, USA, 1990. ACM.
- [57] Alex D. KELLEY, Michael C. MALIN et Gregory M. NIELSON. « Terrain simulation using a model of stream erosion ». Dans *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp. 263–268, New York, NY, USA, 1988. ACM Press.
- [58] Eng-Kiat KOH et D. D. HEARN. « Fast Generation and Surface Structuring Methods for Terrain and Other Natural Phenomena ». *Computer Graphics Forum*, 11(3): pp. 169–180, 1992.
- [59] Der-Tsai LEE et A. K. LIN. « Generalized delaunay triangulation for planar graphs ». *Discrete and Computational Geometry*, 1(1): pp. 201–217, décembre 1986.
- [60] Justin LEGAKIS, Julie DORSEY et Steven GORTLER. « Feature-based cellular texturing for architectural models ». Dans *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 309–316, New York, NY, USA, 2001. ACM.
- [61] J. P. LEWIS. « Generalized stochastic subdivision ». *ACM Trans. Graph.*, 6(3): pp. 167–190, 1987.
- [62] Tsai-Yen LI, Pei-Feng CHEN et Pei-Zhi HUANG. « Motion planning for humanoid walking in a layered environment ». Dans *ICRA '03: Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pp. 3421–3427, septembre 2003.
- [63] Peter LINDSTROM et Valerio PASCUCCI. « Visualization of large terrains made easy ». Dans *VIS '01: Proceedings of the conference on Visualization '01*, pp. 363–371, Washington, DC, USA, 2001. IEEE Computer Society.
- [64] Charles LOOP et T. D. DEROSE. « Generalized B-spline surfaces of arbitrary topology ». Dans *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 347–356, New York, NY, USA, 1990. ACM.
- [65] Frank LOSASSO et Hugues HOPPE. « Geometry clipmaps: terrain rendering using nested regular grids ». *ACM Trans. Graph.*, 23(3): pp. 769–776, 2004.
- [66] Benoît MANDELBROT. *Les Objets Fractals*. Flammarion, quatrième édition, 1975-1995.
- [67] Benoît MANDELBROT. Colliers aléatoires et une alternative aux promenades au hasard sans boucle: les cordonnets discrets et fractals. Dans

*Les Comptes Rendus de l'Académie des Sciences à Paris*, volume 286 de A, pp. 933–936. mai 1978.

- [68] Benoît MANDELBROT. *The fractal geometry of nature*. W.H. Freeman and Co., New York, 1983.
- [69] Benoît MANDELBROT et J.W. Van NESS. « Fractional Brownian motions, fractional noises and applications ». Dans *SIAM Review*, volume 10, pp. 422–437, 1968.
- [70] JPL Robotics: The MARSYARD. <http://www-robotics.jpl.nasa.gov>, 2007.
- [71] Xing MEI, Philippe DECAUDIN et Baogang HU. « Fast Hydraulic Erosion Simulation and Visualization on GPU ». Dans *PG'07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pp. 47–56, 2007.
- [72] Stéphane MÉRILLOU, Jean-Michel DISCHLER et Djamchid GHAZANFARPOUR. « Corrosion : simulating and rendering. ». Dans *Graphics Interface*, pp. 167–174, Ottawa (Canada), 2001.
- [73] Gavin MILLER. « The definition and rendering of terrain maps ». Dans *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 39–48, New York, NY, USA, 1986. ACM Press.
- [74] Pandromeda MOJOWORLD. <http://www.pandromeda.com>, 2007.
- [75] Pascal MÜLLER, Peter WONKA, Simon HAEGLER, Andreas ULMER et Luc Van GOOL. « Procedural modeling of buildings ». Dans *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pp. 614–623, New York, NY, USA, 2006. ACM Press.
- [76] Pascal MÜLLER, Gang ZENG, Peter WONKA et Luc Van GOOL. « Image-based procedural modeling of facades ». Dans *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, pp. 85, New York, NY, USA, 2007. ACM Press.
- [77] F. Kenton MUSGRAVE, Craig E. KOLB et Robert S. MACE. « The Synthesis and Rendering of Eroded Fractal Terrains ». *Computer Graphics*, 23(3): pp. 41–50, 1989.
- [78] Kenji NAGASHIMA. « Computer generation of eroded valley and mountain terrains ». *The Visual Computer*, 13(9–10): pp. 456–464, 1998.
- [79] Ihn NAMGUNG. « Application of Quadratic Algebraic Curve for 2D Collision-Free Path Planning and Path Space Construction ». *International Journal of Control Automation and Systems*, 2(1): pp. 107–117, 2004.

- [80] Benjamin NEIDHOLD, Oliver DEUSSEN et Markus WACKER. « Interactive physically based Fluid and Erosion Simulation ». *Eurographics Workshop on Natural Phenomena*, pp. 25–32, 2005.
- [81] NGDC. « National Geophysical Data Center ». <http://ngdc.noaa.gov>, 2007.
- [82] Clark F. OLSON, Larry H. MATTHIES, John R. WRIGHT, Rongxing LI et Kaichang DI. « Visual terrain mapping for Mars exploration ». *Comput. Vis. Image Underst.*, 105(1): pp. 73–85, 2007.
- [83] Teong Joo ONG, Ryan SAUNDERS, John KEYSER et John J. LEGGETT. « Terrain generation using genetic algorithms ». Dans *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 1463–1470, New York, NY, USA, 2005. ACM Press.
- [84] Dinesh K. PAI et L.-M. REISSELL. « Multiresolution rough terrain motion planning ». Dans *Proceedings of IEEE Transactions on Robotics and Automation*, volume 14, pp. 19–33, février 1998.
- [85] Renato PAJAROLA et Enrico GOBBETTI. « Survey of semi-regular multiresolution models for interactive terrain rendering ». *Vis. Comput.*, 23(8): pp. 583–605, 2007.
- [86] Yoav I. H. PARISH et Pascal MÜLLER. « Procedural modeling of cities ». Dans *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 301–308. ACM Press, 2001.
- [87] Sam H. PARRY. « The generation and use of parameterized terrain in land combat simulation ». Dans *WSC '77: Proceedings of the 9th conference on Winter simulation*, pp. 422–431. Winter Simulation Conference, 1977.
- [88] Heinz-Otto PEITGEN et P.H. RICHTER. *The Beauty of Fractals*. Springer Verlag, Berlin, Allemagne, 1986.
- [89] Heinz-Otto PEITGEN et Dietmar SAUPE, éditeurs. *The Science of Fractal Images*. Springer-Verlag New York, Inc., New York, NY, USA, 1988.
- [90] Ken PERLIN. « An image synthesizer ». Dans *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pp. 287–296, New York, NY, USA, 1985. ACM Press.
- [91] Ken PERLIN. « Improving noise ». Dans *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 681–682, New York, NY, USA, 2002. ACM.
- [92] Ken PERLIN et Luiz VELHO. « Live paint: painting with procedural multiscale textures ». Dans *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 153–160, New York, NY, USA, 1995. ACM.

- [93] J. PETTRÉ, T. SIMÉON et J.P. LAUMOND. « Planning human walk in virtual environments ». Dans *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pp. 3048–3053, 2002.
- [94] Jacques PEYRIÈRE. Processus de naissance avec interaction des voisins, évolution des graphes. Dans *Annales de l'institut Fourier*, volume 31, pp. 187–218. 1981.
- [95] Joachim POUDEROUX, Jean-Christophe GONZATO, Ireneusz TOBOR et Pascal GUITTON. « Adaptive hierarchical RBF interpolation for creating smooth digital elevation models ». Dans *GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pp. 232–240, New York, NY, USA, 2004. ACM Press.
- [96] Joachim POUDEROUX et Jean-Eudes MARVIE. « Adaptive streaming and rendering of large terrains using strip masks ». Dans *GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pp. 299–306, New York, NY, USA, 2005. ACM.
- [97] POVRAY. « The Persistence of Vision Raytracer ». <http://www.povray.org>, 2007.
- [98] Przemyslaw PRUSINKIEWICZ et Mark HAMMEL. « A Fractal Model of Mountains with Rivers ». Dans *Proceeding of Graphics Interface '93*, pp. 174–180, mai 1993.
- [99] Przemyslaw PRUSINKIEWICZ et Aristid LINDENMAYER. *Algorithmic Beauty of Plants*. Springer Verlag, 1990.
- [100] Randi J. ROST. *OpenGL(R) Shading Language (2nd Edition)*. Addison-Wesley Professional, 2005.
- [101] P. ROUDIER, B. PEROCHE et M. PERRIN. « Landscapes Synthesis Achieved through Erosion and Deposition Process Simulation ». *Computer Graphics Forum*, 12(3): pp. 375–383, 1993.
- [102] Robert SEDGEWICK. « *Algorithms in C* », Chapitre FIFO queues and generalized queues, pp. 153–161. Addison-Wesley, 1998.
- [103] Robert SEDGEWICK. *Algorithms in C, Part 5 — Graph Algorithms*. Addison-Wesley, 2002.
- [104] Zvi SHILLER. « Motion Planning for Mars Rover ». Dans *Proceedings of the 1st Workshop on Robot Motion and Control (RoMoCo)*, pp. 257–262, 1999.
- [105] Zvi SHILLER. « Obstacle traversal for space exploration ». Dans *Proceedings of ICRA '00, IEEE International Conference on Robotics and Automation*, volume 2, pp. 989–994, San Francisco, CA, USA, mai 2000.

- [106] Zvi SHILLER et J.C. CHEN. « Optimal motion planning of autonomous vehicles in three dimensional terrains ». Dans *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pp. 198–203, Cincinnati, OH, USA, mai 1990.
- [107] Zvi SHILLER et Steven DUBOWSKY. « On computing the global time-optimal motions of robotic manipulators in the presence of obstacles ». Dans *IEEE Transactions on Robotics and Automation*, volume 7, pp. 785–797, décembre 1991.
- [108] Zvi SHILLER et Yu-Rwei GWO. « Dynamic motion planning of autonomous vehicles ». Dans *IEEE Transactions on Robotics and Automation*, volume 7, pp. 241–249, avril 1991.
- [109] Dave SHREINER, Mason WOO, Jackie NEIDER et Tom DAVIS. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2*. Addison-Wesley, 2005.
- [110] T. SIMEON et B. DACRE-WRIGHT. « A practical motion planner for all-terrain mobile robots ». Dans *Proceedings of IROS '93, IEEE/RSJ International Conference on Intelligent Robots and Systems '93.*, volume 2, pp. 1357–1363, Yokohama, Japan, juillet 1993.
- [111] Guang SONG et Nancy M. AMATO. « Randomized Motion Planning for Car-like Robots with C-PRM ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pp. 37–42, 2001.
- [112] SRTM. « Shuttle Radar Topography Mission ». <http://srtm.usgs.gov>, 2005.
- [113] S. STACHNIAK et W. STUERZLINGER. « An Algorithm for Automated Fractal Terrain Deformation ». Dans *Proceedings of the 7th international conference on Computer Graphics and Artificial Intelligence*, pp. 64–76. Ed. D. Plemenos, mai 2005.
- [114] Jos STAM. « Stable fluids ». Dans *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [115] Roger T. STEVENS. *Advanced Fractal Programming in C*. M&T, 1990.
- [116] Jing SUN, Xiaobo YU, George BACIU et Mark GREEN. « Template-based generation of road networks for virtual city modeling ». Dans *VRST '02: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 33–40, New York, NY, USA, 2002. ACM Press.
- [117] Sayaka SUZUKI et Hirofumi CHIKATSU. « Recreating the Past City Model of Historical Town Kawagoe from Antique Map ». Dans *In-*

- ternational Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIV-5/W10, 2002.
- [118] R. SZELISKI et D. TERZOPOULOS. « From splines to fractals ». Dans *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pp. 51–60, New York, NY, USA, 1989. ACM.
- [119] Richard SZELISKI. « Regularization Uses Fractal Priors ». Dans *AAAI*, pp. 749–754, 1987.
- [120] Y. TAKASE, N. SHO, A. SONE et K. SHIMIYA. « Automatic Generation of 3D City Models and Related Applications ». Dans *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIV-5/W10, 2002.
- [121] David G. TARBOTON. « Terrain Analysis Using Digital Elevation Models in Hydrology ». Dans *Proceedings of the 23rd ESRI International Users Conference*, 2003.
- [122] David G. TARBOTON et Daniel P. AMES. « Advances in the Mapping of Flow Networks from Digital Elevation Data ». Dans Gerald Sehlke DON PHELPS, éditeur, *Proceedings of World Water and Environmental Resources Congress*, mai 2001.
- [123] Hind TAUD et Jean-François PARROT. « Measurement of DEM roughness using the local fractal dimension ». *Géomorphologie: relief, processus, environnement*, (4): pp. 327–338, avril 2005.
- [124] TERRAGEN<sup>TM</sup>. <http://www.planetside.co.uk>, 2007.
- [125] Kelvin THOMPSON. Scanline Depth Gradient of a Z-Buffered Triangle. Dans Graphics GEMS, éditeur, *Andrew S. Glassner*, volume 1, pp. 361–363. Academic Press, Inc., 1990.
- [126] G. TUCCI, F. ALGOSTINO, V. BONORA et G. GUIDI. « 3D Modeling of Boccaccio's Hometown Through a Multisensor Survey ». Dans *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIV-5/W10, 2002.
- [127] Kilian ULM. « Improved 3D City Modeling with CyberCity-Modeler using Aerial-, Satellite Imagery and LaserScanner Data ». Dans *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIV-5/W10, 2002.
- [128] Kilian ULM. « Reality-Based 3D City Models With CyberCity-Modeler and Laserscanner Data ». Dans *Proceedings of the 6th Conference on Optical 3-D Measurement Techniques*, 2003.
- [129] USGS-ARP. « The USGS Astrogeology Research Program ». <http://astrogeology.usgs.gov>, 2007.

- [130] B. C. VEMURI, C. MANDAL et S.-H. LAI. « A Fast Gibbs Sampler for Synthesizing Constrained Fractals ». *IEEE Transactions on Visualization and Computer Graphics*, 3(4): pp. 337–351, 1997.
- [131] Richard F. VOSS. *Random Fractal Forgeries*. Springer-Verlag, Berlin, Allemagne, 1985.
- [132] Benjamin WATSON. « Modeling land use with urban simulation ». Dans *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pp. 185–251, New York, NY, USA, 2006. ACM Press.
- [133] William D. WELLS. « Generating enhanced natural environments and terrain for interactive combat simulations (GENETICS) ». Dans *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 184–191, New York, NY, USA, 2005. ACM.
- [134] John P. WILSON et John C. GALLANT. *Terrain Analysis Principles and Applications*. John Wiley and Sons, 2000.
- [135] Peter WONKA. « Procedural modeling of architecture ». Dans *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pp. 17–83, New York, NY, USA, 2006. ACM Press.
- [136] Peter WONKA. « State of the art in modeling and rendering ». Dans *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, pp. 11–111, New York, NY, USA, 2007. ACM Press.
- [137] Peter WONKA, Michael WIMMER, François SILLION et William RIBARSKY. « Instant Architecture ». *ACM Transactions on Graphics*, 22(4): pp. 669–677, juillet 2003. Proceedings of SIGGRAPH 2003.
- [138] L. YANG et S. LAVALLE. « An Improved Random Neighborhood Graph Approach ». pp. 254–259, mai 2002.
- [139] L. YANG et S. M. LAVALLE. « A Framework for Planning Feedback Motion Strategies Based on a Random Neighborhood Graph ». Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 544–549, 2000.
- [140] C. K. YAP, H. BIERMANN, A. HERTZMAN, C. LI, J. MEYER, H. K. PAO et T. PAXIA. « A Different Manhattan Project: Automatic Statistical Model Generation ». Rapport Technique, Courant Institute of Mathematical Sciences, New York University, 1998. <http://www.cs.nyu.edu/visual/home/proj/manhattan>.
- [141] Christopher ZACH, Andreas KLAUS, Joachim BAUER, Konrad KARNER et Markus GRABNER. « Modeling and visualizing the cultural heritage data set of Graz ». Dans *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pp. 219–226, New York, NY, USA, 2001. ACM.

- [142] Howard ZHOU, Jie SUN, Greg TURK et James M. REHG. « Terrain Synthesis from Digital Elevation Models ». *IEEE Transactions on Visualization and Computer Graphics*, 13(4): pp. 834–848, août 2007.
- [143] Michael ZYDA, Alex MAYBERRY, Casey WARDYNSKI, Russell SHILLING et Margaret DAVIS. « The MOVES institute's America's army operations game ». Dans *I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pp. 219–220, New York, NY, USA, 2003. ACM.