



**HAL**  
open science

## Modélisation de terrains virtuels

Eric Guérin

► **To cite this version:**

Eric Guérin. Modélisation de terrains virtuels. Synthèse d'image et réalité virtuelle [cs.GR]. INSA de LYON; Université Claude Bernard Lyon 1, 2017. tel-01635126

**HAL Id: tel-01635126**

**<https://hal.science/tel-01635126v1>**

Submitted on 14 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# HABILITATION À DIRIGER DES RECHERCHES

PRÉSENTÉE DEVANT  
L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON  
ET L'UNIVERSITÉ CLAUDE BERNARD LYON 1



---

## Modélisation de terrains virtuels

---

SPÉCIALITÉ : INFORMATIQUE

par **Éric GUÉRIN**

soutenue le 14 novembre 2017 devant la commission d'examen

*Rapporteurs* Loïc BARTHE, professeur, Université Paul Sabatier, Toulouse  
Bedrich BENES, professor, Purdue University  
Jean-Michel DISCHLER, professeur, Université de Strasbourg

*Jury* Loïc BARTHE, professeur, Université Paul Sabatier, Toulouse  
Atilla BASKURT, professeur, INSA de Lyon  
Raphaëlle CHAINE, professeure, Université Claude Bernard Lyon 1  
Jean-Michel DISCHLER, professeur, Université de Strasbourg  
Éric GALIN, professeur, Université Claude Bernard Lyon 1

Laboratoire d'InfoRmatique en Image et Systèmes d'information





## TABLE DES MATIÈRES

---

1	CV ET CARRIÈRE	3
2	INTRODUCTION	15
3	MODÉLISATION DE TERRAINS - ÉTAT DE L'ART	17
4	AJOUTS DE DÉTAILS MULTI-RÉSOLUTION PAR MODÈLES FRACTALS	23
5	TERRAINS DE DIFFUSION	31
6	MODÈLE DE TERRAIN FONCTIONNEL ET CONSTRUCTIVISTE	39
7	MODÈLE DE TERRAIN PARCIMONIEUX	51
8	CONCLUSIONS ET PERSPECTIVES	63
A	COMPARATIF DES PUBLICATIONS SUR LES TERRAINS	73



## REMERCIEMENTS

---

Le travail qui est relaté dans ce document a pu se faire notamment grâce à la confiance et l'aide de Samir Akkouché qui m'a confié la direction de la thèse de Houssam Hnaidi et Eric Galin qui m'a permis de co-encadrer la thèse de Nicolas Maréchal, Jean-David Génévaux, puis Antoine Webanck. Qu'ils en soient chaleureusement remerciés.

Je remercie mes doctorants passés et présents, Houssam Hnaidi, Jean-David Genevaux, Nicolas Maréchal, Antoine Webanck et Yann Cortial pour leur confiance et les bons moments passés à travailler ensemble.

Je remercie également tous mes autres co-auteurs qui m'ont accompagné sous diverses formes dans mes travaux : Adrien Peytavie, Julie Digne, Bedrich Benes, Marie-Paule Cani, Guillaume Cordonnier, James Gain, Carlos Andujar, Luiz Velho, Leandro Cruz, Djalma Lucio, Jean Braun, Atilla Baskurt et Eric Tosan.

Pour terminer, je remercie chaleureusement les membres de jury et rapporteurs de ce document pour avoir accepté cette tâche : Raphaëlle Chaine, Loïc Barthe, Atilla Baskurt, Jean-Michel Dischler, Eric Galin et Bedrich Benes.



## CV ET CARRIÈRE

---

Ce chapitre contient des éléments factuels sur ma carrière, mes responsabilités, les enseignements qui m'ont été confiés ainsi que mes mandats électifs. En dernier lieu, je fais un bilan de l'encadrement doctoral que j'ai supervisé pendant ces dix dernières années.

### 1.1 BREF CV

Je suis diplômé de l'INSA de Lyon, département informatique en 1998. J'ai par la suite effectué un DEA d'informatique, option informatique fondamentale, au sein de l'Université Claude Bernard Lyon 1. À l'issue de ce DEA, j'ai obtenu une allocation du ministère pour effectuer une thèse sous la direction d'Eric Tosan et Atilla Baskurt. Cette thèse portait sur *l'approximation fractale de courbes et de surfaces*.

J'ai obtenu mon doctorat en décembre 2002, année pendant laquelle j'ai été ATER à mi-temps à l'université Claude Bernard Lyon 1. J'ai obtenu un second poste d'ATER à l'IUT de Bourg-en-Bresse l'année suivante. En 2004, je candidate à un poste de maître de conférences dans cet IUT, pour lequel je suis classé premier. En 2010, je candidate à un poste de maître de conférences à l'INSA de Lyon, département informatique, que j'obtiens et auquel je suis actuellement rattaché.

Ce manuscrit ne porte que sur la période après la soutenance de ma thèse en 2002, la période antérieure étant déjà traitée dans le manuscrit de ma thèse.

### 1.2 RESPONSABILITÉS LIÉES À L'ENSEIGNEMENT

Les charges administratives les plus lourdes que j'ai eues à assumer sont celles qui m'ont été confiées à l'IUT de Bourg-en-Bresse.

**RESPONSABLE PÉDAGOGIQUE DUT** De 2005 à 2010, j'ai assuré la responsabilité pédagogique du DUT au département informatique de l'IUT de Bourg-en-Bresse. Il s'agit là d'une charge lourde qui porte sur l'organisation d'environ 5000 heures d'enseignement par année. Elle exige une connaissance parfaite des contenus pédagogiques (programme pédagogique national du DUT) ainsi que des intervenants, permanents en poste mais aussi vacataires.

**MONTAGE ET DIRECTION D'UNE LICENCE PROFESSIONNELLE** J'ai participé très activement au montage de la licence professionnelle Informatique Embarquée et Mobile, dont j'ai assumé la direction la première année de sa création, en 2007. Cette formation a été précurseur dans la montée en puissance des technologies mobiles que nous connaissons actuellement.



**IMPLICATION PÉDAGOGIQUE FORTE** Cette implication passe par la responsabilité des projets tutorés, ainsi que l'organisation de nouvelles approches pédagogiques comme les semaines spéciales dont le but est de confronter les étudiants à un sujet unique dans un cadre de travail en équipe de 4, proche des conditions de travail réelles.

À l'INSA, j'ai participé à la création d'une Option Transversale en cinquième année sur la valorisation et l'innovation dans le domaine du logiciel. Depuis 2015, je suis le responsable des études de la troisième année au département informatique de l'INSA. J'ai organisé la refonte de la maquette pédagogique en troisième année que j'ai ensuite mise en œuvre par la création de l'emploi du temps.

**DIVERS** Au niveau de l'antenne de Bourg-en-Bresse (plus de 300 étudiants), j'ai géré l'organisation des sports, ainsi que des épreuves du C2i pendant plusieurs années.

### 1.3 ENSEIGNEMENTS

À l'IUT de Bourg-en-Bresse, j'ai effectué mes enseignements majoritairement dans le domaine de la programmation (C, C++) et de la programmation Web (HTTP, HTML, CSS, JavaScript, PHP).

Depuis mon arrivée à l'INSA, j'enseigne l'algorithmique, la programmation du C++, la théorie des langages et grammaires ainsi que les bases de la synthèse d'image.

Pour tous ces modules d'enseignement, j'ai assuré le montage des cours et de leur support photocopié ainsi que la préparation et l'encadrement des TD, TP et projets.

### 1.4 MANDATS ÉLECTIFS

J'ai été de 2009 à 2015 membre élu au conseil de laboratoire du LIRIS. Cette fonction m'a permis de participer activement à la vie du laboratoire ainsi qu'à prendre part à quelques décisions (budget, allocations doctorales).

Depuis 2010, je fais partie du vivier d'experts en section 27 au sein de l'INSA de Lyon. A ce titre, je participe aux comités de recrutements d'ATER ainsi qu'à de nombreux comités de sélection.

### 1.5 ENCADREMENT DOCTORAL

À la date de l'écriture de ce manuscrit, j'ai co-encadré trois thèses soutenues et je co-encadre actuellement une thèse débutée en octobre 2015.

Voici la liste des co-encadrements de thèses en cours ou soutenues :

- Houssam Hnaidi (2006-2010) : contrôle dans la génération de formes naturelles (co-encadrement avec Samir Akkouche)
- Nicolas Maréchal (2007-2010) : génération de contenu graphique (co-encadrement avec Eric Galin)
- Jean-David Génévaux (2011-2014) : modélisation de mondes virtuels (co-encadrement avec Eric Galin)
- Antoine Webanck (2015-2018) : modélisation d'effets atmosphériques (co-encadrement avec Eric Galin)

J'ai participé également à l'encadrement de nombreux stages de master recherche :

- Gilles Gouaty (2004) : visualisation rapide formes fractales
- Cédric Dussart (2008) : génération automatique d'intérieurs de bâtiments
- Guillaume Hayette (2009) : génération procédurale de routes
- Anca Tirdea (2010) : génération procédurale de terrains
- Mathieu Biston (2011) : génération procédurale et visualisation
- Ulysse Vimont (2013) : génération procédurale de mondes virtuels
- Jérémy Gaillard (2014) : génération de terrains cohérents d'un point de vue hydrologique
- Logan Salmon (2016) : génération de terrains par une méthode parcimonieuse
- Yann Cortial (2017) : modélisation et rendu de paysages nuageux

J'ai eu aussi l'occasion d'encadrer de nombreux stages de recherche d'un ou deux mois dans l'équipe (L3 à Lyon 1, ENS et INSA) ainsi que des travaux encadrés de recherche de M1.

## 1.6 ANIMATION SCIENTIFIQUE

**CHEF D'ÉQUIPE RECHERCHE** Depuis 2015, je dirige l'équipe de recherche Geomod. Cette tâche se traduit par de l'animation scientifique, de la rédaction de bilans intermédiaires et un rôles de représentation.

### ORGANISATION DE CONFÉRENCES



J'ai organisé la conférence Vriphys 2011, participé à l'organisation de SMI 2007, HPG 2014, EGSR 2014, AFIG 2015 et j'ai fait partie du comité d'organisation de la conférence Eurographics 2017 qui a eu lieu à Lyon en avril 2017 dont j'ai été responsable des sessions de *Fast Forward* et photographe officiel. Cette dernière a été un franc succès puisqu'elle a accueilli plus de 400 participants

avec un budget d'environ 200 k€.

**INVESTISSEMENT DANS LA COMMUNAUTÉ** Je suis depuis 2013 secrétaire de l'association Chapitre Français d'Eurographics. Dans ce cadre, je rédige des comptes-rendus et organise les élections du conseil d'administration et du bureau.

**RELECTURE D'ARTICLES** J'ai participé à la relecture de nombreux articles dans des revues et conférences : Eurographics, Siggraph, Computer and Graphics, Computer Graphics Forum, JCISE, WSCG, SINUM, Fractals et CGI. J'ai fait partie des comités de programme des conférences Vriphys, SMI, et *short papers* d'Eurographics.

**COOPÉRATIONS INTERNATIONALES** J'ai contribué à établir des relations internationales fortes avec des universités étrangères, en particulier avec Bedrich Benes de l'Université de Purdue aux Etats-Unis. Cette collaboration a donné lieu à plusieurs publications. Il existe d'autres collaboration avec le Brésil (Instituto Nacional de Matemática Pura e Aplicada, Rio) ; l'Espagne (Universitat Politècnica de Catalunya, Barcelone), l'Afrique du Sud (Université du Cap) ou le Japon (Université d'Hokkaido).

**POLITIQUE DE PUBLICATION** J'ai participé à la mise en place d'une politique de recherche et de diffusion scientifique visant à améliorer la qualité de publication au sein de l'équipe de recherche. Cette politique prime la qualité plutôt que la quantité. Les conférences visées prioritairement sont Siggraph, Eurographics, Siggraph ASIA et Pacific Graphics. Certains travaux qui n'auraient pas pu être publiés dans ce lot de conférences sont toutefois soumis à des revues ou des conférences avec un rayonnement moindre comme Computer Graphics International ou Graphics Interface. Sur un plan personnel, toutes mes publications internationales sont de rang A+, A ou B, ce qui me place bien au dessus de la moyenne du laboratoire.

**COMITÉS DE SÉLECTION** J'ai participé à plusieurs comités de sélection :

- 2 postes de maîtres de conférences à Dijon en 2009 (membre extérieur)
- Une chaire mixte CNRS/INSA en 2011
- 2 postes de maître de conférence à l'INSA en 2012

**MÉDIATION SCIENTIFIQUE** J'ai participé à l'encadrement de plusieurs TPE (travaux personnels encadrés) de première, sur des sujets autour des fractales.

## 1.7 PARTICIPATION À DES PROJETS DE RECHERCHE

### 1.7.1 *Art3D et Eros3D - 2003-2009*



Ces deux projets qui ont été faits dans la continuité avaient pour objet principal la conservation du patrimoine historique et culturel. Par l'intermédiaire de campagnes de numérisations d'œuvres d'art et en collaboration avec le C2RMF (Centre de Recherche et de Restauration des Musées de France), une base de données a ainsi pu être créée, regroupant des outils de visualisation 3D, de recherche, et d'archivage.

### 1.7.2 *Architecture fractale - 2005-2009*



Ce projet avait pour but d'étudier les possibilités de la modélisation géométrique fractale dans le domaine de l'architecture. Il s'est fait en collaboration avec le laboratoire IBois de l'EPFL et a donné lieu au financement de la thèse de Gilles Gouaty.

### 1.7.3 *GENAC II - 2007-2010 - financement 200k€*



Ce projet s'est fait en collaboration avec les studios de jeux vidéo Wide Screen Games et Eden Games. Dans le cadre du pôle de compétitivité Imaginove, il avait pour but de fournir aux studios de jeux vidéo de nouveaux outils pour la génération de contenu graphique. Il a donné lieu au financement de la thèse de Nicolas Maréchal. Les échanges avec ces deux entreprises ont été fructueux et nous avons pu cerner de manière précise les besoins qu'elles avaient.

#### 1.7.4 DNA (Details in Nature) - 2007-2010 - financement 150k€



Sans faire partie officiellement du projet DNA, j'y ai participé par l'intermédiaire d'une collaboration avec Houssam Hnaidi. Ce projet ANR ARA avait pour but la modélisation de détails dans les scènes naturelles.

#### 1.7.5 IMU Bât3D - 2012-2015



Le Labex IMU (Intelligence des Mondes Urbains) regroupe des chercheurs en SHS et en Sciences de l'environnement. Les recherches sont centrées sur la ville, l'urbain, la métropolisation et l'urbanisation. Des contacts ont été pris avec Valérie Matoïan de la MOM (Maison de l'Orient et de la Méditerranée) pour une collaboration dans le cadre archéologique des fouilles sur le site d'Ougarit en Syrie. Ces contacts ont donné lieu à un projet nommé Bât3D dont le but est d'investir les

#### 1.7.6 Mango - 2012-2014 - financement 150k€



L'objectif stratégique du projet était d'accélérer le transfert de technologie entre les laboratoires de recherche et l'industrie pour faire face à l'arrivée des consoles NextGen. Nous avons développé de nouvelles méthodes de génération procédurale de mondes virtuels qui ont pu être intégrées dans la production de jeux vidéos. D'autres travaux ont fait l'objet de transferts de technologie, comme ce fut le cas de deux articles sur la génération de routes et réseaux routiers [GPMG10, GPGB11].

#### 1.7.7 Papaya - 2015-2018 - financement 350k€



Dans la continuité du projet Mango, nous avons initié un nouveau projet dont l'ambition était encore plus grande. Nous devons dans le cadre de ce projet faire face à des étendues de modélisation encore plus grandes, de l'ordre d'une planète entière. Ce projet a plusieurs volets : l'atmosphère, les rivières, les terrains et fait l'objet du financement de deux thèses au sein de l'équipe Geomod.

#### 1.7.8 BQR INSA contrôle non destructif - 2016



Dans le cadre des projets BQR à l'INSA et en collaboration avec le Laboratoire Vibrations Acoustique (LVA) de l'INSA de Lyon, nous avons obtenu le financement d'un projet sur la production d'un logiciel de simulation des rayons X par une méthode semblable au lancer de rayons. Ce projet a donné lieu à deux stages de niveau L3 en 2017.

### 1.7.9 HDWorlds - 2016-2020 - financement 300k€



Dans le but de produire des mondes virtuels de taille gigantesque, le recours à la génération procédurale semble inévitable. Dans ce contexte, nous souhaitons mettre en œuvre des modèles et des processus multi-échelles qui soient capables de générer la géométrie et la texture de manière couplée, intelligente et contrôlée.

## 1.8 MONTAGE DE PROJETS

### 1.8.1 *Mango et Papaya*

J'ai participé activement au montage de ces deux projets même si je n'en suis pas le porteur. Cela concerne le découpage scientifique, le chiffrage du budget et la coordination avec les partenaires académiques et industriels.

### 1.8.2 *Projet ANR 2011 RS2S*

Les derniers mois de l'année 2010 m'ont très largement occupé dans le montage d'un projet ANR pour l'appel à projet Modèles Numériques 2011. Ce projet dont j'ai été le porteur a pour nom RS2S (Realistic Simulation and Rendering of Seasoned Sceneries). L'idée est de simuler l'impact de la météorologie sur une scène naturelle afin d'agrémenter celle-ci d'éléments de détails de manière automatique dans un processus de génération procédurale. Parmi ces détails, on retrouve l'effet de la pluie, de la neige, du givre, les effets atmosphériques, mais aussi tout les paramètres biologiques (apparence de la végétation, feuilles mortes, etc.). La direction Marketing et Commerciale de Météo France a été fortement intéressée par ce projet et a fait partie des partenaires. Dans le cadre du projet, la collaboration avec deux équipes du CNRM (Centre National de la Recherche Météorologique) était prévue. Ce projet n'a finalement pas été financé par l'ANR. J'ai gardé de bons contacts avec des personnes de Météo France qui me sont maintenant utiles dans le cadre des recherches sur les effets atmosphériques.

### 1.8.3 *Projet ANR Genuine 2016 et 2017*

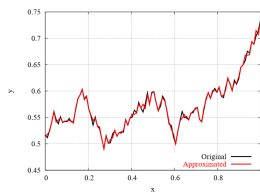
Ce projet dont je suis le porteur a été monté en 2016, finalement non financé par l'ANR, puis resoumis en 2017 où il vient d'être refusé. Un des buts est d'utiliser des méthodes de génération procédurale de scènes virtuelles pour alimenter des algorithmes de reconnaissance automatique d'éléments visuels dans une scène. La quantité de données que l'on peut générer est bien plus élevée que ce que l'on peut produire à la main et permettrait donc d'augmenter la qualité de la reconnaissance par des méthodes d'apprentissage profond. D'autres problématiques se posent, comme celle du biais introduit par la différence entre les données synthétiques et les données réelles (notion de *transfer learning*). Un autre volet concerne un dispositif assez complexe de réalité virtuelle combinée à de l'*eye-tracking* permettant de faire des simulations de scénarios de réhabilitation d'accotements de fleuves. Le dispositif complet permet de savoir où l'utilisateur regarde et l'analyse automatique permet de savoir le type d'objet qui retient son attention pour une analyse perceptuelle.

## 1.9 PUBLICATIONS

Dans cette section, je fais un tour d’horizon de mes publications principales en essayant de donner un aperçu de l’étendue des domaines couverts. Certaines de ces publications seront plus détaillées dans les chapitres suivants.

1.9.1 *Modèles fractals*

**FRACTAL INVERSE PROBLEM : APPROXIMATION FORMULATION AND DIFFERENTIAL METHODS.** Eric Guérin & Eric Tosan. *Fractal in Engineering : new trends in theory and applications*, chapitre Image Processing, pages 271-285. Springer 2005. [GT05]



L’utilisation de méthodes différentielles dans le problème fractal inverse est originale car on ne s’attend pas à ce qu’un modèle fractal possède des propriétés différentielles qui permettent de le faire. Cet article reprend une partie des résultats de différentiabilité ainsi que des applications de cette propriété. Il sera développé dans le chapitre 4.

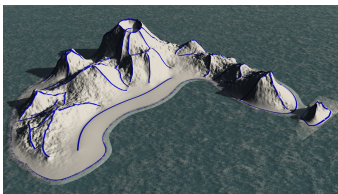
**MULTIRESOLUTION CONTROL OF CURVES AND SURFACES WITH A SELF-SIMILAR MODEL.** Houssam Hnaidi, Eric Guérin & Samir Akkouche. *Fractals*, 18(3), 271-286, 2010. [HGA10a]



Nous présentons dans cet article un nouveau modèle issu d’une hybridation entre les fractales et la notion de détails proche des ondelettes. Ce nouveau modèle permet de combiner l’aspect rugueux des fractales avec la facilité d’utilisation d’une forme contrôlée par des points. Ce travail sera développé dans le chapitre 4.

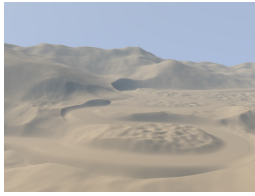
1.9.2 *Modélisation de terrains*

**FEATURE BASED TERRAIN GENERATION USING DIFFUSION EQUATION.** Houssam Hnaidi, Eric Guérin, Samir Akkouche, Adirne Peytavie & Eric Galin. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 29(7), 2179-2186, 2010. [HGA<sup>+</sup>10b]



Ce travail a pour but de reconstruire automatiquement un terrain à partir d’une description vectorielle de celui-ci mettant en avant ses caractéristiques principales (rivières, crêtes, etc.). La méthode de reconstruction repose sur l’utilisation de l’équation aux dérivées partielles de diffusion de la chaleur. Ce travail sera développé dans le chapitre 5.

**TERRAIN MODELING FROM FEATURE PRIMITIVES.** Jean-David Génevaux, Eric Galin, Adrien Peytavie, Eric Guérin, Cyril Briquet, François Grosbellet & Bedrich Benes. *Computer Graphics Forum* 34(6), 2015. pp. 198-210. [GGP<sup>+</sup>15]



Il s'agit là de représenter un terrain grâce à une combinaison de fonctions par des opérateurs de type mélange, ajout, remplacement, etc. Ce type de représentation est très adapté à la modélisation de terrains très grands avec un niveau de détail et de contrôle élevé. Des propriétés mathématiques garantissent la continuité et ainsi un affichage par lancer de rayons accéléré. Ce travail sera développé dans

le chapitre 6.

SPARSE REPRESENTATION OF TERRAINS FOR PROCEDURAL MODELING. Eric Guérin, Julie Digne, Eric Galin & Adrien Peytavie. *Computer Graphics Forum* 35(2), 2016. Proceedings of Eurographics 2016, Lisbonne (Portugal). [GDGP16]



La modélisation parcimonieuse consiste à représenter un signal de manière optimale en construisant un dictionnaire adapté à ce signal. Nous avons transposé ce formalisme à la représentation des terrains. Ce travail a énormément d'applications avec un formalisme commun : synthèse de terrains, amplification (au sens super-résolution) de terrains, modélisation

procédurale inverse (trouver une fonction qui décrit un terrain réel), ainsi que représentation compacte. Ce travail sera développé dans le chapitre 7.

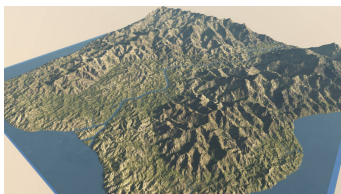
### 1.9.3 Synthèse de terrains

TERRAIN GENERATION USING PROCEDURAL MODELS BASED ON HYDROLOGY. J.D. Genevoux, E. Galin, E. Guérin, A. Peytavie & B. Benes. *ACM Transactions on Graphics* 32(4), 2013. Proceedings of Siggraph 2013, Anaheim (USA). [GGG<sup>+</sup>13]



L'idée de ce travail est de générer un réseau d'écoulement de rivières afin d'ensuite faire le terrain qui est compatible avec cet écoulement. Ceci a pour effet d'obtenir un terrain dont l'écoulement est réaliste et surtout possible, là où toutes les méthodes de l'état de l'art ne garantissent rien sur ce point.

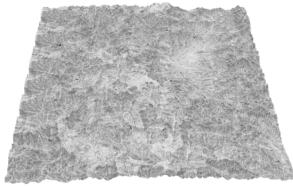
LARGE SCALE TERRAIN GENERATION FROM TECTONIC UPLIFT AND FLUVIAL EROSION. Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Adrien Peytavie & Eric Guérin. *Computer Graphics Forum* 35(2), 2016. Proceedings of Eurographics 2016, Lisbonne (Portugal). [CBC<sup>+</sup>16]



Dans cet article, nous utilisons la combinaison des deux phénomènes qui sculptent les montagnes à très grandes échelles spatiales et temporelles : la tectonique et l'érosion fluviale. Grâce à une équation tirée de la littérature géologique *stream power equation*, une approximation de ces deux phénomènes est possible et permet de générer des terrains à grande échelle

de manière réaliste.

PATCH-BASED TERRAIN SYNTHESIS. Leandro Cruz, Luiz Velho, Eric Galin, Adrien Peytavie & Eric Guérin. GRAPP 2015 – Proceedings of the 10th International Conference on Computer Graphics Theory and Applications – Berlin (Allemagne). [CVG<sup>+</sup>15]



laboratoire Visgraf.

L'idée de cet article est d'utiliser des méthodes de synthèse de textures reposant sur le découpage et le collage de tuiles. Appliquées à la synthèse de terrains, ces méthodes permettent de générer un terrain à partir d'un exemple et d'une version guide grossière du terrain. Ce travail a été le fruit d'une collaboration avec l'IMPA au Brésil, dans le

#### 1.9.4 Détails et variétés d'objets

ENVIRONMENTAL OBJECTS FOR AUTHORIZING PROCEDURAL SCENES. François Grosbellet, Adrien Peytavie, Eric Guérin, Eric Galin, Stéphane Mérillou & Bedrich Benes. *Computer Graphics Forum* 35(1), 2016. pp. 296-308. [GPG<sup>+</sup>16]



Dans ces travaux, les objets peuvent en même temps réagir à l'environnement autour d'eux et influencer cet environnement. Exemple : un lampadaire émet de la chaleur et va faire fondre la neige qu'il a reçue mais aussi celle d'un objet qui se trouverait à proximité. Ce framework permet de construire des scènes qui sont ensuite facilement paramétrables et modifiables afin de les visualiser dans différentes conditions saisonnières (feuilles dans les arbres au printemps, feuilles au sol à l'automne, neige en hiver, etc.).

EFFICIENT MODELING OF ENTANGLED DETAILS FOR NATURAL SCENES. Eric Guérin, Eric Galin, François Grosbellet, Adrien Peytavie & Jean-David Genevaux. *Computer Graphics Forum* 35(7), 2016. *Proceedings of Pacific Graphics 2016, Okinawa (Japon)*. [GGG<sup>+</sup>16]



Dans ce travail, nous introduisons un modèle permettant de décorer automatiquement les scènes naturelles avec de petits objets tels que des cailloux, des brins d'herbes, des branches, etc. Ce modèle appelé *ghosts tile* pré-calcule une grande quantité d'intersection dans un volume de référence. Ensuite, ce volume de référence est plongé dans la scène à décorer et les détails ajoutés ont la propriété de ne pas entrer en intersection, ce qui les rend plus réalistes.

COMPONENT-BASED MODEL SYNTHESIS FOR LOW POLYGONAL MODELS. Nicolas Marchal, Eric Guérin, Eric Galin & S. Akkouche. *Graphics Interface*, 217-224, 2010. [MGGA10]



À partir d'un objet unique, ce travail consiste à construire des variations de l'objet de manière semi-automatique afin de pouvoir de reproduire sans pour autant qu'il y ait de répétition visible. L'objet est découpé en morceaux et chaque morceau est déformé localement sans toucher à sa frontière. Le collage des morceaux par une méthode combinatoire permet d'obtenir une multitude de variantes sans pour autant être coûteux en mémoire.



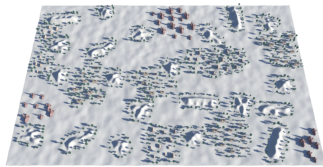
1.9.5 *Paysages*

**AUTHORING LANDSCAPES BY COMBINING ECOSYSTEM AND TERRAIN EROSION SIMULATION.** Guillaume Cordonnier, Eric Galin, James Gain, Bedrich Benes, Eric Guérin, Adrien Peytavie & Marie-Paule Cani. ACM Transactions on Graphics 2017. Proceedings of Siggraph 2017, Los Angeles, USA. [CGG<sup>+</sup>17]



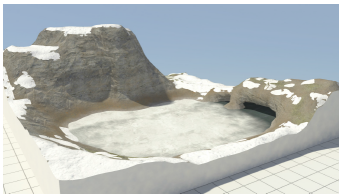
Ce travail récent introduit une méthode de simulation stochastique qui permet de combiner les effets d'érosion et la simulation d'écosystèmes. On peut ainsi représenter l'interaction qui existe entre la végétation et l'érosion : la végétation a tendance à ralentir l'érosion, et certains effets d'érosion peuvent détruire la végétation.

**LANDSCAPE SPECIFICATION RESIZING.** Leandro Cruz, Luiz Velho, Djalma Lucio, Eric Galin, Adrien Peytavie & Eric Guérin. CLEI 2014, Latin American Computing Conference, 15 septembre 2014, Montevideo (Uruguay). [CVL<sup>+</sup>14]



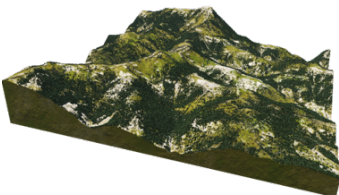
À partir d'une distribution d'objets sur une scène virtuelle, la méthode permet de modifier automatiquement la taille de la scène, la distribution d'objets étant automatiquement adaptée de manière cohérente sur le reste de la scène. Ce travail a été le fruit d'une collaboration avec l'IMPA au Brésil, dans le laboratoire Visgraf.

**HEAT TRANSFER SIMULATION FOR MODELING REALISTIC WINTER SCENERIES.** Nicolas Maréchal, Eric Guérin, Eric Galin, Stéphane Mérillou & Nicolas Mérillou. Computer Graphics Forum (Proceedings of Eurographics), 29(2), 449-458, 2010. [MGG<sup>+</sup>10]



Ce travail permet d'ajouter automatiquement des éléments de paysage hivernaux. Cela est effectué grâce à une simulation thermique précise couplée à l'utilisation d'un scénario climatique. Les éléments visuel ajoutés à la scène sont la neige et la glace sur les lacs. Il a été effectué dans le cadre du projet Genac 2.

**COHERENT MULTI-LAYER LANDSCAPE SYNTHESIS.** Oscar Argudo, Carlos Andujar, Antonio Chica, Eric Guérin, Julie Digne, Adrien Peytavie & Eric Galin. Visual Computer 2017. Proceedings of Computer Graphics International 2017, Yokohama, Japan. [AAC<sup>+</sup>17]



La modélisation parcimonieuse appliquée aux terrains a été introduite dans [GDGP16]. Dans ce travail, nous généralisons le modèle à des données contenant plus d'informations, comme la végétation, la texture ou l'application de styles particuliers. Ce travail a été effectué dans le cadre d'une collaboration avec l'UPC de Catalogne, au laboratoire ViRVIG.

### 1.9.6 Routes et réseaux de routes

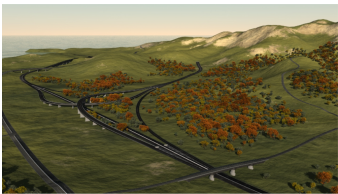
PROCEDURAL GENERATION OF ROADS. Eric Galin, Adrien Peytavie, Nicolas Maréchal & Eric Guérin. *Computer Graphics Forum (Proceedings of Eurographics)*, 29(2), 429-438, 2010. [GPMG10]



Cet article permet de calculer de manière automatique une trajectoire de route réaliste étant donné un terrain et deux points de départ et arrivée. Il repose sur l'utilisation d'un algorithme de plus court chemin dont la fonction de coût est anisotrope, sur un graphe de dimension topologique 3. Il permet ainsi de créer des trajectoires originales comme celles des

lacets dans les montagnes.

AUTHORING HIERARCHICAL ROAD NETWORKS. Eric Galin, Adrien Peytavie, Eric Guérin & Bedrich Benes. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 30(7), 2021-2030, 2011. [GPGB11]



Cet article généralise [GPMG10] en permettant de calculer des réseaux de routes hiérarchiques (autoroutes, nationales, départementales). L'algorithme repose sur l'utilisation d'un graphe de Gabriel, un sous-graphe de la triangulation de Delaunay.



## INTRODUCTION

J'ai effectué ma thèse (1999-2002) sous la direction d'Atilla Baskurt et Eric Tosan. Le sujet portait sur l'approximation de formes par un modèle fractal. Ces travaux ont mené à des résultats à la frontière entre le monde de l'analyse d'image, de l'informatique graphique et de la modélisation fractale. Les quelques années qui ont suivi ma thèse m'ont permis d'asseoir quelques résultats théoriques et d'en développer certains autres en lien étroit avec mon sujet de thèse.

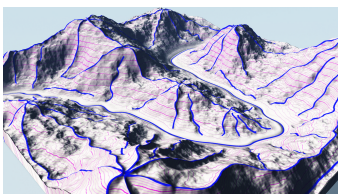
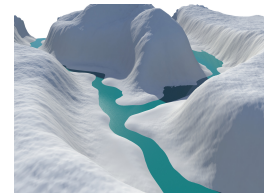
Depuis 2007, j'ai initié une nouvelle thématique portant sur la modélisation et la simulation de terrains. Cette ouverture thématique s'inscrivait dans le cadre plus large du thème de recherche portant sur la simulation de phénomènes naturels dans l'équipe Geomod.

Mes travaux combinent une expertise dans différents domaines (mécanique, mathématiques, informatique, géométrie) et j'ai à cœur de me situer à la frontière fertile de ces domaines.

Dans ce document, j'ai volontairement choisi de limiter les travaux que je présente à ceux qui concernent la modélisation de terrains. Il y a deux raisons principales à cela : d'une part, ce sont ceux qui représentent le plus de volume en terme de publications, d'autre part, cela permet de créer une cohérence thématique globale au document.

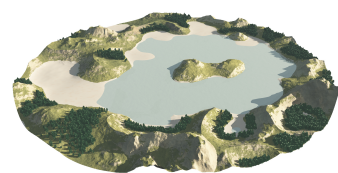
Le chapitre 3 donne un aperçu de l'état de l'art en matière de modélisation de terrains.

Le chapitre 4 reprend de manière synthétique les travaux de recherche consécutifs à ma thèse, donc en lien avec la modélisation géométrique fractale. Il s'agit là de deux contributions, l'une est un résultat théorique de différentiabilité du modèle utilisé au cours de ma thèse, l'autre concerne des travaux sur l'insertion de détail dans des modèles auto-similaires. Les résultats présentés seront uniquement ceux qui concernent la modélisation de terrains.



Dans le chapitre 5, je présente un modèle de génération de terrains basé sur l'équation aux dérivées partielles de la diffusion de la chaleur [HGA<sup>+</sup>10b]. Ce modèle permet de dessiner des courbes (*features*) qui décrivent les principales caractéristiques du terrain (lignes de crêtes et vallées) et de combler les zones non définies par un algorithme itératif. L'implémentation sur carte graphique donne des résultats interactifs.

J'exposerai ensuite dans le chapitre 6 un modèle de terrain fonctionnel et hiérarchique qui permet de construire des terrains par assemblage de portions de terrains [GGP<sup>+</sup>15]. Les feuilles de l'arbre de construction sont des primitives de terrains, tandis que les nœuds sont des opérateurs permettant de combiner plusieurs portions de terrains. Ce modèle a l'avantage d'offrir un contrôle précis ainsi qu'une description fonctionnelle rapide à évaluer.





Je présenterai ensuite dans le chapitre 7 un modèle de terrain qui utilise le formalisme parcimonieux [GDGP16]. Il s'agit de décrire un terrain par la combinaison de portions de terrains issues d'un dictionnaire. L'idée repose sur le fait qu'il faut un nombre limité d'atomes du dictionnaire pour représenter un bout de terrain. Ce modèle trouve des applications dans la modélisation procédurale inverse, la synthèse de terrains mais aussi l'amplification de terrains.

Enfin, je présenterai une synthèse et des perspectives de recherches dans le chapitre 8. De nouveaux challenges devront être surmontés dans le futur, notamment à cause d'une demande toujours croissante de mondes virtuels de plus en plus étendus. Ces challenges devront être résolus par de nouvelles méthodes, telles que la modélisation parcimonieuse, ou l'apprentissage par réseaux de neurones profonds.

La génération de mondes virtuels est un domaine très vaste qui peut faire appel à des techniques aussi variées que la simulation, l'instanciation de masse, les modèles procéduraux ou l'optimisation. Les objectifs recherchés sont le contrôle utilisateur, l'étendue du monde généré ainsi que le niveau de précision. Ce chapitre recense les travaux en informatique graphique dont l'objet de recherche est les terrains. L'accent sera mis sur les modèles de représentation utilisés plus que sur les techniques de génération ou de contrôle utilisateur.

### 3.1 FONDAMENTAUX EN GÉOMORPHOLOGIE

Dans cette section, nous allons passer en revue les phénomènes qui peuvent influencer l'aspect visuel d'un terrain. Ces phénomènes sont très nombreux et interviennent à des échelles temporelles et spatiales très variées. En nous focalisant sur les paysages naturels, nous allons mettre de côté l'action de l'homme sur les terrains (routes, barrages, agriculture, etc.). Notons toutefois que cette dernière peut avoir un impact visuel très important.

**OROGÉNÈSE.** Le relief à une échelle planétaire a été formé par les mouvements des plaques tectoniques. Ces énormes plaques naissent majoritairement dans les océans et peuvent être vues comme deux tapis roulants qui partent dans des directions opposées. De l'autre côté de ces plaques, des phénomènes de collisions apparaissent, faisant naître des montagnes. Ces phénomènes se déroulent à des échelles de temps très grandes (dizaines de millions d'années) car les mouvements associés sont lents (de l'ordre de quelques centimètres par an). Suivant le type de plaque (continentale ou océanique) qui entrent en collision, le massif sera de type alpin ou andin.

**ÉROSION.** Le relief est modifié de manière continue par une multitudes de facteurs d'érosions dont voici les principaux :

- Érosion thermique. Les fissures présentes dans les roches sont imprégnées d'eau, qui soumise au gel fait éclater la roche sous l'effet de la dilatation. Des blocs de roche se détachent et sont transportés sous l'effet de la gravité, formant des tas avec un angle au repos caractéristique (comme dans un pierrier). Ce phénomène peut même être présent en l'absence de gel, par simple chocs thermiques répétés, fragilisant la roche.
- Érosion liée à l'eau (hydrique<sup>1</sup>). L'eau omniprésente a un effet sur le relief dès lors qu'elle entre en mouvement. C'est le cas du mouvement des vagues sur une paroi rocheuse en bordure de mer ou de l'eau qui ruisselle dans un cours d'eau. De nombreux paramètres interviennent dans l'érosion hydrique, mais d'une manière générale, on peut le décrire par un transport de matière d'un endroit à un autre.
- Érosion éolienne. Il s'agit de l'effet du vent sur le relief. Ce phénomène est très souvent associé aux dunes de sable mais il intervient aussi dans d'autres contextes comme la formation des *goblins*. Un peu comme pour l'érosion hydrique, l'effet principal est le transport de matière d'un endroit à un autre mais cette fois provoqué par la force du vent.

1. la traduction exacte du terme anglais *hydraulic erosion* est érosion hydrique

- Érosion par la foudre. Lorsque la foudre frappe un relief, elle peut le briser et ainsi faire tomber des blocs rocheux. L'effet est assez similaire à celui de l'érosion thermique mais les localisations sont différentes.
- Érosion chimique, comme par exemple le *karst*. Ce type d'érosion va grignoter la roche en profondeur mais aussi en surface, par réaction chimique. Il est souvent à l'origine de la formation de grottes mais aussi de canyons.

Tous ces phénomènes interviennent à des échelles temporelles et spatiales plus petites que l'orogénèse. Récemment, un article propose de mettre en parallèle le mouvement d'élévation des montagnes (*uplift*) avec l'érosion hydrique, grâce à une équation connue en géologie sous le nom de *stream power equation* [CBC<sup>+</sup>16].

**LES PHÉNOMÈNES EXCEPTIONNELS.** Certains phénomènes ne sont pas répétitifs, ils se produisent de manière ponctuelle et accidentelle, mais peuvent profondément modifier un terrain. C'est le cas du glissement de terrain, de l'éruption volcanique ou du séisme.

**L'ÉCOSYSTÈME.** L'écosystème vient modifier les phénomènes naturels d'érosion et vice-versa. D'une manière générale, la végétation tend à faire diminuer l'effet de l'érosion hydrique et éolienne car elle stabilise les sols grâce aux racines. Inversement, les phénomènes naturels ont un effet sur l'écosystème. Par exemple, l'érosion hydrique transporte les sédiments dans les plaines et les rend propice au développement de végétation.

Il est virtuellement impossible de construire un modèle qui tiendrait compte de toutes les causes de la formation des reliefs de terrains qui composent la terre. De nombreux auteurs se sont focalisés sur une petite partie des phénomènes ayant un impact visuel fort (érosion hydrique notamment).

### 3.2 CARTOGRAPHIE DE LA LITTÉRATURE

La modélisation et la génération de terrains lancent quatre défis principaux :

1. représenter des terrains vastes avec de nombreux détails ;
2. prendre en compte les matériaux ;
3. contrôler les modelés ;
4. être réaliste d'un point de vue géomorphologique.

Nous verrons tout au long de ce document qu'il est difficile de satisfaire tous ces critères dans un seul modèle de représentation. Certains de ces défis sont même antagonistes.

Nous proposons ici de faire une cartographie des articles qui traitent de la modélisation ou de la génération de terrains dans l'optique de faire ressortir des tendances générales, particulièrement par rapport aux défis énumérés précédemment. Nous encourageons le lecteur à se référer à l'état de l'art présent dans [Gen15] pour une description plus précise des méthodes. L'article de synthèse [NLP<sup>+</sup>13] est aussi une très bonne lecture pour ce qui concerne les représentations utilisées pour les terrains ainsi que la géologie sous-terrainne.

Le tableau 6 en annexe A liste une soixantaine de travaux dont l'objet principal est la modélisation, la représentation ou la synthèse de terrains. Ce tableau inclut les travaux qui vont être présentés dans le présent manuscrit (à la fin du tableau, en rouge). Chaque colonne indique le niveau estimé de satisfaction de chacun des critères associés aux quatre défis :

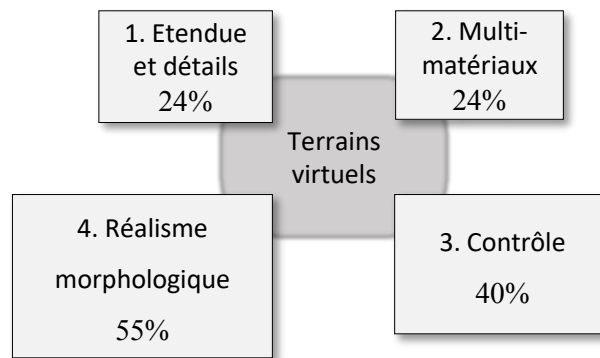


FIGURE 1 – Les différents défis ou critères et leur part dans l'état de l'art.

1. Étendue/détails : le modèle est-il capable de représenter des étendues vastes tout en associant un haut niveau de détails ?
2. Matériaux : le modèle permet-il de représenter différents matériaux ?
3. Contrôle : le modèle permet-il d'avoir un bon contrôle des modelés ?
4. Réalisme géomorphologique : le modèle permet-il de représenter des caractéristiques réalistes de terrains ?

Une pastille noire indique la présence forte du critère. Une pastille grise indique une présence moins marquée. La dernière colonne indique quel est le mode de représentation utilisé dans l'article en référence aux modèles que nous allons décrire dans les sections 3.3.1, 3.3.2 et 3.3.3.

Nous tirons plusieurs conclusions de ce comparatif (voir figure 1). Tout d'abord il y a un déséquilibre certain entre les critères. De nombreux auteurs se sont attachés au réalisme des terrain générés, très souvent au moyen de simulations. Ensuite, nous retiendrons qu'assez peu de travaux se sont intéressés à traiter de vastes étendues détaillées (moins d'un quart). Dans le même ordre d'idées, l'utilisation de modèles permettant de représenter plusieurs matériaux n'a pas été beaucoup abordée (moins d'un quart là encore). Le contrôle, un élément qui semble pourtant très important, n'a pas été énormément traité dans la bibliographie. **Enfin, sur la présence conjointe de critères, nous noterons que la modélisation de terrains vastes et détaillés ne rime pas avec le contrôle. C'est sur ce point que nous allons particulièrement axer nos contributions.**

### 3.3 MODÈLES DE REPRÉSENTATION

Dans cette dernière section de l'état de l'art, nous dressons une liste des différents modèles permettant de représenter un terrain virtuel.



### 3.3.1 Cartes de hauteurs et dérivés

Une carte de hauteur (*heightmap* ou *heightfield*) est un terme générique pour parler d'une fonction qui à un point d'un domaine 2D fait correspondre une altitude :

$$\begin{aligned} h : \mathcal{D} \subset \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (x, y) &\mapsto h(x, y) \end{aligned}$$

On peut qualifier ces représentations de 2,5D car avec cette description, le terrain peut être vu comme un intermédiaire entre un objet 2D et un objet 3D.

**FONCTIONS DE BRUIT PLANAIRES.** Une des manières les plus directes de construire une carte de hauteur est de trouver une fonction mathématique analytique  $h$  qui a des propriétés similaires à un terrain. Les auteurs se sont intéressés à ces représentations depuis longtemps. Le bruit de Perlin [Per85] ne permet pas à lui tout seul d'obtenir quelque chose qui ressemble à un terrain, mais lorsque que l'on combine plusieurs de ces bruits à différentes fréquences (notion de *turbulence*), les fonctions ont plus de caractéristiques communes avec un terrain. Les auteurs ont, au fil des années, essayé d'améliorer les fonctions de bruit afin qu'elles ressemblent plus à des terrains. En introduisant une valeur absolue dans la formule, il est possible de faire apparaître par exemple des lignes de crêtes (*ridged noise*). En diminuant l'amplitude des hautes fréquences dans les altitudes les plus basses, il est possible de reproduire un pseudo-effet de vallée [MKM89]. Certains auteurs ont même réussi à imiter un phénomène d'érosion en déplaçant les parois de la montagne artificiellement dans la direction et proportionnellement à l'intensité du gradient de la fonction [dCB09].

**IMAGE DE PROFONDEUR.** La difficulté de trouver des fonctions  $h$  qui représentent un terrain fait qu'en général on les représente sous la forme d'une image en niveaux de gris, c'est-à-dire une discrétisation de la carte de hauteur sur une grille rectangulaire :

$$\begin{aligned} h : \{0, \dots, m-1\} \times \{0, \dots, n-1\} &\rightarrow \mathbb{R} \\ (i, j) &\mapsto H_{ij} \end{aligned}$$

Son utilisation est très courante pour représenter les terrains car elle est simple à utiliser, à stocker, et les calculs de simulation que l'on peut effectuer par une méthode Eulérienne sont adaptés. Cette représentation est explicite, planaire et très souvent non-volumique. On veillera cependant à ne pas assimiler un terrain à une image, car pour transformer une image en terrain il faut toute une mécanique qui n'est pas évidente à maîtriser. Il faut notamment donner une échelle au terrain ( $x, y$  mais aussi en altitude  $z$ ). Il faudra aussi pour reconstruire une fonction s'assurer que l'on interpole correctement les données discrètes afin d'avoir de bonnes propriétés. Une interpolation bilinéaire permettra d'avoir une continuité de type  $C^0$  alors qu'une interpolation bicubique permettra d'avoir des raccords de tangentes et donc une fonction de continuité  $C^1$ .

**CARTES D'ISO-CONTOURS.** Très utilisés en cartographie pour mettre en emphase les pentes du terrain, les iso-contours indiquent sous forme de courbes tous les points du terrain qui possèdent une altitude identique. Le même principe peut aussi permettre de représenter un terrain. L'inconvénient de cette représentation est qu'il faudra une reconstruction pour le visualiser ou effectuer des calculs. C'est pourquoi cette représentation est plutôt non-explicite et nous sommes ici à la limite de ce que l'on peut appeler modèle de terrain. Elle est planaire et non-volumique. La plupart des travaux de recherche qui les utilisent traitent surtout de la reconstruction de terrains à partir d'iso-contours [HSS03].

**TINS** Les TINs (Triangular Irregular Networks) sont une tessellation d'un terrain par des triangles, dans une représentation planaire. Cela offre l'avantage de pouvoir avoir une densité de points (et donc de triangles) variable en fonction de la précision voulue. Certaines zones plates n'ont pas besoin de beaucoup de précision, alors que d'autres plus accidentées pourront contenir une densité de points plus élevée. On peut construire un TIN à partir d'un modèle de type image de profondeur afin de pouvoir optimiser son affichage par exemple. Les calculs de simulation ne seront en revanche plus possibles. On ne cherche en général pas dans ces représentations à obtenir une fonction, mais plus à avoir une façon optimisée du terrain pour l'affichage.

**SUBDIVISIONS** Parfois le modèle est directement un algorithme. C'est le cas des terrains obtenus par subdivisions récursives. L'algorithme le plus connu est une variante du *mid-point displacement* appelée *diamond-square*, ou ses dérivées *diamond-square-diamond* et *square-square* [Mil86]. L'idée est de faire un barycentre de plusieurs points et de perturber aléatoirement ce barycentre.

### 3.3.2 Représentation par couches

Afin de pouvoir représenter plusieurs matériaux ou des surplombs d'un même matériau, l'utilisation de couches successives peut être un bon compromis entre la carte de hauteur et la représentation volumique complète. Pour chaque position  $(x, y)$  du plan, plusieurs piles de matières sont représentées. Nous pouvons qualifier ces représentations de  $n_{2,5}D$  pour montrer qu'elles ne font qu'empiler les cartes de hauteurs abordées précédemment.

**MATÉRIAUX MULTIPLES.** Il s'agit dans ce modèle de généraliser l'image de profondeur en indiquant en chaque point de la grille une pile de matériaux plutôt qu'une simple hauteur (sous forme d'un tableau). La première utilisation de ce modèle à couches permet de faire des simulations d'érosion avec différents types de matériaux en plaçant des couches géologiques de duretés différentes [BF01].

**SURPLOMBES.** Si on ajoute comme type de couche l'air, alors il est possible de produire des surplombs, la roche pouvant elle-même être au dessus et en dessous de l'air [PGMG09]. Par nature, le modèle est composé de parallélépipèdes. Pour reconstruire une surface proprement dans le but d'une visualisation, on peut recourir à une convolution spatiale d'un noyau simple (un cube) avec le volume de matériau. L'application d'un classique *marching cube* permet de produire un maillage triangulaire.

### 3.3.3 Représentations 3D

Lorsque la topologie du terrain ne permet l'utilisation d'une représentation sous forme de grille bi-dimensionnelle ou de couches, on doit recourir à un modèle volumique complet. Avec ce type de modèle, toutes sortes de topologies peuvent être représentées, comme par exemple les arches (qui sont des anses topologiques), les grottes, et bien sûr les surplombs.

**FONCTIONS IMPLICITES.** Les surfaces implicites permettent de décrire des surfaces complexes grâce à l'usage de fonction dont on extrait une iso-surface par résolution d'une équation. Cette représentation est peu utilisée en modélisation de terrains.

**MAILLAGES TRIANGULAIRES.** Plus général que les TINs vus précédemment, les maillages triangulaires sont souvent utilisés en complément d'un modèle de carte de hauteur afin d'ajouter des exceptions topologiques locales. Cette technique est très utilisée dans le domaine du jeu vidéo car les outils de sculpture numérique 3D permettent de manipuler les maillages triangulaires. Cette représentation est explicite, non-planaire et souvent non volumique sauf à utiliser plusieurs maillages pour décrire les différents matériaux qui se trouvent sous le sol.

**MAILLAGES TOPOLOGIQUES.** Afin de représenter finement la structure interne sous-terrainne, on peut vouloir utiliser un modèle topologique complet, comme les G-cartes ou les cartes topologiques. Ce modèle encode la topologie de manière cohérente et permet de faire des opérations qui conservent cette cohérence.

**VOXELS.** Cette représentation consiste à partitionner l'espace suivant une grille régulière 3D et à indiquer pour chacun des éléments de volume (voxels) sa propriété. La plus simple des propriétés indique uniquement si de la matière se trouve ou non dans l'élément (valeur binaire). On peut généraliser cette information à plusieurs matériaux. La quantité de données devenant assez vite énorme, on associe souvent ces représentations à un algorithme de compression (*Run Length Encoding*, *Octree*, ou *Sparse Voxel Octree*).

### 3.4 CONCLUSION

La majorité des travaux de recherche en modélisation et synthèse de terrains virtuels fait un **compromis entre plusieurs critères**. Nous n'échappons pas à cette règle puisque nous présentons dans ce document des contributions qui mettent l'accent sur **les terrains vastes et détaillés, ainsi que sur le contrôle** tout en laissant de côté les aspects liés au réalisme géomorphologique. L'analyse de l'état de l'art nous montre que la combinaison de ces critères a été peu étudiée et rend ces contributions originales.

## AJOUTS DE DÉTAILS MULTI-RÉSOLUTION PAR MODÈLES FRACTALS

---

Ces travaux décrits dans ce chapitre ont été réalisés juste après ma thèse et durant la thèse de Houssam Hnaidi dans la période 2004-2010.

### 4.1 DIFFÉRENTIABILITÉ D'IFS PROJÉTÉS

Lors de ma thèse, j'ai appliqué des méthodes d'optimisation [GTBo1] afin de résoudre le problème inverse fractal. Ces méthodes s'appuient sur l'utilisation de la propriété de différentiabilité. Un de mes premiers travaux de recherche après la thèse a été de prouver cette propriété sur la sous-classe de fractales appelée IFS projeté [ZT96].

#### 4.1.1 Motivations

Les modèles fractals sont connus pour générer une grande quantité de détails de manière procédurale. Ils souffrent cependant d'un manque de contrôle intuitif entre la description du modèle et le résultat produit. Il est donc difficile pour un utilisateur non averti de modéliser la forme qu'il imagine. Les IFS projetés sont une réponse partielle à ce problème car ils permettent un contrôle double :

- La texture fractale est contrôlée par l'IFS, notamment l'aspect rugueux et la dimension fractale.
- La forme générale est contrôlée par des points comme pour une forme à pôle (courbe de Béziérs par exemple).

Malgré tout, l'utilisateur a envie de ne pas partir de rien dans son processus de création, et il aimerait disposer d'un outil qui lui permette de calculer un modèle approchant une figure de départ. Il s'agit là du problème inverse.

#### 4.1.2 IFS projeté

Rappelons tout d'abord le formalisme des IFS, ainsi que des IFS projetés.

Un IFS (Iterated Function System) est un outil permettant de générer des ensembles de points qui ont des propriétés fractales d'auto-similarité. Toute l'astuce de ce formalisme introduit par Hutchinson et Barnsley [Bar93] est d'appliquer le théorème du point fixe à un ensemble de compacts. Le théorème du point fixe nécessitant un espace métrique complet, ces auteurs ont imaginé la construction d'un ensemble de compacts  $\mathcal{H}(\mathcal{P})$  associé à la distance de Hausdorff basé sur un espace métrique complet initial  $\mathcal{P}$ . Les compacts de  $\mathcal{H}(\mathcal{P})$  représentent alors des figures de l'espace  $\mathcal{P}$ .

Alors que le théorème du point fixe appliqué dans  $\mathcal{P}$  nécessite une seule transformation contractante dans cet espace, lorsque l'on passe à  $\mathcal{H}(\mathcal{P})$ , c'est l'union de plusieurs transformations qui sera alors utilisé. Le compact associé à un IFS est appelé attracteur de cet IFS.

Dans la suite nous nous limitons aux transformations affines qui peuvent être représentées assez simplement par des matrices de transformation.

L'idée des IFS projetés est de travailler dans un espace barycentrique (la somme des coordonnées des points est égale à 1) pour produire un IFS avant de le projeter dans un espace de modélisation par l'intermédiaire de points de contrôle. L'IFS prend exactement la place des fonctions de mélange utilisées de manière traditionnelle dans les formes à pôles comme les courbes de Bézier ou les B-splines.

#### 4.1.3 *Théorème du collage*

Barnsley[Bar93] a démontré que n'importe quelle figure pouvait être approximée par un IFS. Il suffit pour cela de trouver un ensemble de transformations contractantes de rapport  $s$  qui une fois appliquées à la figure entière vont donner une distance de Hausdorff inférieure à une valeur  $\epsilon$ . Alors, la distance de Hausdorff entre la figure de départ et l'attracteur construit sera inférieure à  $\epsilon/(1-s)$ . Ce théorème a donné naissance à la compression fractale d'images [Fis95]. Néanmoins, il ne donne pas de méthode pratique pour construire un ensemble de transformations en fonction d'une figure à approximer.

#### 4.1.4 *Méthodes différentielles*

Différentes heuristiques peuvent être imaginées afin de résoudre le problème inverse fractal. Nous avons orienté nos recherches vers des méthodes numériques et plus particulièrement qui utilisent les propriétés de différentiabilité.

Il convient ici de bien définir le problème d'optimisation. Etant donné une figure que l'on veut approximer et une famille d'IFS projetés décrits par une série de paramètres numériques, nous voulons trouver quels sont les paramètres qui minimisent la distance entre l'IFS projeté et la figure.

La distance de Hausdorff habituellement utilisée dans le domaine des fractales ne donnera pas de résultat de différentiabilité de manière assez évidente. Nous avons considéré des figures dont la structure topologique est restreinte aux classes de courbes et de surfaces. Ainsi, les distances que nous utilisons pourront avoir des propriétés de différentiabilité.

La preuve de la différentiabilité est technique, j'en ferai uniquement un développement intuitif. Tout d'abord, écartons le problème de la projection des IFS qui est une opération linéaire donc différentiable. Penchons nous maintenant du côté de l'IFS qui est constitué d'opérations affines. Le problème réside dans l'application d'une infinité de ces opérations affines. Afin de prouver la différentiabilité et même plus, son analyticit , nous avons d compos  chaque op ration affine en une combinaison d'une op ration lin aire et d'une translation. Nous avons ensuite calcul  l'expression de chacune des diff rentielles que nous avons  valu es au point 0 dans un d veloppement de Mac-Laurin. Le r sultat du d veloppement de Mac-Laurin est identique   la fonction, nous avons conclu qu'elle  tait analytique, donc infiniment d rivable.

#### 4.1.5 *R sultats*

Le r sultat de ces travaux est th orique mais il consitue des fondements solides n cessaires aux m thodes que nous avons imagin es en mati re d'approximation fractale de

courbes et de surfaces. Les dérivées que nous calculons dans les optimisations sont des dérivées numériques d'ordre 1 et 2. Les développements que nous avons obtenus lors de la démonstration de différentiabilité n'ont pas permis d'accélérer ce calcul numérique par un calcul analytique étant donné que le nombre d'opérations matricielles restait le même. Le résultat de différentiabilité nous prouve autre chose : la modification d'un paramètre de l'IFS par l'utilisateur implique une modification de la figure engendrée de manière continue. Mais nous nous heurtons à un problème majeur pour l'édition de tels modèles : il n'existe pas de moyen intuitif d'éditer une matrice de subdivision. En effet, il est difficile d'en faire une représentation géométrique. C'est ce qui a motivé les travaux sur l'insertion de détail.

## 4.2 INSERTION DE DÉTAIL

Les IFS sont capables de produire une grande quantité de détails avec un nombre de paramètres assez limité. Malheureusement, cette expressivité est relative car tous les détails produits ont le même aspect visuel. C'est ce qui motive la création d'un nouveau modèle qui permet d'ajouter des modifications dans le calcul de la fractale à chaque itération de son calcul. Ce mode d'ajout de détail s'apparente aux ondelettes.

### 4.2.1 Modèles basé sur des IFS projetés

Une des façons d'afficher un IFS projeté est assez proche de l'algorithme de De Casteljau pour afficher une courbe de Bézières. Il consiste à considérer le polygone de contrôle initial et à le subdiviser par application de masques. Ce principe est assez voisin des surfaces de subdivision, la seule différence réside dans l'utilisation de masques spéciaux fractals.

Prenons le cas d'une courbe dont le polygone de contrôle est  $P$ . Ce polygone sera subdivisé en  $n$  sous-polygones  $P_0, \dots, P_{n-1}$  par application de  $n$  transformations  $T_0, \dots, T_{n-1}$  :

$$P_i = PT_i, i = 0, \dots, n - 1$$

L'insertion de détail peut être directement greffée dans cette équation en ajoutant au polygone une série de déplacements relatifs. Afin de garder un formalisme cohérent, nous avons défini ce déplacement comme étant le produit de deux matrices  $\delta P_i$  et  $U_i$  :

$$P_i = PT_i + \delta P_i U_i, i = 0, \dots, n - 1$$

Les matrices  $U_i$  peuvent être vues comme des matrices de déplacement de détail. Le gros avantage de cette formulation est que les matrices  $U_i$  peuvent être optimisées afin de mieux coller à une figure donnée. L'autre avantage est qu'un bon choix de taille pour ces matrices conduit à une formulation matricielle complète pour passer d'un niveau au suivant. Dans certains cas, cette formulation matricielle est même inversible rendant l'analyse directe : étant donné une liste de matrices  $T_i$  et  $U_i$ , quelle est la représentation du détail qu'il faut pour obtenir la figure souhaitée ?

Nous avons testé et utilisé ce modèle, il offre des avantages notamment pour ce qui est du problème inverse. Il est possible d'optimiser les matrices de subdivision, comme pour un IFS normal, mais aussi d'optimiser les matrices de déplacement de détail en fonction d'un objet à représenter.

Ces résultats prouvent cependant que ce modèle souffre d'un manque d'intuition pour ce qui est de son édition : les modifications ont un comportement global et n'ont pas d'interprétation géométrique directe. Cela nous a poussé à la formalisation d'un autre modèle basé sur les surfaces de subdivision.

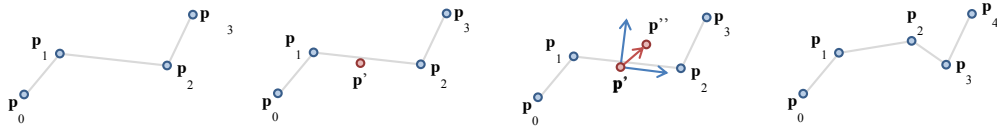
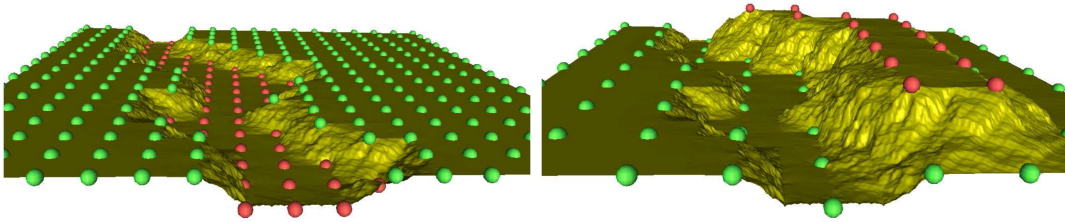
FIGURE 2 – Insertion d'un nouveau point entre  $p_1$  et  $p_2$ 

FIGURE 3 – Edition d'un terrain avec le modèle de surface de subdivision fractal et insertion de détail.

#### 4.2.2 Modèle basé sur les surfaces de subdivision

Dans ce modèle, nous simplifions la subdivision en la rendant interpolante. Le modèle se contente donc d'ajouter de nouveaux points sans modifier les points existants. Avec cette simplification, l'analyse devient évidente dès lors que la synthèse est explicitée.

La figure 2 illustre le principe de subdivision avec ajout de détail. Le point  $p'$  est créé par application d'un masque standard, par combinaison barycentrique des points  $p_0$ ,  $p_1$ ,  $p_2$  et  $p_3$ . Un repère local basé sur les points  $p_1$  et  $p_2$  est ensuite construit. Ce repère est ensuite utilisé pour exprimer le déplacement du point  $p'$  pour former le point  $p''$ .

Chaque segment peut ainsi être subdivisé (sauf le premier et le dernier dans le cas d'un masque de taille 4). Pour reconstruire la courbe, nous avons besoin du masque et des informations de détail à chaque niveau de subdivision. Inversement, pour retrouver les informations de détail, il suffit de calculer l'écart entre le point subdivisé par le masque et le point réel, et l'exprimer dans le repère local qui peut lui aussi être calculé facilement.

La généralisation de ce modèle aux surfaces (images de profondeur) est assez directe, il suffit de définir des masques qui agissent sur des grilles de points. Le détail quant à lui peut être exprimé directement comme une différence d'altitude entre le point subdivisé et le point réel.

### 4.3 ÉDITION MULTI-RÉSOLUTION SUR UN MODÈLE DE SUBDIVISION QUELCONQUE

Le principe d'édition multi-résolution peut être appliqué à n'importe quel algorithme de subdivision récursive, et non uniquement les IFS projetés. Prenons l'exemple du schéma de subdivision de type *mid-point subdivision* pour les courbes ou *diamond-square* [Mil86] pour les surfaces. Bien que basé sur des déplacements aléatoires des points subdivisés, on peut facilement imaginer un déplacement de ces points de manière forcée, après coup. Ainsi, le modèle peut être généré uniquement grâce aux informations de perturbation des points.



FIGURE 4 – Exemple de terrain final obtenu par un modèle de surface de subdivision fractal et insertion de détail.

Pour représenter une perturbation, il suffit de stocker un niveau de subdivision, des coordonnées d'indices et la valeur relative de la perturbation.

En ajoutant à ce modèle une carte grossière d'amplitude de perturbation, on obtient un modèle de terrain compact et adapté à l'édition interactive.

Différents outils d'édition multi-résolution permettent de tirer parti de la structure hiérarchique et des possibilités de raffinement par le modèle fractal, et particulier :

- Elever/baisser une portion de terrain (boule de rayon variable) ;
- Mettre le terrain à un niveau donné (boule de rayon variable + plan indiquant la hauteur) ;
- Augmenter/diminuer la rugosité (boule de rayon variable) ;
- Crêtes : cet outil permet de tracer une crête à un niveau de hauteur qui diminue avec la distance parcourue ;
- Rivière : creuse localement le terrain (boule de rayon variable) ;
- Gomme : permet d'enlever toutes les informations de détail sur une zone donnée (boule de rayon variable).

La boule de rayon variable permet de sélectionner automatiquement le niveau d'édition grâce à un seuil (paramétrable) qui détermine le nombre de points minimum à éditer.

L'édition à l'aide de cet outil s'avère rapide, et le modèle est assez léger en termes de stockage.

La figure 5 montre les étapes d'édition d'un canyon et deux affluents qui se rejoignent avec l'outil développé. La totalité de la session d'édition n'a duré que quelques minutes. Le terrain final avec tous ses détails est stocké dans un fichier de seulement 19 *kio* et comporte 2527 modifications. Les terrains générés comportent une grille de  $4 \times 4$  points au départ et ont été subdivisés 8 fois pour une grille finale de  $769 \times 769$ . Le temps de génération pour cette précision est de 100 *ms* et ne dépend quasiment pas du nombre d'insertions de détails. Ce temps de génération est compatible avec une édition interactive (à chaque modification de l'utilisateur, le terrain est recalculé entièrement).



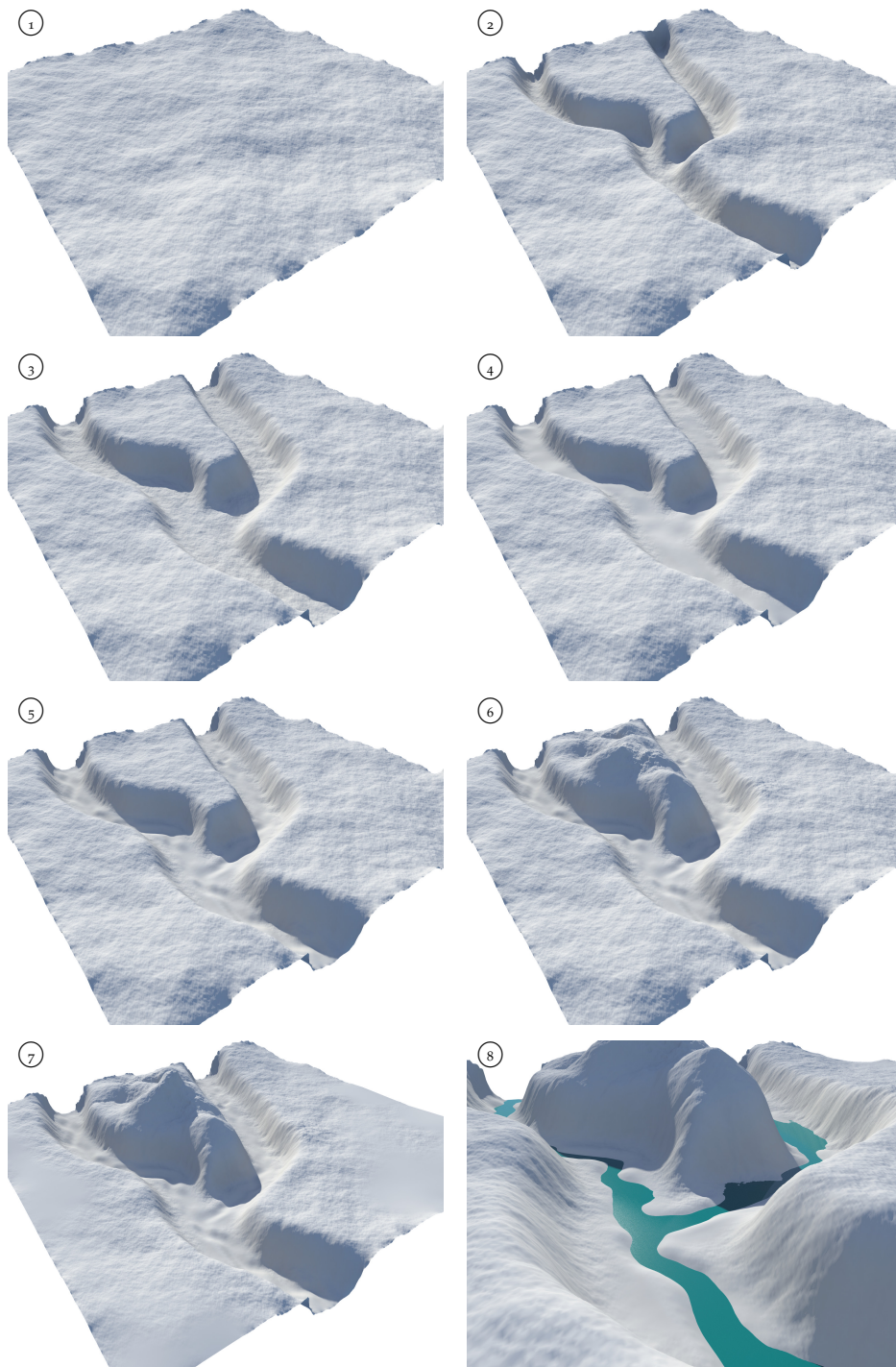


FIGURE 5 – Processus de création. ① le terrain initial est plat avec un niveau de rugosité homogène. ② l'utilisateur crée une faille grâce à l'outil mettre le terrain à un niveau donné. ③ l'utilisateur affine la faille avec le même outil. ④ l'utilisateur rend le fond du canyon moins rugueux. ⑤ l'utilisateur crée le lit de la rivière grâce à l'outil mettre le terrain à un niveau donné (un peu plus bas que le niveau de la faille). ⑥ l'utilisateur rend la zone de confluence plus montagneuse. ⑦ l'utilisateur creuse quelques rivières le long des montagnes. ⑧ un autre point de vue, avec une rivière.

#### 4.4 CONCLUSION

Les formes fractales ont un côté théorique fascinant et en correspondance avec des phénomènes naturels observés mais souffrent d'un manque de contrôle qui les rend difficilement utilisables dans un contexte de création artistique. Nous avons montré dans ce chapitre que moyennant l'utilisation de modèles dérivés tels que les IFS projetés et l'insertion de détails, ces inconvénients se gomment et le modèle devient tout à fait contrôlable. Au sein de l'équipe, nous travaillons encore activement au développement de méthodes de subdivision contrôlées à l'aide de détail. Une idée qui pourrait être intéressante serait de combiner ce formalisme avec celui utilisé dans le chapitre 7 sur la modélisation parcimonieuse. En effet, ce dernier offre de nombreux avantages mais n'intègre pas d'édition interactive.

Dans la chapitre suivant, nous introduisons un modèle qui offre un contrôle utilisateur très précis en utilisant l'équation de la chaleur pour générer des terrains à partir de caractéristiques telles que les lignes de crêtes ou les rivières.



## TERRAINS DE DIFFUSION

Ce travail est inspiré d'un article qui permet de colorier automatiquement un dessin vectoriel [OBW<sup>+</sup>08]. Cela consiste à utiliser l'équation de la diffusion de la chaleur (une équation différentielle simple et stable) afin de remplir des zones inconnues. Pour [OBW<sup>+</sup>08], il s'agit de colorier des zones du dessin avec la contrainte des coups de crayons et une couleur de chaque côté. Pour les terrains, il s'agit de remplir des zones de terrains avec des contraintes d'altitude sur les lignes fortes (crêtes, crevasses) ainsi que des pentes de chaque côté de ces lignes.

Ce travail a fait l'objet d'une publication à la conférence Pacific Graphics 2010 [HGA<sup>+</sup>10b]. Il a été effectué dans le cadre de la thèse d'Houssam Hnaidi.

## 5.1 PRINCIPE

L'utilisateur fournit une carte vectorielle des contraintes qu'il veut appliquer. Ces contraintes sont exprimées sous la forme d'une liste de courbes sur lesquelles sont indiquées des contraintes (voir figure 6). Les contraintes vectorielles sont ensuite traduites sous la forme de plusieurs cartes discrétisées :

- Une carte de paramètres de bruits permettant de contrôler le spectre et l'amplitude du bruit qui sera généré.
- Une carte de gradient qui définit et va guider la diffusion dans la dernière étape de l'algorithme.
- Une carte d'élévation qui contient les contraintes strictes d'altitudes qui ont été indiquées sur les courbes de contrôle.

Toutes ces cartes sont obtenues par un processus simple de *rasterisation* grâce à un rendu 2D (dans un *stencil buffer*) de primitives de type segment ou polygone dont les couleurs correspondent aux différentes contraintes. Chaque courbe possède une liste de nœuds de contrainte pour lesquels les informations suivantes sont stockées (voir figure 7) :

- contrainte d'altitude ( $h_i, r_i$ ) où  $h_i$  est l'altitude et  $r_i$  est le rayon d'influence ;
- contrainte d'angle ( $a_i, b_i, \theta_i, \phi_i$ ) où  $\theta_i$  et  $\phi_i$  sont des angles et  $a_i, b_i$  des rayons d'influence ;
- contrainte de bruit ( $\mathcal{A}_i, \mathcal{R}_i$ ) où  $\mathcal{A}_i$  représente l'amplitude du bruit et  $\mathcal{R}_i$  sa rugosité.

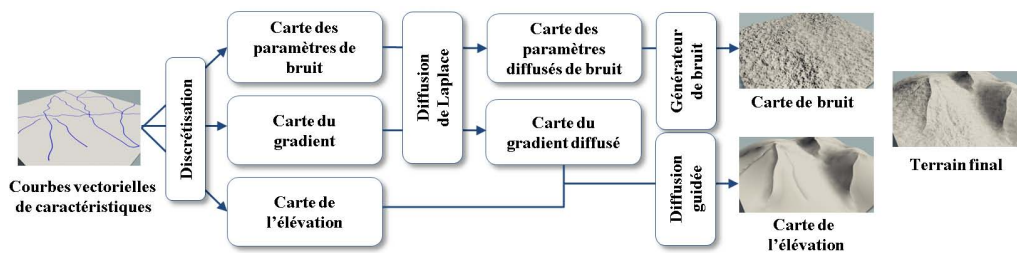


FIGURE 6 – Processus de construction du terrain par diffusion.

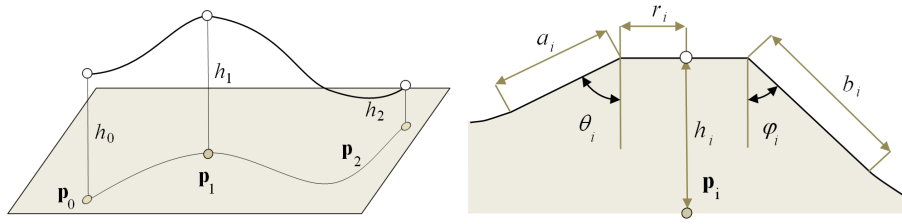


FIGURE 7 – Contraintes attachées aux courbes de contrôle.

Une fois que les cartes de contraintes sont discrétisées, nous leur appliquons une diffusion pour remplir les zones vides. Deux types de diffusion sont utilisés :

- Diffusion de Laplace simple. Cette diffusion prend en compte uniquement l'équation de diffusion (Laplacien nul) et des contraintes de Dirichlet (valeurs imposées).
- Diffusion guidée par le gradient. Cette diffusion prend en compte en plus des contraintes de type Neumann c'est-à-dire sur le gradient.

Les paramètres de bruit et de gradient subissent une diffusion simple. La carte d'élévation est calculée grâce à une diffusion guidée par la carte de gradient diffusée.

Finalement, la carte de paramètre de bruits est traduite par une carte de bruit et le bruit est ajouté à l'élévation pour former le terrain final.

## 5.2 DIFFUSION

### 5.2.1 Équation de diffusion de la chaleur

L'équation de diffusion de la chaleur est une équation aux dérivées partielles parabolique. Soit  $F$  une fonction spatiale (ici dans le plan) qui décrit le champ de température. Ce champ évolue aussi dans le temps :

$$\begin{aligned} F : \mathbb{R}^2 \times \mathbb{R} &\rightarrow \mathbb{R} \\ (\mathbf{p}, t) &\mapsto F(\mathbf{p}, t) \end{aligned}$$

Si le champ de température  $F$  a une valeur initiale à l'instant 0 qui est donnée  $F(\mathbf{p}, 0) = F_0(\mathbf{p})$ , alors son évolution dans le temps sera régie par l'équation aux dérivées partielles suivante :

$$\frac{\partial F}{\partial t}(\mathbf{p}, t) = D \Delta F(\mathbf{p}, t)$$

Cette équation est valable si aucune source de chaleur n'existe. Le coefficient  $D$  indique le coefficient de diffusivité thermique du matériau (qui peut être un champ spatial). Le symbole  $\Delta$  est l'opérateur de Laplace (spatial) qui correspond à la somme des dérivées partielles d'ordre deux. Ce que cette équation traduit c'est que plus les écarts de température sont importants plus l'évolution dans le temps va être importante et ce proportionnellement à la conduction du matériau considéré. Le sens de l'évolution tend vers une homogénéisation des valeurs. Si on laisse le système évoluer asymptotiquement et qu'il converge donc n'évolue plus dans le temps, la partie gauche de l'équation devient nulle et que le Laplacien du champ de température devient nul. L'intérêt de l'équation est plus grand lorsque l'on fixe des conditions aux limites du domaine. Les conditions aux limites de Dirichlet permettent d'imposer une valeur figée de  $F$  sur la bordure du domaine. Pour l'équation de diffusion de

la chaleur, cela revient à fixer une valeur de température à certains points. Les conditions aux limites de Neumann permettent de fixer en certains points la direction et l'amplitude du gradient. Traditionnellement, la direction est celle de la normale à la frontière du domaine. Si l'on veut par exemple modéliser qu'à la frontière du domaine il n'y a aucun échange thermique possible, on fixe la valeur du gradient à 0 sur le bord du domaine (l'équivalent d'une bouteille thermos parfaite).

### 5.2.2 Construction de la solution

Pour construire la solution de l'équation aux dérivées partielles, il est assez courant d'avoir recours à une simulation qui va imiter le comportement du système au cours du temps.

Ainsi, en choisissant un pas de temps  $\delta t$  suffisamment petit, nous pouvons simplifier l'équation en la rendant linéaire :

$$F(\mathbf{p}, t + \delta t) = F(\mathbf{p}, t) + \delta t D \Delta F(\mathbf{p}, t)$$

Dans un schéma d'intégration discret sur une grille (schéma Eulérien) de maille  $h$ , cette équation se transforme en :

$$F(i, j, t + \delta t) = F(i, j, t) + \frac{\delta t D}{h^2} (F(i+1, j, t) + F(i-1, j, t) + F(i, j+1) + F(i, j-1, t) - 4F(i, j, t))$$

La valeur particulière du pas de temps  $\delta t = h^2/4D$  permet de simplifier énormément l'équation et garantit la stabilité de la simulation. En effet, un pas de temps plus grand ferait prendre le risque d'obtenir une valeur résultante en dehors de l'intervalle des valeurs présentes, solution qui est impossible. Cette valeur correspond au cas où le pas de simulation temporel et spatial ont pour effet de faire la moyenne exacte des valeurs du 4-voisinage. Pour exemple, dans le cas du fer ayant une conductivité de  $22.8 \cdot 10^{-6} \text{ m}^2 \text{ s}^{-1}$  et d'une maille d'un millimètre, il faudrait un pas de temps de 11 ms pour atteindre ce cas de figure. Dans la suite de l'exposé, nous utiliserons la notation suivante :  $F^k(i, j) = F(i, j, k\delta t)$  pour plus de lisibilité.

### 5.2.3 Application aux terrains

Nous cherchons à construire une fonction dont nous connaissons uniquement quelques contraintes. Autrement dit, nous cherchons un moyen de remplir les trous. Ce remplissage doit produire une fonction continue. L'inconvénient de l'équation de la diffusion est qu'elle produit des fonctions harmoniques (deux fois dérivables et dont la dérivée seconde est nulle). Or un terrain par nature ne peut pas être décrit par une fonction harmonique, il comporte des discontinuités de tangente et donc la fonction qui le décrit n'est même pas de classe  $C^1$  en pratique. Nous verrons dans la suite de ce document qu'un terrain pourrait être plutôt assimilé à une fonction Lipschitzienne. C'est pour cela que nous avons introduit un modèle où nous pouvons fixer des contraintes de tangente, mais qui ne sont pas forcément continues.

Concrètement, cela revient à dire qu'en chaque point, l'équation aux dérivées partielles est la suivante :

$$\nabla F(\mathbf{p}) = G(\mathbf{p})$$

où  $G$  est notre fonction de contrainte de gradient (il s'agit d'un vecteur). Cette équation représente la solution que nous voudrions avoir à un équilibre du système et n'a donc pas de variable temporelle. Étant donné que la construction de la solution va reposer sur une résolution itérative, nous allons essayer de trouver une équation faisant intervenir le temps et dont la solution serait compatible avec notre contrainte de gradient, autrement dit, une équation qui fait se rapprocher de la solution à chaque fois que l'on itère. Prenons tout simplement la valeur de  $F$  dans la direction inverse du gradient et ajoutons lui une valeur qui dépend de la norme du gradient :

$$F(\mathbf{p}, t + \delta t) = F(\mathbf{p} - h G(\mathbf{p}), t) + h \|G(\mathbf{p}, t)\|$$

Dans la version discrète, nous utiliserons pour valeur de  $h$  la taille de la maille de discrétisation et nous ferons une approximation bilinéaire de  $F(\mathbf{p} - h G(\mathbf{p}), t)$ . Appliquée à un terrain, cette équation revient à dire que l'on calcule la nouvelle élévation en allant voir l'élévation d'un point dans le voisinage que l'on élève d'une hauteur proportionnelle au gradient donc à la pente. C'est bien ce que nous cherchons à construire.

Le dernier type de contrainte que nous allons introduire dans notre système est une contrainte de Dirichlet qui consiste à imposer une valeur fixe à un point donné par la fonction  $H$  :

$$F^{k+1}(\mathbf{p}) = H(\mathbf{p})$$

En pratique, certains points vont avoir des contraintes de Dirichlet, d'autres des contraintes de gradient et pour tous les autres points l'équation de diffusion sera utilisée. Afin de garantir que la solution est bien continue, nous devons passer d'un type de contrainte à une autre de manière continue. C'est pourquoi nous avons introduit un masque de mélange qui consiste à définir la manière dont nous mélangeons les différentes contraintes, avec des coefficients  $\alpha$ ,  $\beta$  et  $(1 - \alpha - \beta)$  :

$$F^{k+1}(i, j) = \alpha(i, j)F_L^{k+1}(i, j) + \beta(i, j)F_G^{k+1}(i, j) + (1 - \alpha(i, j) - \beta(i, j))F_D^{k+1}(i, j)$$

où  $F_L^{k+1}(i, j)$  est la composante calculée par l'équation de diffusion,  $F_G^{k+1}(i, j)$  est celle calculée par la contrainte de gradient et enfin  $F_D^{k+1}(i, j)$  est la contrainte de Dirichlet.

Afin de garantir leur continuité, les champs  $G$ ,  $\alpha$  et  $\beta$  sont eux mêmes calculés par la diffusion de contraintes ponctuelles [HGA<sup>+</sup>10b].

### 5.3 BIBLIOTHÈQUE DE MODELÉS

Nous avons construit une bibliothèque de modelés qui permet de représenter la plupart des caractéristiques visibles dans un terrain naturel. Ces modelés se présentent sous la forme de courbes de contrôle qui vont avoir un effet sur les champs  $H$ ,  $G$ ,  $\alpha$  et  $\beta$ . D'une manière générale, sur la courbe, nous plaçons des contraintes de type Dirichlet avec  $\alpha = \beta = 0$ . Proche de la courbe, la valeur de  $\beta$  va être de 1 puis progressivement diminuer lorsque l'on s'en éloigne pour laisser place à la diffusion.

La figure 8 montre des exemples typiques des réglages utilisés pour représenter ces caractéristiques de terrains. Nous ajoutons à ces modelés la carte de bruits dont les paramètres sont générés aussi par les paramètres de bruits diffusés.

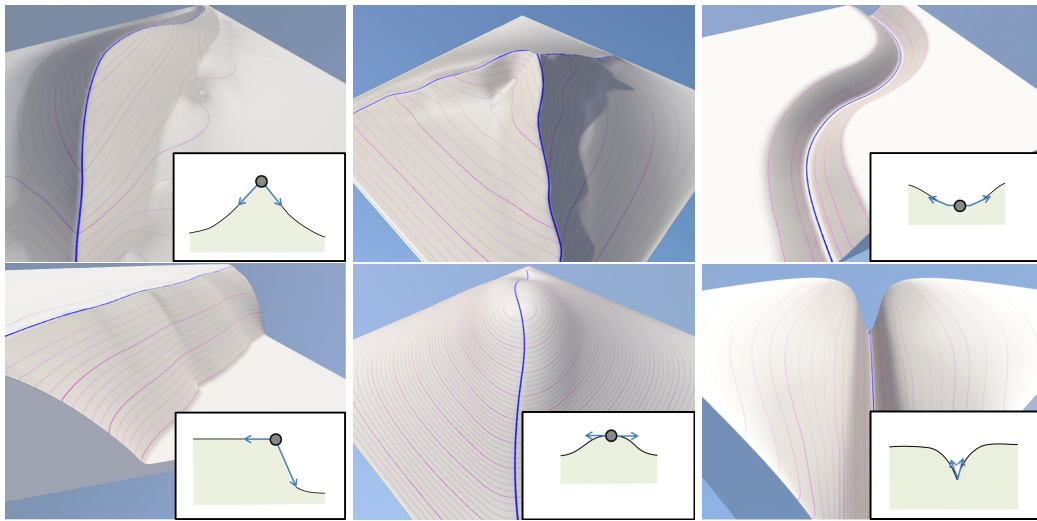


FIGURE 8 – En utilisant des valeurs particulières de gradient, nous pouvons reproduire des caractéristiques de terrains. De gauche à droite et de bas en haut : ligne de crête simple, ligne de crête triple, lit de rivière, falaise, colline, faille.

#### 5.4 IMPLÉMENTATION

Ces opérations sont faites dans un solveur multi-grille sur GPU [HGA<sup>+</sup>10b]. L'idée du solveur multi-grille est de résoudre l'équation de manière approchée à une résolution grossière, puis de raffiner successivement cette solution en diminuant la taille de la maille spatiale. Ce type de solveur nécessite l'implémentation d'un sous-échantillonnage de la carte de contrainte, et d'un sur-échantillonnage de la solution.

#### 5.5 RÉSULTATS

**CONTRÔLE.** Cette méthode allie simplicité d'utilisation et contrôle. La représentation des formes caractéristiques est très compacte en mémoire car ne nécessite que quelques courbes pour décrire des scènes complexes. Le tableau 1 donne le nombre de courbes nécessaires à la modélisation des différentes scènes. Notons que ces tailles ne sont pas dépendantes de la résolution du terrain, qui elle est arbitraire.

La figure 9 montre trois étapes de création d'une scène complexe de canyon. La première étape grossière n'a nécessité que quelques minutes et donne les lignes principales du terrain. Pour arriver à l'étape finale, il a fallu environ 45 minutes, et la plupart du temps a été utilisé à régler de manière précise les paramètres de bruit pour qu'ils collent à la pente.

**RÉALISME.** Cette méthode permet de construire des terrains de nature assez variée, allant d'une montagne à des dunes dans un désert (figure 10). Le réalisme des résultats est en partie dû au fait que notre perception est très sensible aux caractéristiques fortes du terrain telles que les lignes de crêtes et les crevasses dues à l'érosion. Or notre méthode est justement axée sur la modélisation de ces objets. Notre perception est aussi sensible au réalisme géologique et hydrographique du terrain. Dans cette méthode, la structure du terrain est entièrement produite par édition ce qui permet de construire des terrains qui sont d'emblée



Modèle	Nombre de courbes	Stockage
Montagne	59	2,89
Lac	45	2,26
Canyon	74	4,48
Collines	55	2,25
Île volcanique	48	2,95
Désert	26	1,3

TABLE 1 – Statistiques des modèles vectoriels utilisés pour représenter les différentes scènes. Le stockage est exprimé en kilo-octets.

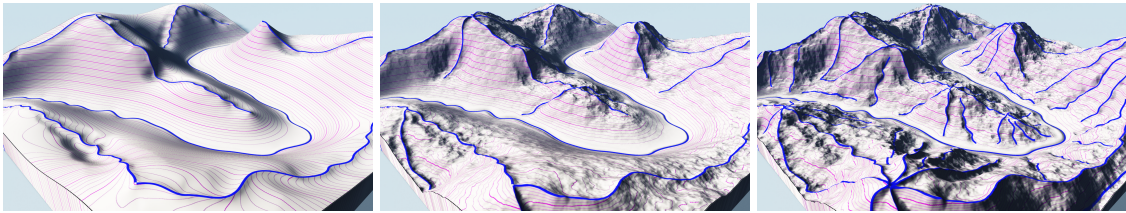


FIGURE 9 – Trois étapes de modélisation successives pour arriver à un canyon.

réalistes. La figure 11 montre un exemple d'une île volcanique modélisée par notre méthode. Là encore, la puissance de notre méthode est de nécessiter peu de courbes pour obtenir un réalisme correct.

PERFORMANCES. La table 2 indique les performances de l'algorithme à diverses résolutions de grilles. Ces performances ont été obtenues avec une carte graphique NVidia 8800 GTS. On pourrait donc certainement attendre des résultats bien meilleurs avec une carte graphique plus récente. Ces temps de calcul inférieurs à la seconde permettent l'édition interactive des courbes de contrôle avec un affichage direct du résultat. Notre implémentation nécessite le stockage de 10 cartes de la résolution finale sur le GPU, ce qui explique que nous n'ayons pas réussi à monter au delà de 2048 comme résolution. Avec des

## 5.6 CONCLUSION

Le modèle proposé est très contrôlable et permet de représenter des terrains de natures assez variées. Il se trouve limité à produire en sortie des cartes de hauteurs discrètes dont

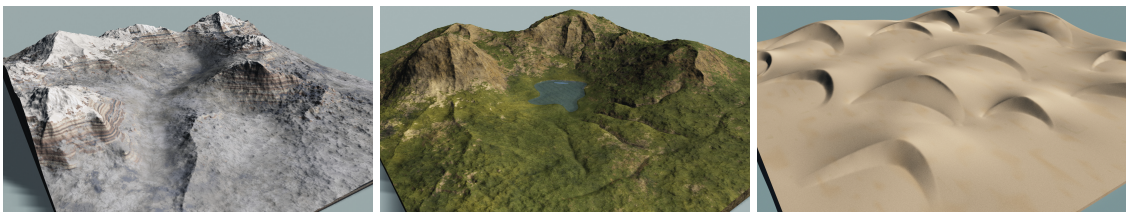


FIGURE 10 – Paysages de montagne, lac, et désert.



FIGURE 11 – Une île volcanique et les courbes qui permettent de la représenter.

Modèle	Temps de génération		
	512 <sup>2</sup>	1024 <sup>2</sup>	2048 <sup>2</sup>
Montagne	0,187	0,270	0,670
Lac	0,185	0,261	0,690
Canyon	0,216	0,339	0,811
Collines	0,188	0,266	0,611
Île volcanique	0,168	0,275	0,675
Désert	0,171	0,253	0,635

TABLE 2 – Temps en secondes pour générer les différents terrains à des résolutions de grille de 512, 1024 et 2048.

la résolution est limitée du fait de la mémoire de la carte graphique. On notera cependant que rien n'empêche avec une implémentation sur CPU de casser cette limitation. L'une des limitations les plus importantes réside aussi dans le fait que l'on ne peut calculer le terrain que dans sa globalité. Chaque courbe caractéristique peut en effet avoir un effet sur tout le terrain entier.

Afin de proposer un modèle qui soit réellement continu et puisse donc produire des terrains à des résolutions illimitées, nous avons proposé et développé un modèle de représentation qui combine des fonctions permettant un contrôle local grâce à une approche constructiviste. C'est l'objet du prochain chapitre.



L'idée de ces travaux a germé alors que nous travaillions sur un modèle de génération de terrain correct du point de vue hydrologique [GGG<sup>+</sup>13]. Nous avons alors à notre disposition des informations de haut niveau sur le réseau hydrologique du terrain, ainsi que des informations assez grossières sur les altitudes dans des cellules de terrain associées. Comment construire un modèle de terrain à partir de ces méta-données ? L'utilisation des terrains de diffusion présentés précédemment posait un problème, ils n'étaient pas adaptés à des terrains immenses du fait du caractère global de l'algorithme de reconstruction.

Ce travail a d'abord été effectué dans le cadre d'une publication à Siggraph en 2013 [GGG<sup>+</sup>13] et a ensuite été développé dans la revue Computer Graphics Forum en 2015 [GGP<sup>+</sup>15]. Il a été effectué dans le cadre de la thèse de Jean-David Gènevaux.

### 6.1 PRINCIPE

Il fallait donc un modèle qui permette de reconstruire une portion seulement du terrain, avec une précision arbitraire, et pouvant utiliser des données générées. Les fonctions procédurales sont un bon outil pour les deux premiers critères, mais elles n'offrent en revanche pas un contrôle suffisant pour pouvoir les exploiter dans le cadre de méta-données générées. Nous avons donc imaginé un système constructiviste (arbre de construction) de terrain fonctionnel qui permette de pallier ce problème.

La figure 12 montre le fonctionnement général de ce modèle de représentation. Ce modèle tire sa versatilité des multiples sources qui permettent de l'alimenter. Outre les données procédurales, qui sont historiquement celles qui ont poussé la création du modèle, nous sommes capables d'utiliser des données réelles de terrain (modèles numériques d'élévation), mais aussi de laisser l'utilisateur modifier intuitivement les caractéristiques du modèle. En termes de formats de sortie, là aussi les possibilités sont multiples, du simple export de maillage visualisable sur GPU, CPU ou n'importe quel logiciel de modélisation 3D, au rendu en temps-réel sur GPU par une méthode de *ray-marching*.

Le principe du modèle repose sur le fait que des éléments de terrains sont combinés incrémentalement pour former des terrains de plus en plus grands. La construction sous forme d'arbre hiérarchique présentée ici est très adaptée à ce genre de modélisation.

### 6.2 MODÈLE HIÉRARCHIQUE DE TERRAIN

Notre modèle s'appuie sur une hiérarchie d'éléments qui définissent tous deux fonctions. La première fonction  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  définit l'altitude du terrain  $f(\mathbf{p})$  (ou de la portion de terrain) au point  $\mathbf{p} \in \mathbb{R}^2$ . La deuxième fonction est une fonction de poids  $\alpha : \mathbb{R}^2 \rightarrow \mathbb{R}^+$  indique la façon dont l'élément de terrain sera combiné dans son environnement. Cette fonction de poids définit un domaine compact  $\Omega^0$  pour lequel la fonction est strictement positive :

$$\Omega^0 = \{\mathbf{p} \in \mathbb{R}^2 \mid \alpha(\mathbf{p}) > 0\}$$

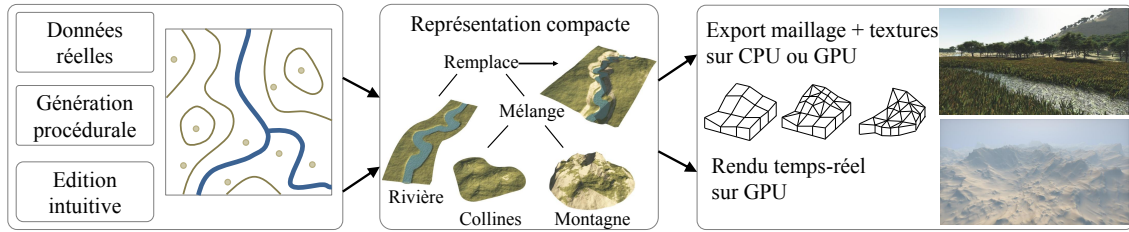


FIGURE 12 – Fonctionnement général du modèle de terrain fonctionnel constructiviste.

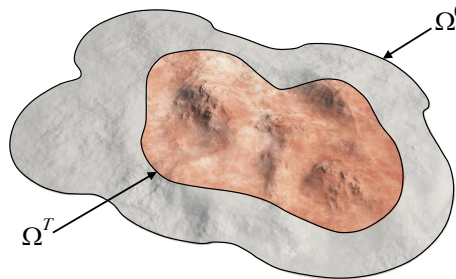


FIGURE 13 – Les deux domaines associés à un élément de terrain.

La fonction  $f$  se doit d'être définie sur  $\Omega^0$ . Nous définissons un deuxième domaine associé à une valeur  $T > 0$ , qui est inclus dans  $\Omega^0$  et qui définit le domaine du terrain (celui qui sera visualisé) :

$$\Omega^T = \{\mathbf{p} \in \mathbb{R}^2 \mid \alpha(\mathbf{p}) \geq T > 0\}$$

La figure 13 illustre ce principe. Dans tout le reste du document, nous ne montrerons pas la zone de transition  $\Omega^0 \setminus \Omega^T$  qui apparaît en gris sur la figure. Dans notre implémentation, nous avons utilisé la valeur  $T = 1/2$ . Ce modèle fonctionnel permet de construire une représentation ensembliste du terrain de la manière suivante :

$$\mathcal{T} = \{(\mathbf{p}, f(\mathbf{p})) \in \mathbb{R}^3, \mathbf{p} \in \Omega^T\}$$

La structure en arbre permet d'adapter l'évaluation de la fonction d'élévation en tenant compte d'un niveau de détail souhaité. En plus de cela, nous allons garantir par construction que toutes les fonctions d'altitude  $f$  sont Lipschitziennes, et que les fonctions de poids  $\alpha$  sont de classe de continuité  $C^1$ . Ces deux propriétés vont nous permettre d'accélérer les calculs de visualisation. Enfin, le fait d'avoir un arbre de construction nous permet aussi de construire une hiérarchie de volumes englobants qui va elle aussi engendrer une accélération des calculs.

Pour calculer la normale au terrain, nous utilisons le gradient de la fonction  $f$ . Ce gradient peut être calculé de deux manières. La première possibilité est de faire une approximation numérique comme suit :

$$\nabla f(\mathbf{p}) \simeq \frac{1}{2\varepsilon} \begin{pmatrix} f(\mathbf{p} + \varepsilon_x) - f(\mathbf{p} - \varepsilon_x) \\ f(\mathbf{p} + \varepsilon_y) - f(\mathbf{p} - \varepsilon_y) \end{pmatrix}$$

Cette formule bien qu'utilisable dans tous les cas de figure nécessite l'évaluation de la fonction  $f$  à quatre reprises. Afin d'éviter cette lourdeur, nous avons aussi implémenté un calcul

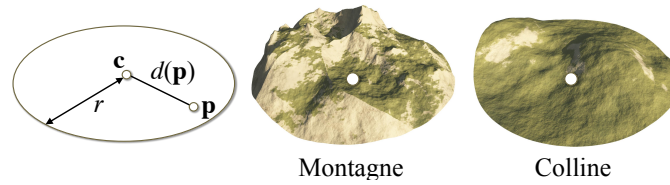


FIGURE 14 – Les primitives de type disque permettent de représenter aussi bien des collines que des montagnes un peu plus accidentées grâce l’utilisation de fonctions de bruits plus élaborées comme le *ridged multifractal* [MKM89].

analytique du gradient pour chaque nœud et primitive. Bien souvent, une partie des calculs de valeur de  $f$  et de son gradient est commune, donc nous avons aussi implémenté l’optimisation de ces calculs lorsqu’ils sont demandés de manière simultanée.

### 6.3 PRIMITIVES

Les primitives sont les feuilles de l’arbre de construction du modèle. Ils représentent une unité de terrain que nous appelons modelé. Nous avons développé deux types de primitives, celles qui s’appuient sur un squelette et celles qui utilisent des images. Les premières ont une empreinte mémoire très faible et sont évaluables rapidement. Les secondes ont en revanche une empreinte mémoire plus élevée, mais permettent aux artistes de réutiliser des modelés du monde réel ou des terrains édités dans d’autres logiciels.

#### 6.3.1 Primitives à squelette

Les primitives à squelettes sont inspirées du modèle de Blob-Tree [WGG99]. Elles sont définies par un squelette géométrique (un point, un segment, une courbe ou un contour) et un ensemble de paramètres qui définissent la fonction de poids et l’élévation en fonction de la distance à ce squelette.

**PRIMITIVE DISQUE.** Cette primitive est l’une des plus simples, elle représente une portion circulaire de terrain. Elle est décrite par son centre  $c \in \mathbb{R}^3$  et son rayon. Afin de contrôler le bruit que l’on ajoute à l’altitude de ce disque, nous introduisons un ensemble d’échelles de fréquences  $\{s_i\}$  associées à un ensemble d’amplitudes  $\{a_i\}$  qui vont permettre de combiner un bruit de base noté  $\eta : \mathbb{R}^2 \rightarrow \mathbb{R}$ . La fonction d’élévation sera ainsi :

$$f(\mathbf{p}) = c_z + \sum_{i=0}^{n-1} a_i \eta(s_i (\mathbf{p} - c_{xy}))$$

où  $c_z$  représente la composante  $z$  du point  $c$  et  $c_{xy}$  les composantes  $x$  et  $y$  de  $c$  (voir figure 14).

**PRIMITIVE COURBE.** Cette primitive est décrite par une courbe tridimensionnelle  $\Gamma$  à laquelle sont associés plusieurs profils  $\{\pi_i\}$ . Un profil est placé le long de la courbe à une abscisse curviligne donnée de telle sorte que l’on peut interpoler linéairement l’ensemble des profils en chaque point de la courbe. Cette interpolation sera notée  $\pi : \mathbb{R} \rightarrow \mathbb{R}$ . Pour

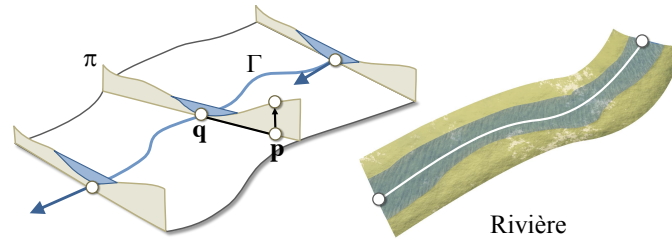


FIGURE 15 – Les primitives de type courbe permettent de décrire facilement des modelés qui sont structurés autour de trajectoires comme les routes ou les rivières.

évaluer la fonction d'élévation associée au point  $\mathbf{p}$ , nous commençons par calculer la projection de  $\mathbf{p}$  sur  $\Gamma$ . Cette projection est notée  $\mathbf{q}$ . La fonction d'élévation est définie comme suit :

$$f(\mathbf{p}) = \mathbf{q}_z + \pi \circ d_{\Gamma}(\mathbf{p}_{xy})$$

où  $d_{\Gamma}$  représente la distance signée d'un point jusqu'à la courbe  $\Gamma$ . Afin de garantir des temps d'évaluation assez courts, il est primordial d'utiliser des courbes suffisamment simples. Par exemple, on favorisera l'utilisation de courbes de degré 2 plutôt que 3.

**PRIMITIVE CONTOUR.** Toujours sur le même principe, il est possible d'interpoler un profil mais cette fois ci de manière angulaire autour d'un point. Soit  $c$  le centre de la primitive et  $v$  la direction de référence. Nous définissons un ensemble de profils  $\{\pi_i\}$  associés à des angles. Pour un point donné  $\mathbf{p}$ , nous calculons la distance de ce point jusqu'à  $c$  notée  $d(\mathbf{p})$ . Soit  $\pi$  l'interpolation angulaire des profils suivant la direction de  $\mathbf{p} - \mathbf{c}$  par rapport à  $v$ . La valeur d'élévation sera :

$$f(\mathbf{p}) = \mathbf{c}_z + \pi \circ d(\mathbf{p})$$

**FONCTIONS DE POIDS.** Sur toutes les primitives à squelettes, nous utilisons une fonction de poids qui décroît en fonction de la distance au squelette. Cette fonction est de classe  $C^1$  afin de garantir que les opérateurs vont fonctionner sans créer de discontinuités dans la fonction d'élévation.

Soit  $d(\mathbf{p})$  la distance du point  $\mathbf{p}$  au squelette. Nous construisons la fonction de poids comme la composition de  $d$  avec une fonction décroissante  $g$  :

$$\alpha(\mathbf{p}) = g(d(\mathbf{p})/R)$$

où  $R$  est le rayon d'influence. Dans notre implémentation nous utilisons la fonction de Wyvill [WGG99] définie comme suit :

$$g(x) = \begin{cases} (1 - x^2)^3 & \text{si } x < 1, \\ 0 & \text{sinon.} \end{cases}$$

Cette fonction est de classe  $C^2$ , ce qui permet d'assurer la dérivabilité du gradient.

### 6.3.2 Primitives images

Certains modelés de terrains sont difficiles à représenter de manière procédurale ou fonctionnelle. Afin de faciliter l'utilisation de terrains issus d'autres processus d'éditations ou de

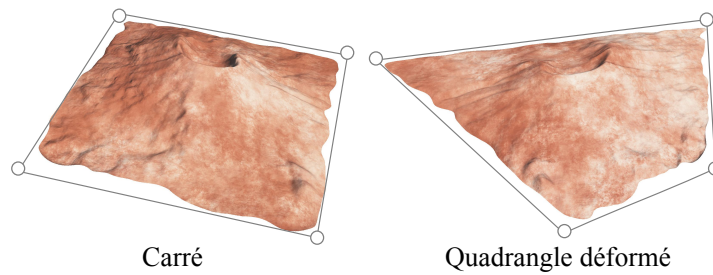


FIGURE 16 – Les primitives images permettent d'utiliser des données issues de l'extérieur, ici le mont St Helens, USA.

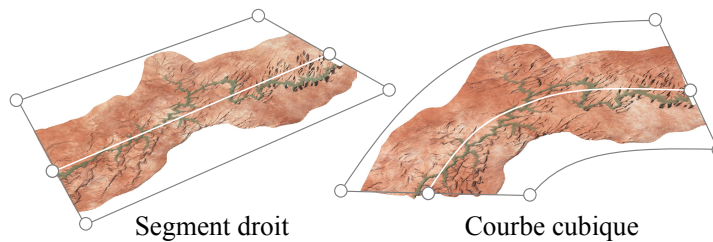


FIGURE 17 – Une carte hauteur du grand Canyon a été utilisée et déformée le long d'une courbe cubique.

données réelles, nous offrons la possibilité d'inclure une image de hauteur. Cette opération se fait grâce à des interpolations des données discrètes avec une classe de continuité suffisante (bi-cubique). En ce qui concerne les fonctions de poids, nous utilisons des formules similaires à celles des primitives à squelettes ou des cartes de poids interpolées.

Dans notre implémentation nous proposons deux types de primitives à images : quadrangulaire et courbes. Les figures 16 et 17 illustrent l'utilisation de ces primitives.

## 6.4 OPÉRATEURS

Les opérateurs sont au cœur de la structure hiérarchique du modèle. En effet, c'est grâce à eux que nous pouvons combiner plusieurs modèles de terrains tout en conservant une classe de continuité. Dans la description qui suit, nous considérons toujours des nœuds binaires mais dans la plupart des cas ces opérateurs sont extensibles à des opérandes n-aires. Chaque opérateur prend en argument deux sous-arbres  $A$  et  $B$  ayant chacun leur fonction d'altitude  $f_A$  et  $f_B$  ainsi que leur fonction de poids  $\alpha_A$  et  $\alpha_B$ .

### 6.4.1 Mélange

Cet opérateur permet de mélanger les altitudes par une combinaison barycentrique à partir des fonctions de poids (voir figure 18) :

$$f = \frac{\alpha_A f_A + \alpha_B f_B}{\alpha_A + \alpha_B}$$

La fonction de poids résultante est elle la somme des deux fonctions de poids :

$$\alpha = \alpha_A + \alpha_B$$



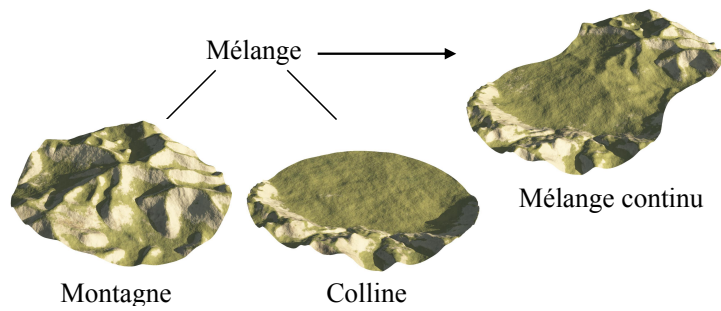


FIGURE 18 – Exemple d'utilisation de l'opérateur de mélange.

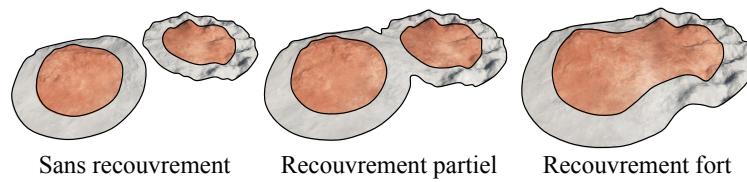


FIGURE 19 – L'effet de mélange est visible lorsque les domaines se recouvrent.

Au niveau des domaines, il en résulte que  $\Omega^0 = \Omega_A^0 \cup \Omega_B^0$ . Cette assertion est par contre fautive pour  $\Omega^T$ . En effet, comme les fonctions de poids s'ajoutent, on peut retrouver des parties de  $\Omega_A^0$  et  $\Omega_B^0$  dans  $\Omega^T$ . Ceci est illustré dans la figure 19.

Cet opérateur est commutatif et associatif. Il peut être généralisé à un opérateur n-aire. Les opérations peuvent alors être groupées afin d'être calculées de manière parallèle et l'ordre n'a pas d'importance sur le résultat.

#### 6.4.2 Remplacement

Cet opérateur permet de remplacer une partie d'un terrain par une autre. Il est donc forcément asymétrique. Le calcul du mélange ne tient compte que du poids de l'opérande de droite :

$$f = (1 - \alpha_B)f_A + \alpha_B f_B$$

La fonction de poids est égale à celle de la première primitive :

$$\alpha = \alpha_A$$

Cela correspond au fait que l'opérande de gauche est le terrain initial, et on veut conserver sa fonction de poids. Une autre option consiste à mélanger de manière continue les deux poids en utilisant comme poids celui de l'opérande de droite. Cela revient à appliquer la même formule à  $\alpha$  qu'à  $f$  :

$$\alpha = (1 - \alpha_B)\alpha_A + \alpha_B^2$$

L'opérateur de remplacement permet d'introduire une sorte de temporalité dans les actions de l'utilisateur. Par exemple, lorsque l'on remplace un terrain localement par un cratère puis le résultat par un autre cratère, l'ordre dans lequel vont être effectuées les opérations va imiter une séquentialité dans les événements (voir figure 20).

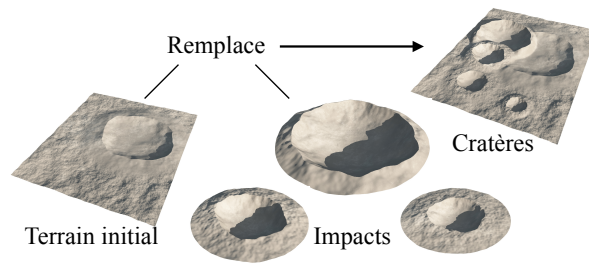


FIGURE 20 – L’opérateur de remplacement permet de produire un effet d’enchaînements successifs.

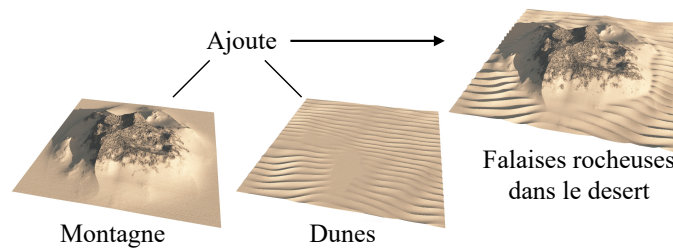


FIGURE 21 – L’opérateur d’ajout permet d’ajouter de considérer l’opérande de droite comme quelque chose de relatif.

### 6.4.3 Ajout

Contrairement aux opérateurs précédents, cet opérateur nécessite que l’opérande de droite exprime une quantité relative qui va pouvoir être ajoutée par dessus un terrain (ou supprimée si cette quantité est négative). Le mélange n’est donc plus barycentrique (voir figure 21) :

$$f = f_A + \alpha_B f_B$$

Comme pour l’opérateur précédent, on veut pouvoir conserver le poids de l’opérande de gauche :

$$\alpha = \alpha_A$$

### 6.4.4 Déformation

L’opérateur de déformation est un peu différent des précédents, il est unaire et va venir déformer le domaine de la fonction plutôt que la fonction d’altitude directement. Notons  $\omega$  une fonction bijective de classe  $C^1$  opérant dans  $\mathbb{R}^2$ . L’opérateur de déformation est défini comme ceci :

$$f = f_A \circ \omega^{-1}$$

et de manière identique pour la fonction de poids :

$$\alpha = \alpha_A \circ \omega^{-1}$$

Cet opérateur va donc modifier aussi les domaines  $\Omega^T$  ainsi que  $\Omega^0$  (voir figure 22).

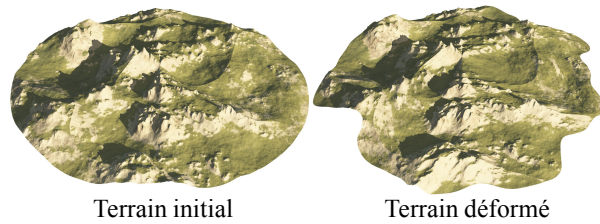


FIGURE 22 – L’opérateur de déformation permet de déformer une primitive ou un terrain plus complexe en opérant directement sur le domaine. Cela permet de casser la répétitivité due à l’utilisation d’instanciation.

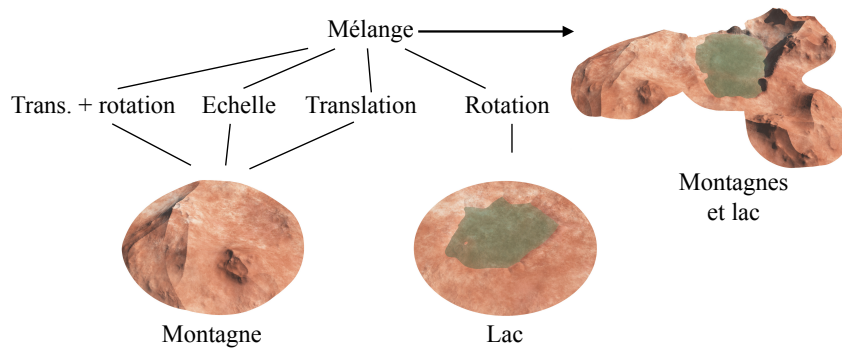


FIGURE 23 – L’instanciation associée à l’utilisation de transformations affines permet de réutiliser des portions de terrains à plusieurs endroits différents et à différentes échelles.

Afin de calculer le gradient de la fonction  $f$  après qu’on lui a appliqué une déformation il convient d’utiliser la formule suivante :

$$\nabla f(\mathbf{p}) = \nabla f_A \circ \omega^{-1}(\mathbf{p}) \times J_{\omega^{-1}}(\mathbf{p})$$

Le Jacobien  $J_{\omega^{-1}}$  pourra être précalculé de manière analytique pour chaque fonction de déformation  $\omega$  afin d’accélérer les calculs.

Les transformations affines sont des opérateurs de déformation particuliers dont la fonction de déformation sera une fonction affine. Pour la transformation de primitives simples, c’est équivalent à transformer directement les coordonnées du squelette. Malgré tout, cela permet d’unifier le mécanisme de transformation et de réutiliser des morceaux d’arbres de construction à plusieurs endroits différents. L’arbre devient un graphe acyclique orienté (voir figure 23) et permet une représentation et un stockage compact de grands terrains grâce à la réutilisation et l’instanciation de modèles de référence. Cette propriété extrêmement intéressante sera réutilisée dans le chapitre 7 pour la modélisation parcimonieuse de terrains.

## 6.5 NIVEAUX DE DÉTAIL

La complexité d’évaluation du modèle en un point dépend directement de la complexité de l’arbre et de la complexité de chacune des primitives utilisées. Nous introduisons deux mécanismes qui permettent d’optimiser d’une part la complexité de l’arbre et d’autre part le calcul d’une primitive individuelle en fonction du niveau de détail souhaité. Soit  $\kappa$  :

$\mathbb{R}^2 \rightarrow [0, 1]$  une fonction de classe  $C^1$  indiquant le niveau de détail en tout point. Un réglage standard de cette fonction de niveau de détail consiste à la faire diminuer quand la distance à la caméra augmente.

### 6.5.1 Opérateur de niveau de détail

Un opérateur de niveau de détail est un opérateur binaire qui va faire la transition douce de l'opérande  $A$  vers l'opérande  $B$  en fonction du niveau de détail. Pour cela, nous introduisons deux seuils  $k_A$  et  $k_B$  entre lesquels la transition se fera. En dehors de cette plage de niveau de détail, ce sera l'une des deux opérandes seulement qui sera effective :

$$f(\mathbf{p}) = \begin{cases} f_A(\mathbf{p}) & \text{si } \kappa(\mathbf{p}) \leq k_A, \\ (1-t)f_A(\mathbf{p}) + tf_B(\mathbf{p}) & \text{si } k_A < \kappa(\mathbf{p}) \leq k_B, \\ f_B(\mathbf{p}) & \text{sinon.} \end{cases}$$

avec

$$t = \frac{\kappa(\mathbf{p}) - k_A}{k_B - k_A}.$$

Le poids est calculé de la même manière.

### 6.5.2 Primitive de niveau de détail

Une primitive de niveau de détail est capable de calculer une fonction dont la complexité d'évaluation dépend directement du niveau de détail. De nombreuses possibilités sont offertes afin de mettre en oeuvre ce mécanisme. Nous allons l'illustrer sur le cas d'une primitive de forme circulaire dont le nombre de bandes de fréquences de bruit va augmenter lorsque le niveau de détail augmente :

$$f(\mathbf{p}) = \mathbf{c}_z + \sum_{i=0}^{n-1} a_i \eta((\mathbf{p} - \mathbf{c}) s_i) b_i \circ \kappa(\mathbf{p}).$$

Les fonctions de filtre  $b_i : [0, 1] \rightarrow [0, 1]$  sont uniformément croissantes et de classe  $C^2$  sur les intervalles  $[k_i^-, k_i^+]$  de niveau de détail  $\kappa(\mathbf{p})$  :

$$b_i \circ \kappa(\mathbf{p}) = \begin{cases} 0 & \text{si } \kappa \leq k_i^-, \\ (1-t^2)^3 & \text{si } k_i^- < \kappa \leq k_i^+, \\ 1 & \text{sinon.} \end{cases}$$

avec

$$t = \frac{\kappa(\mathbf{p}) - k_i^-}{k_i^+ - k_i^-}.$$

Un contrôle peut être effectué sur les différents seuils d'activation de filtres (voir figure 24).

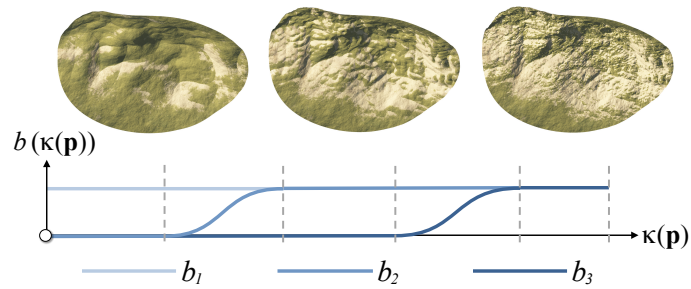


FIGURE 24 – Exemple de primitive de niveau de détail. Quand le niveau de détail  $\kappa$  augmente, les fonctions filtres viennent activer et faire apparaître des bandes de fréquences plus élevées.

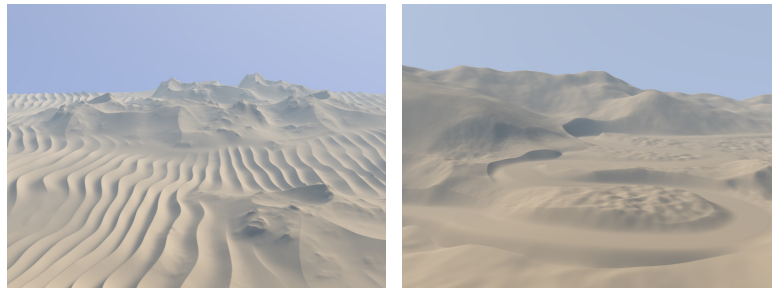


FIGURE 25 – Captures d'écrans de l'éditeur interactif de terrain, dans lequel la visualisation se fait grâce à une technique de *sphere tracing*.

## 6.6 VISUALISATION

Afin de visualiser le modèle que nous venons de décrire, nous avons plusieurs possibilités. La plus simple consiste à évaluer la fonction d'élévation en chaque point d'une grille régulière. Ensuite, comme pour une carte de hauteurs, nous construisons un maillage de cette grille de points que nous pouvons exploiter pour un affichage direct ou l'utilisation dans un logiciel de rendu. Une deuxième option consiste à faire du lancer de rayon directement sur le modèle, grâce à une technique de *sphere tracing* améliorée qui utilise les bonnes propriétés des fonctions que nous construisons. En effet, la fonction d'élévation  $f$  construite par notre modèle n'est pas seulement continue, elle est aussi Lipschitzienne, c'est-à-dire que sa variation locale est bornée par un coefficient. Nous ne détaillerons pas dans ce document le mode de calcul de la borne de Lipschitz des primitives et opérateurs, ni la technique de *sphere tracing* que nous utilisons pour en tenir compte.

Le point commun de ces deux techniques de visualisation est qu'elles reposent sur un grand nombre d'évaluation de la fonction d'altitude  $f$ . C'est pourquoi nous avons implémenté une version GPU de cette évaluation qui permet d'obtenir des temps d'affichage très courts, voire même interactifs pour le *sphere tracing*. La figure 25 donne un exemple de visualisation de notre modèle dans un éditeur interactif. Les détails de l'algorithme de *sphere tracing* peuvent être trouvés dans [GGP<sup>+</sup>15] et dans l'article qui lui est dédié [Har96].

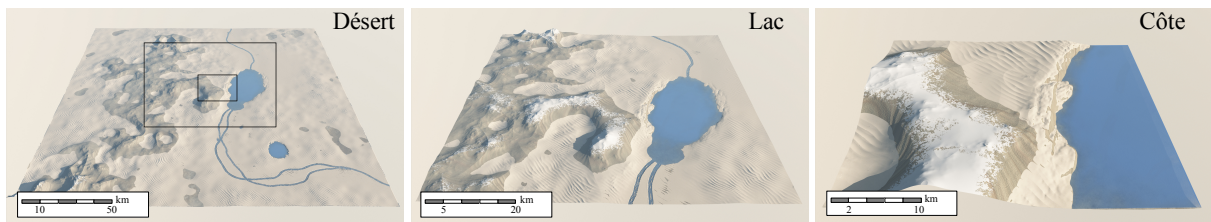


FIGURE 26 – Le modèle permet de représenter de vastes terrains de plus de 100 *km* avec un niveau de détails élevé.

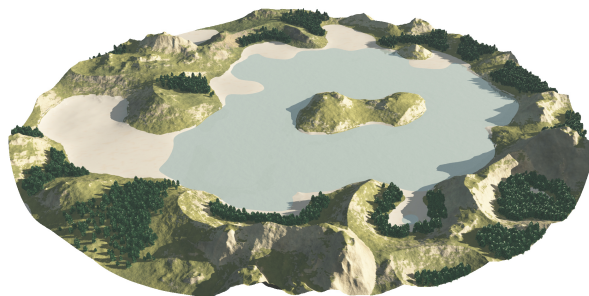


FIGURE 27 – L'utilisation de plusieurs arbres de constructions permet de combiner plusieurs matériaux, ici la roche, le sable, l'eau et la végétation.

## 6.7 RÉSULTATS

Le modèle que nous proposons permet de représenter une grande variété de terrains. Ces terrains ont des caractéristiques à grande échelle comme des montagnes ou des plaines tout en conservant une possibilité de contrôle très local comme une rivière ou un lac. La structure de donnée qui permet de stocker l'arbre est très légère, avec seulement une centaine de primitives par kilomètre carré (figure 26).

Il est possible de définir plusieurs arbres de construction associés à différents matériaux, comme le sable, la roche, l'eau et même la végétation. La figure 27 montre un exemple d'une scène qui a été modélisée ainsi.

L'évaluation de la fonction d'élévation peut se faire de manière rapide sur GPU grâce à diverses optimisations. Le caractère local de chaque primitive permet en effet d'élaguer naturellement des branches complètes de l'arbre. Le tableau 3 donne les statistiques sur les temps d'exécution ainsi que le stockage nécessaire et l'aire des scènes.

Toutes les scènes ont été créées durant une session allant de 15 à 55 minutes. La plupart de ce temps a été utilisé à placer des détails. Le coût mémoire de la primitive image utilisée dans la figure Figure 26 est d'environ 61 *kB*.

## 6.8 CONCLUSION

Nous avons proposé un modèle qui combine plusieurs atouts :

- Une description continue permettant d'avoir un degré de précision infini ;

Scene	Aire	Sphere tracing		Maillage		Stockage		
		Appels à $f(\mathbf{p})$	Temps	Appels à $f(\mathbf{p})$	Temps	$\mathcal{P}$	$\mathcal{O}$	$\mathcal{M}$
Figure 27	$2 \times 2$	66M	280	20M	80	78	10	12
Figure 26	$100 \times 100$	79M	561	80M	600	50	25	7

TABLE 3 – Statistiques pour deux scènes. Le nombre d’appels à  $f(\mathbf{p})$  correspond à une évaluation sur une grille.  $\mathcal{O}$  représente le nombre de nœuds de type opérateurs,  $\mathcal{P}$  le nombre de nœuds de type primitives et  $\mathcal{M}$  la mémoire utilisée pour représenter le modèle en  $kB$ . Les temps de calculs sont mesurés en  $ms$  et l’aire de la surface en  $km^2$ . L’algorithme de *sphere tracing* a été utilisé à une résolution de  $1920 \times 1080$ .

- Une approche constructiviste qui offre de la flexibilité par rapport à un processus de production ;
- Un contrôle fin grâce à la localisation des primitives de terrains ;
- Un mécanisme de visualisation en temps-réel par *sphere tracing*.

Ce modèle est très utile pour ce qui est de l’édition et de *l’authoring* de terrains. Ce ne sont pas les seuls traitements associés aux terrains, nous devons pouvoir représenter un terrain existant de manière optimale. Le modèle proposé permet de faire cette opération mais au prix d’un stockage lourd puisque le terrain est entièrement représenté sous la forme d’une grille et ensuite interpolé. Parmi les autres traitements classiques nous comptons aussi l’ajout de détails réalistes sur un terrain existant, ou encore la modélisation inverse procédurale. Dans le chapitre suivant, nous allons voir comment répondre à toutes ces attentes grâce à l’introduction de primitives issues d’un dictionnaire et au modèle parcimonieux.

## MODÈLE DE TERRAIN PARCIMONIEUX

---

Notre modèle parcimonieux est la prolongation du modèle fonctionnel présenté dans le chapitre précédent. L'idée originale consiste à utiliser la modélisation parcimonieuse afin de décrire localement un bout de terrain. Un bout de terrain va donc être la combinaison linéaire d'atomes d'un dictionnaire. Cette formulation générale trouve un grand nombre d'applications, suivant que l'on choisit les atomes du dictionnaire analytiques ou provenant de terrains réels.

Ce travail a fait l'objet d'une publication à la conférence Eurographics 2016 [GDGP16].

### 7.1 PRINCIPE

La modélisation parcimonieuse a été utilisée dans divers domaines tels que le débruitage [EAo6] ou la synthèse de texture [Peyog]. Dans le domaine de la géométrie, elle a fait l'objet de recherches sur la reconstruction de surfaces [XZZ<sup>+</sup>14] ou la compression de surfaces [DCV14]. Elle peut être aussi utilisée pour la représentation locale de surfaces et la paramétrisation [ZDL<sup>+</sup>14, JBPS11].

Le modèle de terrain que nous proposons repose sur les mêmes outils et méthodes que ceux utilisés en traitement du signal, mais avec une adaptation liée aux terrains, avec des applications inédites en modélisation de terrains, comme la modélisation procédurale inverse, ou l'amplification de terrains. Nous avons introduit le terme *amplification* dans la publication associée [GDGP16]. Cela désigne l'augmentation de la résolution d'un terrain donné par l'ajout de détails. En traitement d'image, on parle souvent de super-résolution.

### 7.2 MODÉLISATION PARCIMONIEUSE DE TERRAINS

Dans cette section, nous introduisons un nouveau modèle de terrain que nous appelons *Arbre de Construction Parcimonieux*. Ce modèle est une extension du modèle fonctionnel présenté dans le chapitre précédent. Cet arbre de construction possède trois niveaux (voir figure 28). Le niveau supérieur représente la totalité du terrain. Grâce à un opérateur de mélange  $n$ -aire, il combine une série de primitives qui représentent chacune une portion du terrain (étage du milieu). Chacune de ces primitives combine à son tour des atomes d'un dictionnaire (étage inférieur) de manière linéaire. Le caractère parcimonieux du modèle vient du fait que le nombre d'atomes utilisés par une primitive est faible par rapport au nombre total d'atomes présents dans le dictionnaire.

Soit  $\Omega \subset \mathbf{R}^2$  un compact de  $\mathbf{R}^2$ , un terrain est formellement défini comme une fonction d'élévation  $f$  :

$$f(\mathbf{p}) = \sum_{i=0}^{n-1} f_i(\mathbf{p})\alpha_i(\mathbf{p}) / \sum_{i=0}^{n-1} \alpha_i(\mathbf{p}) \quad \mathbf{p} \in \Omega$$

où les  $f_i$  sont les  $n$  primitives et les  $\alpha_i$  les poids associés.



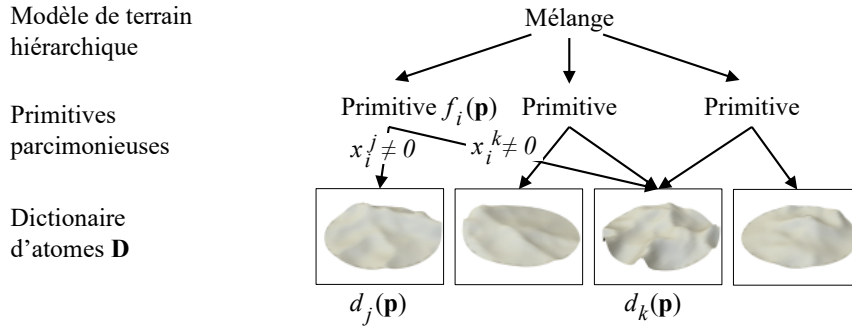


FIGURE 28 – Modèle de terrain hiérarchique à trois couches

Chaque primitive est donc caractérisée par une fonction d'élévation  $f_i$  et sa fonction de poids  $\alpha_i$  associée.  $f_i$  est définie sur un compact  $\Omega_i$  comme une combinaison linéaire d'atomes. Dans le système que nous avons implémenté, la  $i^{\text{ème}}$  primitive est définie sur une portion de terrain en forme de disque paramétré par son centre  $\mathbf{c}_i \in \mathbf{R}^2$ , l'élévation à son centre  $z_i$  et un vecteur de coefficients noté  $\mathbf{X}_i = \{x_i^j\}$ , où  $j = 1, \dots, n$  (voir figure 28). Ces coefficients sont utilisés pour définir la fonction d'élévation  $f_i$  comme une combinaison linéaire d'atomes notés  $d_j$  :

$$f_i(\mathbf{p}) = z_i + \sum_{j=0}^{N-1} x_i^j d_j(\mathbf{p} - \mathbf{c}_i)$$

Les fonctions  $\alpha_i$  sont continues de classe  $C^1$ , décroissantes et radiales sur un disque support de rayon  $R$  noté  $\Omega_i = \mathcal{B}(\mathbf{c}_i, R)$ . Cela permet de limiter l'influence spatiale de la primitive. La fonction radiale est la composition d'une fonction distance par rapport au centre  $\mathbf{c}_i$  avec une fonction de transfert décroissante lisse  $g$  :

$$\alpha_i(\mathbf{p}) = g(\|\mathbf{p} - \mathbf{c}_i\|/R) \quad g(r) = \begin{cases} (1 - r^2)^3 & \text{si } r < 1 \\ 0 & \text{sinon} \end{cases}$$

Un dictionnaire est un ensemble de  $N$  atomes  $d_i$  qui définissent des portions de terrains. Ces atomes sont des vecteurs, ils sont définis dans un espace vectoriel de dimension  $K$ . Cette dimension représente le nombre de scalaires qu'il faut pour représenter la portion de terrain. Par exemple, une portion  $16 \times 16$  de terrain sera représentée comme un atome de dimension  $K = 256$ . Dans certaines applications, l'atome est issu d'un échantillonnage d'une fonction complexe (un bruit par exemple). Cela permet de pouvoir confronter un vrai morceau de terrain à une approximation d'une fonction de manière transparente et sera utilisé dans la modélisation procédurale inverse notamment. Le dictionnaire peut être représenté dans sa globalité par une matrice  $\mathbf{D}$  de taille  $K \times N$  où chaque atome  $d_i$  est représenté par une colonne. Nous considérerons que les atomes  $d_i$  sont normalisés. Le modèle parcimonieux n'impose aucune contrainte sur l'orthogonalité des atomes entre eux, c'est même ce qui en fait la force.

Un dictionnaire multi-résolution est un ensemble de deux dictionnaires  $(\mathcal{H}, \mathcal{L})$  ayant exactement le même nombre d'atomes. Le premier dictionnaire  $\mathcal{H}$  est la version haute résolution, tandis que le deuxième  $\mathcal{L}$  est la version basse résolution. Il existe une correspondance entre ces atomes de telle manière que le  $i^{\text{ème}}$  atome du premier dictionnaire est la version haute résolution du  $i^{\text{ème}}$  atome du deuxième. S'il existe une décomposition sur le dictionnaire

basse résolution  $\mathcal{L}$  alors il suffit donc d'utiliser exactement les mêmes coefficients mais avec  $\mathcal{H}$  pour reconstruire une version haute résolution. Ce principe sera utilisé pour la synthèse ou l'amplification de terrains.

### 7.2.1 Construction de la représentation parcimonieuse

Le but est de construire une représentation parcimonieuse d'un terrain donné  $\mathcal{T}$ . Notons que cette construction va engendrer une perte d'information. Nous commençons par extraire des tuiles, puis nous générons un dictionnaire. Rappelons-nous que notre fonction d'élévation  $f$  est définie comme la somme de primitives  $f_i$ . Nous devons donc décomposer le terrain en tuiles qui se chevauchent. Ces tuiles, une fois qu'elles sont toutes positionnées, doivent couvrir entièrement le terrain. Afin de simplifier le processus, nous utilisons des tuiles en forme de disque disposées sur une grille régulière. Cette distribution garantit que nulle part sur le domaine la fonction de poids sera nulle :

$$\forall \mathbf{p} \in \Omega, \sum_{i=0}^{n-1} \alpha_i(\mathbf{p}) = \sum_{i=0}^{n-1} \alpha(\mathbf{p} - \mathbf{c}_i) \neq 0$$

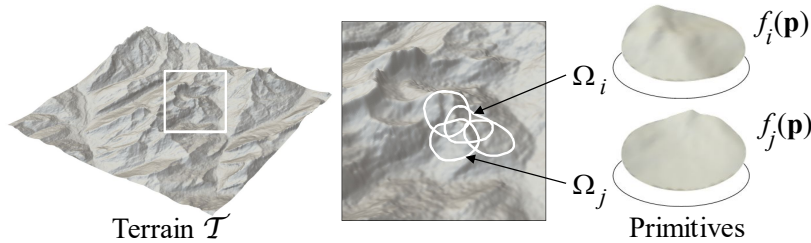


FIGURE 29 – Dans notre modèle, les disques se chevauchent et couvrent tout le domaine.

Chaque primitive ne doit utiliser qu'un petit sous-ensemble des atomes,  $\|\mathbf{X}_i\|_0$  représente le nombre de coefficients non nuls dans la décomposition, que l'on appelle aussi *sparsité* et que l'on notera  $s$ . Attention, cette appellation suggère que plus le nombre est élevé moins il y aura de coefficients alors que c'est l'inverse. Afin de construire la représentation parcimonieuse, il faut tout d'abord extraire les tuiles du terrain, et ensuite décrire chacune de ces tuiles grâce au dictionnaire et à la contrainte de sparsité  $s$ .

L'extraction de la tuile se fait tout d'abord par l'obtention de la composante constante sur le sous-domaine voulu  $\Omega_i$  centré en  $\mathbf{c}_i$  :

$$z_i = \int_{\Omega_i} \mathcal{T}(\mathbf{p}) dx dy / \int_{\Omega_i} dx dy$$

Cette composante est retirée de la tuile  $T_i$  afin de ne pas avoir à en tenir compte dans la décomposition :

$$T_i(\mathbf{p}) = \mathcal{T}(\mathbf{p} + \mathbf{c}_i) - z_i$$

### 7.2.2 Décomposition parcimonieuse

Le problème consiste à représenter dans le dictionnaire  $\mathbf{D}$  une tuile  $T_i$ , avec une contrainte de sparsité  $s$ . Chercher à minimiser une erreur de type  $\ell^2$  avec une contrainte de sparsité de

type  $\ell^0$  est un problème NP-difficile, nous allons donc utiliser une solution approchée pour résoudre le problème.

L'algorithme OMP (pour *Orthogonal Matching Pursuit*) introduit dans [MZ93] permet de résoudre le problème de manière itérative. Il consiste à trouver l'atome qui maximise la projection, retirer cette projection et trouver un deuxième atome qui minimise le résidu. Tant que le nombre d'atomes correspondant à la sparsité  $s$  n'est pas atteint, l'algorithme continue. Formellement, l'algorithme va stocker un ensemble d'indices  $\Gamma$  qui correspond aux atomes déjà utilisés. Cet ensemble au départ est vide et les étapes suivantes sont exécutées :

1. Sélectionner la coordonnée  $\hat{k}$  qui minimise l'erreur lorsqu'on l'ajoute à l'ensemble des atomes déjà présents dans  $\Gamma$  :

$$(\hat{k}, \hat{\beta}) = \operatorname{argmin}_{k \in \bar{\Gamma}, \beta \in \mathbf{R}^{|\Gamma|+1}} \|T_i - \mathbf{D}_{\Gamma \cup \{k\}} \beta\|_2^2.$$

2. Mettre à jour l'ensemble des indices  $\Gamma$  et la solution  $\mathbf{X}_i$  :  $\Gamma = \Gamma \cup \{\hat{k}\}$  ;  $\mathbf{X}_i^\Gamma = \hat{\beta}$  et  $\mathbf{X}_i^{\bar{\Gamma}} = 0$ , où  $\mathbf{X}_i^\Gamma$  (resp.  $\mathbf{X}_i^{\bar{\Gamma}}$ ) sont les coefficients du vecteur  $\mathbf{X}_i$  avec les indices dans  $\Gamma$  (resp.  $\bar{\Gamma}$ ).

On stoppe la boucle lorsque  $\Gamma$  a atteint la taille voulue, c'est-à-dire  $|\Gamma| = s$ .

Un exemple des itérations est décrit dans la figure 30. Plus le nombre d'atomes est important, meilleure sera la reconstruction. À la fin de cette étape d'OMP, chaque tuile peut être reconstruite de manière approchée.

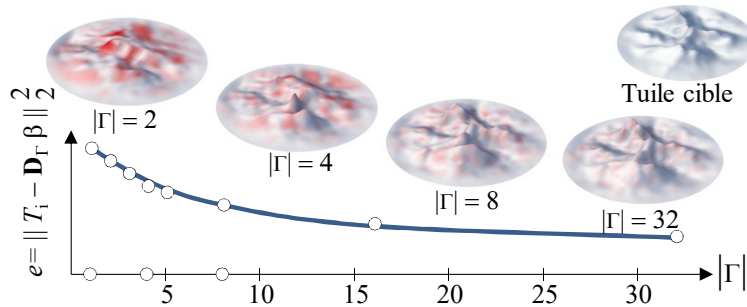


FIGURE 30 – Aperçu de l'algorithme OMP : l'erreur diminue au fur et à mesure que le nombre d'atomes dans  $\Gamma$  augmente.

Dans des conditions particulières où le dictionnaire utilisé est directement issu d'un terrain et que nous essayons de reconstruire ce même terrain avec une sparsité de 1, alors la reconstruction sera parfaite. En effet, chaque tuile trouvera un atome qui la reconstruit de manière parfaite.

### 7.2.3 Apprentissage de dictionnaire

Les conditions particulières qui permettent une reconstruction exacte dans le cas précédent ne tiennent pas compte de la possible redondance car  $N = n$ . Pour optimiser le dictionnaire en fonction des données à représenter donc en fonction du terrain lui-même, nous devons considérer  $N < n$  et  $s < N$ . Cela correspond à un dictionnaire plus petit et l'utilisation d'un nombre réduit d'atomes pour représenter chaque tuile de terrain.

Pour une lecture plus facile, nous utiliserons la notation suivante :  $\mathbf{d}_i = (d_{ik})$  et  $\mathbf{f}_i = (f_{ik})$  avec  $i = 1, \dots, N, k = 1, \dots, K$ . Le problème d'apprentissage de dictionnaire peut s'écrire sous la forme :

$$\begin{aligned} \min_{\mathbf{D} \in \mathbf{R}^{K,N}, \mathbf{X} \in \mathbf{R}^{N,n}} \|\mathbf{F} - \mathbf{D} \cdot \mathbf{X}\|_F^2 \\ \text{tel que } \forall i = 1, \dots, n, \|\mathbf{X}_i\|_0 \leq s \end{aligned} \quad (1)$$

$\mathbf{F}$  est la matrice  $K \times n$  dont les colonnes sont des primitives du terrain à représenter,  $\mathbf{X}$  est la matrice des coefficients de décomposition et  $\mathbf{X}_i$  est la  $i^{\text{ème}}$  colonne de  $\mathbf{X}$ , *i.e.* les coefficients de la décomposition de  $f_i$  sur  $\mathbf{D}$ . La contrainte assure que chaque  $f_i$  utilise au plus  $s$  atomes du dictionnaire pour sa décomposition.

Il existe plusieurs algorithmes qui permettent de résoudre le problème (1) de manière approchée. Parmi ces méthodes, nous avons choisi l'algorithme de K-SVD [AEB06]. À partir d'un dictionnaire initialisé avec des éléments choisis aléatoirement parmi les  $n$  primitives, l'algorithme alterne entre deux étapes jusqu'à une convergence d'erreur ou un nombre d'itérations atteint :

1. Le dictionnaire  $\mathbf{D}$  étant fixé, trouver la meilleure représentation  $\mathbf{X}$ .
2. La représentation  $\mathbf{X}$  étant fixée, trouver le meilleur dictionnaire  $\mathbf{D}$  tout en assurant la contrainte de sparsité.

La première étape peut être résolue grâce à l'algorithme OMP décrit précédemment. La deuxième étape cherche à optimiser le dictionnaire tout en assurant que chaque tuile est bien décrite par le même ensemble d'atomes. Pour ce faire,  $\mathbf{D}$  est mis à jour atome par atome à partir du dictionnaire de l'itération précédente. On étudie la suppression de chaque atome  $d_i \in \mathbf{D}$  en sélectionnant toutes les décompositions actuelles qui l'utilisent. Supprimer cet atome mène à une erreur sur cet ensemble de décompositions. On remplace alors l'atome grâce à une décomposition SVD (Single Value Decomposition) de l'erreur. Cela permet d'obtenir un bon substitut tout en maintenant la même sparsité (voir [AEB06] pour plus de détails).

Cette construction de dictionnaire est très efficace mais souffre d'un problème : elle détruit les atomes initiaux. Pour certaines applications, cela n'est pas souhaitable et nous utilisons une formalisation du problème différente :

$$\begin{aligned} \min_{\mathbf{D} \in \mathbf{R}^{K,N}, \mathbf{X} \in \mathbf{R}^{N,n}} \|\mathbf{F} - \mathbf{D} \cdot \mathbf{X}\|_F^2 \\ \text{tel que } \forall i \in \{1, \dots, n\}, \|\mathbf{X}_i\|_0 \leq s \\ \forall j \in \{1, \dots, N\}, \exists l \in \{1, \dots, n\}, d_j = f_l \end{aligned} \quad (2)$$

Pour résoudre ce problème, nous utilisons une approche similaire en deux étapes dont seule la deuxième diffère. Au lieu de trouver un nouvel atome par une décomposition SVD, nous cherchons de manière exhaustive un atome de substitution dans l'ensemble des primitives disponibles. Ce processus est plus coûteux car pour chaque nouvelle substitution, nous devons tester tous les atomes disponibles. Malgré tout, cela reste important dans les applications d'amplification où des vrais morceaux de terrains doivent pouvoir être visibles. Nous utilisons donc les deux approches selon le type d'application.

### 7.3 APPLICATIONS

Ce formalisme parcimonieux pour les terrains peut être utilisé dans de nombreuses applications. Nous en proposons quatre : la représentation compacte de terrains, la modélisation procédurale inverse, la synthèse et l'amplification de terrains et enfin un cadre général pour la création de terrains.

## 7.3.1 Représentation compacte de terrains

Le modèle parcimonieux permet de représenter un terrain  $\mathcal{T}$  de manière compacte notamment à l'aide de l'optimisation de dictionnaire. Au lieu de représenter le terrain lui-même, il suffit de stocker le dictionnaire  $\mathbf{D}$  (censé être petit) ainsi que les coefficients de décomposition  $\mathbf{X}$ . Dans notre implémentation, les centres des tuiles n'ont pas besoin d'être représentés car ils sont positionnés sur une grille régulière. De plus, le stockage des coefficients  $\mathbf{X}$  peut être optimisé car cette matrice est très largement creuse. Nous n'avons donc qu'à stocker une liste assez courte de paires  $(j, x_i^j)$ .

Pour un terrain de taille  $w \times h$ , la quantité d'information à stocker pour représenter ce terrain est égale à  $t_0 = w \times h = nK/4$  où  $n$  est le nombre de tuiles et  $K$  est la taille (en quantité d'information à stocker). Ce calcul est fait en considérant que les tuiles se chevauchent de moitié, c'est-à-dire que le pas de la grille est égal à la moitié de la taille de la tuile. Nous utilisons dans tous nos résultats ce paramétrage qui s'est avéré efficace et sans artefacts.

Pour une représentation avec un dictionnaire, il faut représenter d'une part le dictionnaire lui-même et d'autre part les coefficients. Le dictionnaire correspond au stockage de  $N$  tuiles donc  $N \times K$  coefficients. Les coefficients de la matrice creuse  $\mathbf{X}$  sont représentés par seulement  $n(2s + 1) + 1$  valeurs. Rappelons ici que nous ne stockons pas explicitement les centres des tuiles car leurs positions sont implicites. Il reste seulement l'élévation moyenne  $z_i$  de chaque tuile à stocker afin de pouvoir restituer le terrain. Au total, le nombre de valeurs à stocker est donc :

$$t(s, N) = KN + n(2s + 2) + 1$$

Il est important maintenant de se souvenir que la sparsité  $s$  (le nombre de coefficients  $x_i^j$  non nuls) est beaucoup plus petite que le nombre d'atomes  $N$  et que la taille de la tuile  $K$ , qui eux-mêmes sont beaucoup plus petits que le nombre de tuiles contenues dans le terrain  $n$ . On a donc  $s \ll N \ll n$ , et  $N = O(K)$  ce qui mène à une représentation compacte :

$$t(s, N) = KN + n(2s + 2) + 1 \ll nK/4$$

Notons  $\rho = t(s, N)/t_0$  le ratio de stockage entre le modèle parcimonieux et le modèle de terrain original.

Terrain	Taille $t_0$	Sparsité $s$	Ratio $\rho$	PSNR
Alpes (512 × 512)	0.26 M	1	14.9 %	23.7
		2	20.0 %	26.4
		8	50.9 %	31.1
Italie (3600 × 3600)	12.96 M	1	4.9 %	31.7
		2	9.7 %	34.2
		8	38.0 %	39.2

TABLE 4 – Stockage initial  $t_0$ , sparsité  $s$ , ratio de stockage  $\rho$  et PSNR (en dB) de deux terrains avec différentes valeurs de sparsité. Les calculs ont été effectués avec une taille de dictionnaire de 128 atomes.

Dans ce type d'application, nous voulons essayer de minimiser l'erreur de reconstruction, donc nous utilisons une taille de tuile assez faible de l'ordre de  $K \sim 10^2$ . Le tableau 4 reporte les ratio de stockage  $\rho$  et les erreurs de reconstruction pour des terrains encodés avec

différentes sparsités  $s$ . Les résultats montrent que de grands terrains de plusieurs centaines de kilomètres carrés à une résolution de quelques mètres peuvent être représentés efficacement par notre modèle parcimonieux en quelques méga-octets. En utilisant la méthode classique mais très efficace de compression JPEG2000 et un taux de compression identique sur le modèle de terrain *Italie*, notre méthode permet d'obtenir une PSNR de 31.7 dB là où JPEG2000 se contente de 20 dB.

### 7.3.2 Modélisation procédurale inverse

Nous proposons ici une application originale du modèle parcimonieux. Il s'agit de retrouver, à partir d'un terrain donné  $\mathcal{T}$ , une représentation sous forme d'arbre de construction parcimonieux à base de fonctions. Chaque primitive n'est donc plus une tuile qui contient une version échantillonnée d'un bout de terrain, mais une fenêtre de fonction, typiquement un bruit.

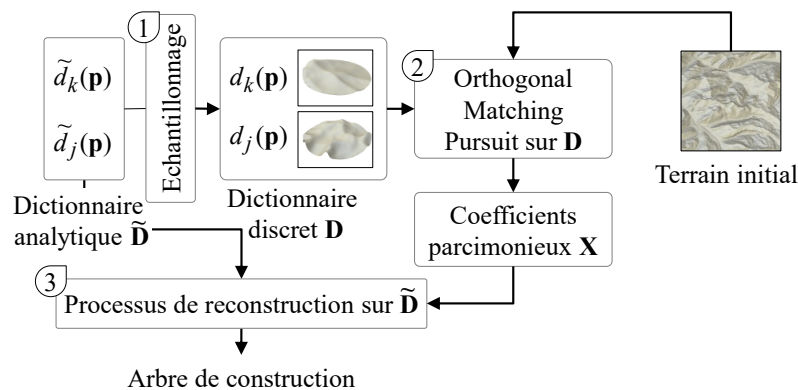


FIGURE 31 – Processus de création d'un modèle procédural à partir d'un terrain. Nous disposons d'un dictionnaire d'atomes analytiques  $\tilde{\mathbf{D}}$ , à partir desquels nous construisons une version échantillonnée  $\mathbf{D}$ . Grâce à une décomposition OMP sur  $\mathbf{D}$  et une reconstruction sur  $\tilde{\mathbf{D}}$ , nous obtenons une représentation analytique du terrain.

Imaginons que nous voulions représenter un terrain à partir d'un dictionnaire de fonctions issues d'un bruit. Si nous utilisons un dictionnaire construit par échantillonnage de ces fonctions de bruit, nous allons obtenir une représentation discrète et lourde en terme de stockage. C'est absurde car au départ, les fonctions étaient très légères à représenter (il faut seulement stocker leurs paramètres) et ont naturellement une représentation continue. L'idée de cette application est d'établir une correspondance entre des atomes analytiques d'un dictionnaire  $\tilde{\mathbf{D}}$  et leurs versions discrètes dans un autre dictionnaire  $\mathbf{D}$ , tout en gardant l'information d'association entre les deux. Cette construction est triviale puisqu'elle correspond juste à un échantillonnage des fonctions. Nous sommes ainsi capables de faire une représentation d'un terrain donné sur  $\mathbf{D}$  grâce à l'algorithme d'OMP, puis de remplacer les atomes par ceux de  $\tilde{\mathbf{D}}$  (voir figure 31). L'OMP permet de trouver les coefficients de décomposition  $\mathbf{X}$  et de les utiliser pour une reconstruction analytique :

$$\tilde{\mathbf{F}} = \tilde{\mathbf{D}} \cdot \mathbf{X}$$

Une des limitations se trouve dans la taille du dictionnaire qui peut devenir très grande si on veut garantir une bonne diversité dans les atomes. Nous pouvons utiliser la résolution du problème (2) afin de réduire efficacement la taille du dictionnaire tout en garantissant une bonne expressivité.

### 7.3.3 Amplification et synthèse de terrains

Étant donné un terrain basse résolution  $\mathcal{T}$  et un exemple haute résolution  $\mathcal{E}$ , la synthèse et l'amplification de terrain vont permettre de produire un nouveau terrain détaillé qui préserve les caractéristiques du terrain  $\mathcal{T}$ . La résolution du terrain initial va être augmentée grâce à l'insertion de détails trouvés dans l'exemple.

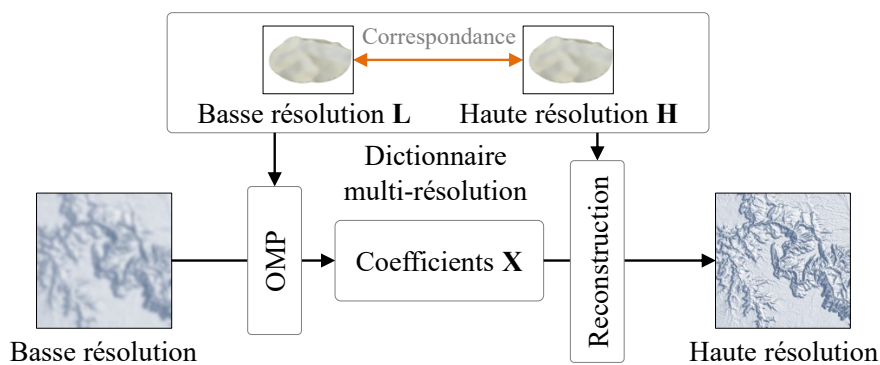


FIGURE 32 – Aperçu de la méthode d'amplification et synthèse de terrains. À partir d'un terrain basse résolution ou d'une esquisse de l'utilisateur, nous utilisons un dictionnaire multi-résolution qui décompose sur les atomes basse résolution et reconstruit avec leur version haute résolution.

Dans ce cas où le terrain basse résolution  $\mathcal{T}$  est une esquisse nous utiliserons le terme *synthèse de terrain*, alors que dans le cas où nous utilisons des données réelles, nous parlerons d'*amplification de terrains*. Dans ce dernier cas, les données d'entrée (le terrain  $\mathcal{T}$  et l'exemple  $\mathcal{E}$ ) devront être cohérentes en termes de résolution et type de terrains. La synthèse de terrain permet de créer des terrains à partir d'esquisses grossières, alors que l'amplification permet de visualiser des terrains réels à une résolution meilleure que l'originale de manière cohérente.

L'algorithme se décompose en trois étapes :

1. Un dictionnaire multi-résolution  $(\mathcal{H}, \mathcal{L})$  est construit à partir de l'exemple  $\mathcal{E}$ .
2. Le terrain d'entrée  $\mathcal{T}$  est décomposé sur la version basse résolution  $\mathcal{L}$  du dictionnaire, donnant une matrice de coefficients parcimonieux  $\mathbf{X}$ .
3. Un arbre de construction parcimonieux  $\mathcal{S}$  est généré par combinaison des coefficients  $\mathbf{X}$  et de la version haute résolution  $\mathcal{H}$  du dictionnaire.

Dans la première étape, un dictionnaire haute résolution est tout d'abord extrait du terrain exemple  $\mathcal{E}$ . Ensuite, chacun des atomes est sous-échantillonné afin de créer un dictionnaire basse résolution. La correspondance entre les atomes haute et basse résolution est conservée de telle sorte que  $\mathcal{H}_i$  est la version haute résolution de  $\mathcal{L}_i$  pour  $i \in 1, \dots, n$  (voir figure 33). C'est cette version basse résolution du dictionnaire qui va permettre de faire une décomposition du terrain d'entrée  $\mathcal{T}$  de la seconde étape de l'algorithme.

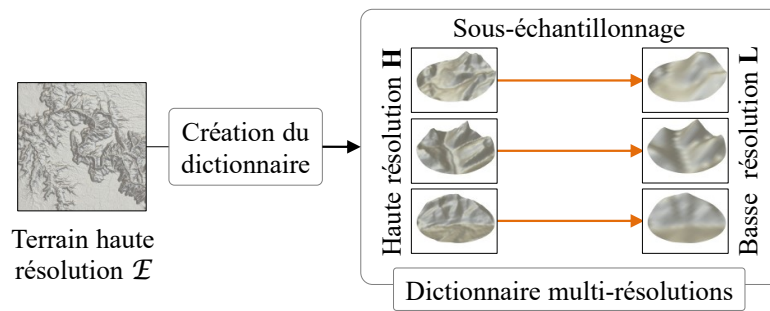


FIGURE 33 – Construction du dictionnaire multi-résolution.

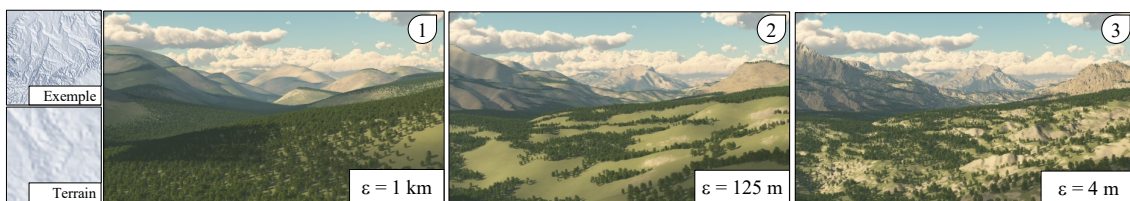


FIGURE 34 – Exemple d’amplification de terrain. Le terrain initial avait une résolution de 1 km. Après avoir amplifié à 4 reprises ce terrain d’un facteur 4, nous obtenons une résolution d’environ 4 m et des détails extrêmement précis.

Pour finir, nous substituons les atomes par leur version haute résolution dans la reconstruction et obtenons un terrain final à une haute résolution. Les résultats montrent que cette méthode permet de transférer automatiquement des détails depuis l’exemple vers le terrain (figure 34).

Le tableau 5 reporte une comparaison des coûts de stockage pour une version détaillée (hypothétique) du terrain avec le stockage du terrain basse résolution et de l’exemple haute-résolution. Ce tableau reporte aussi les temps obtenus pour la synthèse et l’amplification de terrains en utilisant un dictionnaire complet c’est-à-dire contenant toutes les tuiles de l’exemple ou un dictionnaire appris par optimisation du problème (2).

Les résultats montrent que la synthèse peut être faite de manière efficace puisque la sparsité est de  $s = 1$  et encore plus rapidement avec un dictionnaire optimisé. Dans le cas où nous utilisons un dictionnaire complet, il est plus efficace de stocker uniquement l’exemple qui par définition contient toutes les tuiles possibles, et donc mène à une empreinte mémoire moins marquée.

#### 7.3.4 Contrôle et création

Plusieurs moyens permettent de contrôler précisément le résultat obtenu lors de la synthèse de terrains. Le rayon de l’atome  $R$ , lorsqu’il est petit, implique un résultat qui colle mieux au terrain d’entrée car il sera plus aisé de trouver des atomes qui correspondent. Inversement, lorsqu’il devient grand, les atomes choisis seront de plus en plus éloignés du terrain d’entrée, et on reconnaîtra de plus en plus les tuiles de l’exemple. Ce paramètre est donc un bon moyen de positionner la synthèse pour qu’elle soit fidèle au terrain ou à



Appli.	Type dictionnaire	Taille initiale	Taille finale	Ratio $\rho$	$N$	Temps d'app.	Temps de synthèse
Synth.	Complet Appris	$64 \times 128$	$4096 \times 8192$	9.6 %	60 000	0.0	25.5
				0.48 %	64	5.2	10.2
Amp.	Complet Appris	$32 \times 32$	$8192 \times 8192$	19 %	120 000	0.0	84.0
				0.23 %	256	30.0	27.0

TABLE 5 – Statistiques pour la synthèse et l’amplification de terrains montrés dans les figures 34 et 36. La taille de stockage et les temps d’exécution (en secondes) sont reportés pour plusieurs tailles d’atomes. La sparsité est toujours fixée à 1 pour ces applications afin de produire une bonne reconstruction. Les comparaisons sont faites entre deux cas : utilisation d’un dictionnaire complet et dictionnaire appris.

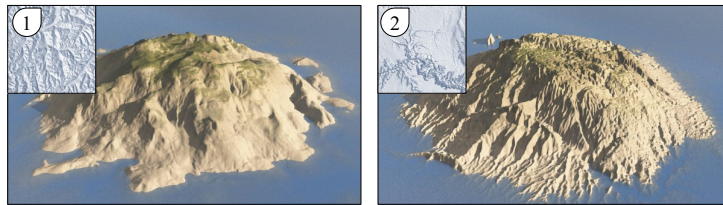


FIGURE 35 – En donnant deux terrains exemples différents, le résultat peut être radicalement différent, même si la forme générale du terrain en entrée est globalement respectée.

l’exemple. Notons ici que le rayon de la tuile est lié directement à ta taille de stockage de l’atome grâce à la relation  $K = \lceil (4R/\delta)^2 \rceil$  où  $\delta$  est le pas d’échantillonnage du terrain.

Il est assez aisé d’utiliser le résultat d’une synthèse ou d’une amplification dans un arbre de construction plus complexe comme le montre la figure 37.

#### 7.4 RÉSULTATS

Le modèle proposé permet de générer ou représenter des terrains vastes avec un haut niveau de détails, qui seraient impossibles à obtenir avec des approches standards. Le modèle est versatile, il permet de représenter des types très variés de terrain dans une forme très



FIGURE 36 – Ce canyon a été créé en utilisant une carte de hauteur peinte à la main à laquelle on a ajouté à 3 reprises des détails. La rivière au fond du canyon a été ajoutée en insérant une primitive de type courbe dans l’arbre de construction.

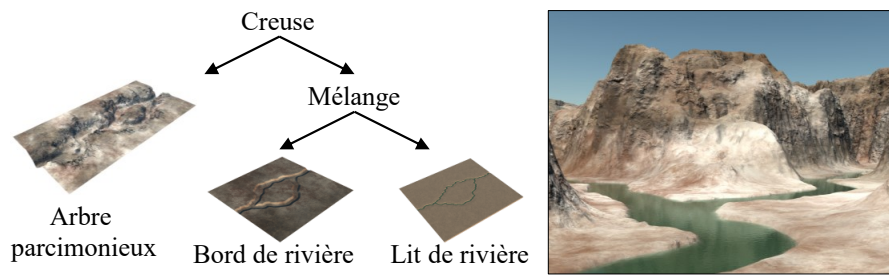


FIGURE 37 – Exemple de création d’un terrain complexe. Le résultat de la modélisation parcimonieuse a été combiné à des primitives de type courbe afin de modéliser les lits mineur et majeur de la rivière au fond du canyon.

compacte. Il offre la possibilité de reproduire des styles géomorphologiques et des caractéristiques de terrains variés dans un formalisme unifié.

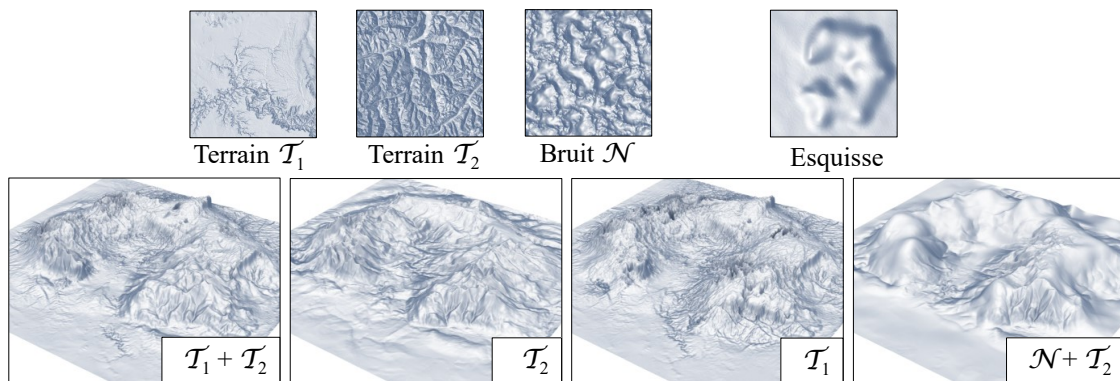


FIGURE 38 – En combinant des atomes issus de sources différentes, des terrains avec des caractéristiques différentes peuvent être obtenus.

Une des contributions principales de cette méthode est sa simplicité d’utilisation grâce à des dictionnaires dont les sources peuvent être variables (analytique, données réelles). Il est d’ailleurs possible de combiner des atomes venant de sources différentes comme le montre la figure 38. L’atome qui sera choisi pourra faire partie d’une source ou d’une autre, menant à un mélange subtil de caractéristiques de terrains. Cette approche a été généralisée dans l’article [AAC<sup>+</sup>17] afin d’avoir un contrôle plus précis sur les styles appliqués.

Notre méthode est capable de générer une représentation de terrains vastes à une précision très fine en quelques secondes. De par son aspect multi-résolution natif, notre méthode se prête bien à une visualisation dépendante du point de vue et utilisant des stratégies d’optimisation des ressources. Le stockage du modèle est très compact comme le montre le tableau 4. La plupart des opérations présentes dans les algorithmes présentés reposent sur de l’algèbre linéaire et bénéficient d’une implémentation parallèle native.

Lorsque le terrain a une résolution trop faible et donne un rendu lisse, la technique la plus répandue pour insérer des détails artificiels consiste à ajouter un bruit fractal au terrain. Cette technique, qui marche bien à petite échelle par exemple pour donner de la rugosité à un matériau, n’est pas adaptée aux terrains plus vastes. En comparaison, la méthode

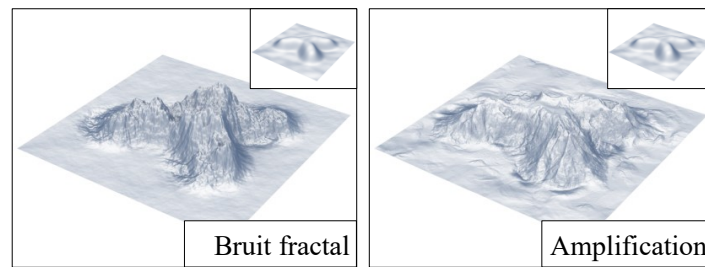


FIGURE 39 – Les techniques standards procédurales telles que l’ajout de bruit (gauche) donnent des résultats peu structurés, là où notre méthode permet de générer des caractéristiques cohérentes de terrain (droite).

parcimonieuse que nous proposons donne des résultats beaucoup plus réalistes comme illustré dans la figure 39.

La principale limitation est l’absence de garanties sur la cohérence géomorphologique du terrain, comme par exemple l’écoulement de l’eau cohérent. Si la version grossière du terrain ne permet pas d’écoulement cohérent, le résultat obtenu par modélisation parcimonieuse non plus. Le rayon  $R$  de la tuile aura sur ce critère un impact déterminant car plus il sera élevé, moins le résultat sera conforme au terrain grossier initial.

## 7.5 CONCLUSION

Nous avons introduit dans ce chapitre un modèle générique permettant de représenter, générer ou amplifier des terrains. Il combine favorablement le formalisme de modélisation parcimonieuse qui consiste à exprimer un signal grâce à une représentation adaptée au signal, et le domaine de la représentation numérique des terrains. Le modèle offre de nombreux avantages parmi lesquels sa compacité, sa facilité d’utilisation ainsi que sa capacité à être utilisé dans de multiples applications. Il permet d’ajouter ce qui manquait au modèle fonctionnel introduit précédemment en introduisant des contextes d’utilisations plus larges, en génération procédurale notamment.

## CONCLUSIONS ET PERSPECTIVES

---

### 8.1 SYNTHÈSE SCIENTIFIQUE

Ma thèse soutenue en 2002 a porté sur le problème inverse de la modélisation fractale, c'est-à-dire comment trouver une représentation fractale d'un objet réel (courbe, surface, image). Rétrospectivement, ce sujet est né de la rencontre entre mes directeurs de thèse, Atilla Baskurt et Eric Tosan. Le premier a plus une sensibilité d'analyseur d'image, tandis que le deuxième a tourné sa carrière vers la modélisation géométrique fractale. C'était donc bien la rencontre de deux mondes qui a donné naissance à ce sujet que l'on peut qualifier de problème inverse. La reconversion thématique enclenchée par la suite a eu pour effet de laisser de côté la dimension analyse des mes travaux de recherche.

J'ai rédigé ce manuscrit de telle sorte qu'il mette l'accent sur mes contributions en modélisation de terrains. J'ai présenté tour à tour un modèle fractal avec insertion de détails, un modèle de terrain obtenu par diffusion de contraintes, un modèle de terrain fonctionnel et constructiviste et pour finir un modèle de terrain parcimonieux. Parmi tous ces travaux, seul le dernier propose une dimension d'analyse. En effet, la description de terrain se fait par l'analyse et l'appariement de portions de terrains. L'efficacité de cette dernière contribution me conforte dans l'idée que l'analyse peut avoir une part importante dans la modélisation afin de mieux appréhender à la réalité.

Un terrain virtuel n'est pas un objet simple qui pourrait par exemple se résumer à une image en niveau de gris. J'espère l'avoir montré tout au long de ce document. Il existe de nombreux modèles permettant de représenter les terrains virtuels. Les défis fondamentaux que j'avais soulevés dans le chapitre 3 sont les suivants :

1. représenter des terrains vastes avec de nombreux détails ;
2. prendre en compte les matériaux ;
3. contrôler les modelés ;
4. être réaliste d'un point de vue géomorphologique.

Dans ce document j'ai montré des travaux qui permettent essentiellement de progresser sur les axes 1 et 3 (voir figure 40). L'utilisation d'algorithmes de subdivision (chapitre 4) ou de fonctions (chapitres 6 et 7) permet de représenter des étendues vastes de terrains grâce à une faible quantité d'informations. La masse de données générée est due au côté récursif des algorithmes de subdivision, ou à la propriété de continuité des fonctions employées. L'utilisation de détails (chapitre 4) ou de spatialisation de primitives (chapitres 5, 6 et 7) permet quant à elle d'avoir un contrôle précis des emplacements et de la forme des modelés utilisés. Dans une moindre mesure, le chapitre 6 présente un modèle qui est facilement exploitable dans un mode multi-matériaux mais qui n'a pas été exposé pour des raisons de concision.

Certains de mes travaux, qui ne sont pas détaillés dans le présent document, ont un impact sur les autres défis. C'est le cas par exemple des travaux sur la simulation combinée d'érosion et de végétation [CGG<sup>+</sup>17] qui permet d'obtenir un terrain amélioré contenant des matériaux comme l'humus, les rochers, et différentes sortes de végétations. Dans le même

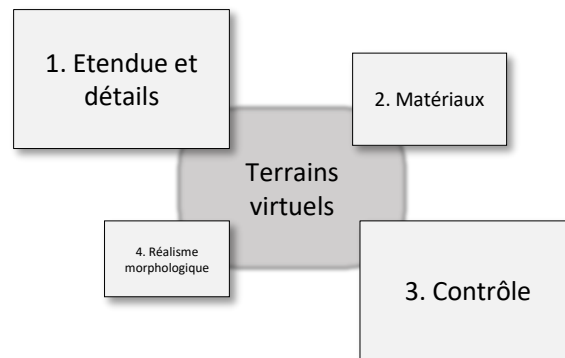


FIGURE 40 – Les quatre défis en modélisation de terrains virtuels et la part des contributions présentées dans ce document.

esprit, les travaux sur la modélisation parcimonieuse multi-canal [AAC<sup>+</sup>17] ont généralisé l’approche parcimonieuse à plusieurs couches de matériaux. Ces travaux permettent donc de progresser sur le défi 2.

Nous avons aussi attaqué la problématique du réalisme géomorphologique dans plusieurs de nos travaux. Dans [GGG<sup>+</sup>13], c’est un réseau hydrographique qui est généré avant de construire le terrain qui le respecte. L’écoulement réaliste de l’eau est donc garanti. Dans [CGG<sup>+</sup>17], la simulation de l’érosion implique une prise en compte du réalisme.

**Il reste extrêmement complexe de progresser sur tous les défis en même temps, car certains d’entre eux ont une nature contradictoire.** Le contrôle est difficile à obtenir lorsque l’on veut pouvoir représenter des terrains étendus. Le réalisme morphologique est souvent incompatible avec le contrôle utilisateur précis. **C’est pourquoi, la modélisation de terrains reste un problème ouvert et complexe.**

## 8.2 PERSPECTIVES DE RECHERCHE

**Un paysage virtuel est composé de bien plus qu’un simple terrain**, il contient de la végétation ainsi qu’une multitude de détails au sol mais aussi une atmosphère, des cours d’eau, *etc.* Restreindre le champ d’action aux simples terrains serait limitant, c’est pourquoi je m’attaque à d’autres thématiques. Un processus standard de création de monde virtuel va procéder en plusieurs étapes, la première étant la création du terrain. Je pense qu’il faut prendre en compte les autres éléments du paysage le plus tôt possible afin d’assurer une cohérence plus fine. Parmi ces éléments, on peut essayer d’inclure les matériaux qui se trouvent au sol, le type de végétation ou la texture. Nos travaux récents vont dans ce sens (voir [AAC<sup>+</sup>17] et [CGG<sup>+</sup>17]). Plus généralement, il convient de s’intéresser à l’atmosphère et ses nuages, aux cours d’eau, et à la végétation. Tous ces éléments ont des interactions fortes avec le terrain sous-jacent et il convient de réfléchir à des méthodes qui vont permettre de bien exprimer ce lien.

Les utilisateurs (surtout dans le monde du jeu vidéo) sont avides **de mondes de plus en plus grands**. Ubisoft avec *Ghost Recon Wildlands* avait mis en avant un monde jouable de  $16 \times 16 \text{ km}$  en 2016. Ils annoncent avec *Beyond Good and Evil 2* un monde dont l’échelle est beaucoup plus vaste encore, puisque des planètes entières seront modélisées. Cette étendue

ne doit pas être synonyme de compromis de qualité et de détails. Il devient indispensable de générer du contenu à la volée, ne serait-ce qu'à cause de l'impossibilité de tout stocker. Cela implique de trouver des méthodes suffisamment rapides pour être mises en production dans des jeux vidéo. Il s'agit là d'un challenge important auquel s'attaque le projet Papaya avec Ubisoft et dont nous sommes partie prenante.

La modélisation parcimonieuse s'est révélée très performante pour une multitude d'applications liées aux terrains. Son efficacité repose sur le fait qu'elle utilise directement des données réelles pour représenter les terrains plutôt que d'essayer d'imiter les processus extrêmement complexes qui leurs donnent naissance. Nous avons généralisé avec succès l'approche parcimonieuse à des données plus complètes, contenant notamment des informations sur la végétation ou la texture du terrain [AAC<sup>+</sup>17]. **La modélisation parcimonieuse peut se prêter à d'autres applications** telles que la génération de détails dans les nuages ou dans l'animation de rivières.

Une autre approche similaire mais en pratique très différente consisterait à utiliser les **méthodes d'apprentissage profond** pour cacher toute cette complexité. Des résultats ont été obtenus sur les terrains grâce à l'utilisation de réseaux de type Generative Adversarial Network et nous comptons les appliquer à d'autres domaines.

### 8.3 STRATÉGIE

Lorsque son domaine de recherche est la modélisation de mondes virtuels, on doit chercher à fournir aux graphistes des outils qui vont leur permettre de mieux travailler tout en conservant leur plume d'artiste. C'est une tâche extrêmement difficile. En effet, il existe deux options : ne pas consulter leur avis sur les méthodes et outils que l'on imagine, ou au contraire les impliquer dans certains choix et leur demander des retours sur l'utilisabilité. C'est cette dernière vers laquelle je veux essayer de tendre. Les artistes veulent du contrôle certes, mais ils veulent que ce contrôle soit intuitif. Si vous leur fournissez un tableau de réglages dont les noms sont incompréhensibles et dont l'action n'a pas de sens, l'outil sera forcément rejeté. Afin de ne pas reproposer des méthodes qui existent, il faut aussi connaître l'ensemble des outils du marché, sans forcément les maîtriser, mais en ayant en tête les possibilités qu'ils offrent. C'est un travail conséquent que de se former régulièrement aux outils que propose le monde professionnel. Voici ceux que j'ai approfondis (à des degrés variables) : Autodesk Maya, Eon Vue-Xstream, Mitsuba, Terragen, Blackmagic Fusion, Erdas Imagine, Isotropix Clarisse, etc.

Début 2016, je suis entré en contact avec le fondateur et directeur de la société Eonsoftware éditrice du logiciel Vue-Xstream. D'un point de vue fonctionnalités, ce logiciel couvre toutes les thématiques de la modélisation de mondes virtuels : terrain, atmosphère, décorations, végétation, écosystèmes et plantes, océan. Une thèse en CIFRE a été envisagée mais la société a été rachetée par un grand groupe ce qui n'a pas permis d'aboutir. Néanmoins, le contact privilégié avec l'entreprise a débouché sur des licences du logiciel ainsi que des sollicitations régulières pour obtenir des avis critiques sur nos résultats de recherche.

Nous bénéficions d'un autre **contact fort avec le monde de l'entreprise** grâce au projet Papaya avec Ubisoft. Même si les préoccupations d'un studio de jeux vidéos sont rythmées par la production des jeux, nous avons réussi à établir des liens étroits avec des artistes de la société, qui nous ont mené à une publication toute récente à Siggraph Asia 2017 d'une méthode d'édition interactive de terrains par le biais du *deep learning* [GDG<sup>+</sup>17].



Mes publications récentes [GDGP16, GGG<sup>+</sup>16, GDG<sup>+</sup>17] témoignent d'une **dynamique de recherche personnelle**. Ces travaux, même s'ils ont été effectués en collaboration avec mes collègues, sont avant tout des thématiques propres de recherche. Ils n'impliquent d'ailleurs pas de doctorants, et j'ai personnellement réalisé la plupart des développements qui ont permis de les publier.

L'obtention de l'Habilitation à Diriger les Recherches me permettra d'avoir une **plus grande autonomie dans l'organisation de la recherche**. L'encadrement de thèses en autonomie doit par ailleurs se développer au sein de l'équipe avec la montée en puissance de nos thématiques de recherches sur la modélisation de mondes virtuels.



**PURDUE**  
UNIVERSITY

Soulever des problématiques de recherche est une bonne chose, mais parfois les compétences requises pour répondre à ces problématiques sont complexes et nécessitent de faire appel à des collègues spécialistes par des **collaborations scientifiques**. Ces compétences peuvent être trouvées en interne au sein même du laboratoire LIRIS, comme c'est le cas de Julie Digne pour ce qui est de la modélisation parcimonieuse, et de Christian Wolf pour le *deep learning*. Pour d'autres il faut aller plus loin, comme par exemple les écosystèmes avec James Gain, la simulation avec Marie-Paule Cani, les villes avec Bedrich Benes, ou les nuages avec Yoshinori Dobashi.

Une thèse en collaboration avec l'université du Cap sur la thématique des éco-systèmes vient de démarrer (Konrad Capp, début en août 2017). Une autre thèse en collaboration avec l'université de Purdue va commencer sur la thématique de la transition entre la ville et la campagne (Mathieu Gaillard, début en janvier 2018).

## BIBLIOGRAPHIE

---

- [AAC<sup>+</sup>17] Oscar Argudo, Carlos Andujar, Antonio Chica, Eric Guérin, Julie Digne, Adrien Peytavie, and Eric Galin. Coherent multi-layer landscape synthesis. *The Visual Computer*, 33(6) :1005–1015, Jun 2017.
- [AEB06] M. Aharon, M. Elad, and A. Bruckstein. K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.*, 54(11) :4311–4322, nov 2006.
- [ASA07] Nguyen Hoang Anh, Alexei Sourin, and Parimal Aswani. Physically based hydraulic erosion simulation on graphics processing unit. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 257–264, New York, NY, USA, 2007. ACM.
- [BA05a] Farès Belhadj and Pierre Audibert. Modeling landscapes with ridges and rivers : bottom up approach. In *GRAPHITE*, pages 447–450, 2005.
- [BA05b] B. Bene and X. Arriaga. Table Mountains by Virtual Erosion. In Pierre Poulin and Eric Galin, editors, *Eurographics Workshop on Natural Phenomena*. The Eurographics Association, 2005.
- [Bar93] Michael Barnsley. *Fractals everywhere (second edition)*. Academic Press Professional, 1993.
- [BCA<sup>+</sup>14] Gwyneth A. Bradbury, Il Choi, Cristina Amati, Kenny Mitchell, and Tim Weyrich. Frequency-based controls for terrain editing. In *Proceedings of the 11th European Conference on Visual Media Production, CVMP '14*, pages 15 :1–15 :10, New York, NY, USA, 2014. ACM.
- [BCS<sup>+</sup>14] Richard Bézin, Benoît Crespin, Xavier Skapin, Olivier Terraz, and Philippe Meseure. Generalized Maps for Erosion and Sedimentation Simulation. *Computers and Graphics*, 45 :1–16, December 2014.
- [BdBG10] Shaun Bangay, David de Bruyn, and Kevin Glass. Minimum spanning trees for valley and ridge characterization in digital elevation maps. In *Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '10, pages 73–82, New York, NY, USA, 2010. ACM.
- [Bel07] Farès Belhadj. Terrain modeling : a constrained fractal model. In *Afrigraph*, pages 197–204, 2007.
- [Ben13] Morten Bendiksen. Rapid modeling of geology, March 2013.
- [BF01] Bedřich Beneš and Rafael Forsbach. Layered data representation for visual simulation of terrain erosion. In *Proceedings of the 17th Spring conference on Computer graphics*, pages 80–85, 2001.
- [BF02] Bedřich Beneš and Rafael Forsbach. Visual simulation of hydraulic erosion. In *Journal of WSCG*, volume 10, pages 79–86, 2002.
- [BFO<sup>+</sup>07] M. Beardall, M. Farley, D. Ouderkirk, C. Reimschuessel, J. Smith, M. Jones, and P. Egbert. Goblins by spheroidal weathering. In *Proceedings of the Third Eurographics Conference on Natural Phenomena*, NPH'07, pages 7–14, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.



- [BMV<sup>+</sup><sub>11</sub>] Adrien Bernhardt, André Maximo, Luiz Velho, Houssam Hnaidi, and Marie-Paule Cani. Real-time Terrain Modeling using CPU-GPU Coupled Computation. In *SIGGRAPH '11, SIGGRAPH '11 ACM SIGGRAPH 2011 Posters*, page Article No. 65, Vancouver, Canada, August 2011. ACM.
- [BNo8] Eric Bruneton and Fabrice Neyret. Real-time rendering and editing of vector-based terrains. *Computer Graphics Forum*, 27(2) :311–320, April 2008.
- [BPC<sup>+</sup><sub>10</sub>] Richard Bezin, Alexandre Peyrat, Benoit Crespin, Olivier Terraz, Xavier Skapin, and Philippe Meseure. *Interactive Hydraulic Erosion Using CUDA*, pages 225–232. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [BSP<sup>+</sup><sub>05</sub>] S. Brandel, S. Schneider, M. Perrin, N. Guiard, Jf Rainaud, P. Lienhardt, and Y. Bertrand. Automatic building of structured geological models. 5(2), 2005. *Journal of Computing and Information Science in Ingeneering*.
- [BSSo7] John Brosz, Faramarz F. Samavati, and Mario Costa Sousa. Terrain synthesis by-example. In José Braz, Alpesh Ranchordas, Helder Araújo, and Joaquim Jorge, editors, *Advances in Computer Graphics and Computer Vision*, volume 4 of *Communications in Computer and Information Science*, pages 58–77. Springer Berlin Heidelberg, 2007.
- [BTHBo6] Bedřich Beneš, Václav Těšínský, Jan Hornyš, and Sanjiv K. Bhatia. Hydraulic erosion : Research articles. *Comput. Animat. Virtual Worlds*, 17(2) :99–108, 2006.
- [CBC<sup>+</sup><sub>16</sub>] Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Adrien Peytavie, and Eric Guérin. Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Computer Graphics Forum*, 35(2) :165–175, May 2016.
- [CBPd11] D.M.D. Carli, F. Bevilacqua, C.T. Pozzer, and M.C. d’Ornellas. A survey of procedural content generation techniques suitable to game development. In *Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on*, pages 26–35, Nov 2011.
- [CCZ<sub>11</sub>] Juncheng Cui, Yang-Wai Chow, and Minjie Zhang. Procedural generation of 3d cave models with stalactites and stalagmites. *International Journal of Computer Science and Network Security*, 11(8) :94–101, 2011.
- [CGG<sup>+</sup><sub>03</sub>] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Bdam — batched dynamic adaptive meshes for high performance terrain visualization. *Computer Graphics Forum*, 22(3) :505–514, 2003.
- [CGG<sup>+</sup><sub>17</sub>] Guillaume Cordonnier, Eric Galin, James Gain, Bedrich Benes, Eric Guérin, Adrien Peytavie, and Marie-Paule Cani. Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation. *ACM Transactions on Graphics*, 36(4), 2017. The paper was presented at Siggraph 2017.
- [CMo6] R. William Clark and Ryan Mauer. Visual terrain editor : An interactive editor for real terrains. *J. Comput. Sci. Coll.*, 22(2) :12–19, December 2006.
- [CMF98] N. Chiba, K. Muraoka, and K. Fujita. An erosion model based on velocity fields for the visual simulation of mountain scenery. *Journal of Visualization and Computer Animation*, 9(4) :185–194, 1998.
- [CVG<sup>+</sup><sub>15</sub>] Leandro Cruz, Luiz Velho, Eric Galin, Adrien Peytavie, and Eric Guerin. Patch-based terrain synthesis. In *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications - Volume 1 : GRAPP, (VISIGRAPP 2015)*, pages 189–194. INSTICC, ScitePress, 2015.

- [CVL<sup>+</sup>14] Leandro Cruz, Luiz Velho, Djalma Lucio, Eric Galin, Adrien Peytavie, and Eric Guérin. Landscape Specification Resizing. In *CLEI*, pages 1–10, Montevideo, Uruguay, September 2014.
- [dCB09] Giliam J. P. de Carpentier and Rafael Bidarra. Interactive gpu-based procedural heightfield brushes. In *Proceedings of the 4th International Conference on Foundations of Digital Games, FDG '09*, pages 55–62, New York, NY, USA, 2009. ACM.
- [DCV14] Julie Digne, Raphaëlle Chaine, and Sébastien Valette. Self-similarity for accurate compression of point sampled surfaces. *Computer Graphics Forum, Proc. Eurographics 2014*, 33(2) :155–164, 2014.
- [DDGG01] D. Dambrosio, S. Digregorio, S. Gabriele, and R. Gaudio. A Cellular Automata model for soil erosion by water. *Physics and Chemistry of the Earth B*, 26 :33–39, 2001.
- [DGGK11] Evgenij Derzapf, Björn Ganster, Michael Guthe, and Reinhard Klein. River networks for instant procedural planets. *Comput. Graph. Forum*, 30(7) :2031–2040, 2011.
- [DS04] Carsten Dachsbacher and Marc Stamminger. Rendering Procedural Terrain by Geometry Image Warping. In Alexander Keller and Henrik Wann Jensen, editors, *Eurographics Workshop on Rendering*. The Eurographics Association, 2004.
- [EA06] Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR*, pages 17–22, 2006.
- [FFC82] Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Commun. ACM*, 25(6) :371–384, June 1982.
- [Fis95] Yuval Fisher, editor. *Fractal Image Compression : Theory and Application*. Springer-Verlag, London, UK, UK, 1995.
- [GDG<sup>+</sup>17] Eric Guérin, Julie Digne, Eric Galin, Adrien Peytavie, Christian Wolf, Bedrich Benes, and Benoit Martinez. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Transactions on Graphics (proceedings of Siggraph Asia 2017)*, 36(6), 2017.
- [GDGP16] Eric Guérin, Julie Digne, Eric Galin, and Adrien Peytavie. Sparse representation of terrains for procedural modeling. *Computer Graphics Forum (proceedings of Eurographics 2016)*, 35(2) :177–187, 2016.
- [Gen15] Jean-David Genevaux. *Représentation, modélisation et génération procédurale de terrains*. PhD thesis, 2015.
- [GGG<sup>+</sup>13] Jean-David Genevaux, Eric Galin, Eric Guérin, Adrien Peytavie, and Bedrich Benes. Terrain Generation using Procedural Models based on Hydrology. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 32(4), 2013.
- [GGG<sup>+</sup>16] Eric Guérin, Eric Galin, François Grosbellet, Adrien Peytavie, and Jean-David Genevaux. Efficient modeling of entangled details for natural scenes. *Computer Graphics Forum (Proceedings of Pacific Graphics 2016)*, 35(7) :257–267, 2016.
- [GGP<sup>+</sup>15] Jean-David Genevaux, Eric Galin, Adrien Peytavie, Eric Guérin, Cyril Briquet, François Grosbellet, and Bedrich Benes. Terrain Modelling from Feature Primitives. *Computer Graphics Forum*, 34(6) :198–210, May 2015.
- [GM01] Manuel Gamito and F. Kenton Musgrave. Procedural landscapes with overhangs. In *10th Portuguese Computer Graphics Meeting*, 2001.

- [GMM15] J. Gain, B. Merry, and P. Marais. Parallel, realistic and controllable terrain synthesis. *Computer Graphics Forum*, 34(2) :105–116, 2015.
- [GMS09] James Gain, Patrick Marais, and Wolfgang Strasser. Terrain sketching. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 31–38, 2009.
- [GPG<sup>+</sup>16] Francois Grosbellet, Adrien Peytavie, Éric Guérin, Éric Galin, Stéphane Mérillou, and Bedrich Benes. Environmental objects for authoring procedural scenes. *Comput. Graph. Forum*, 35(1) :296–308, February 2016.
- [GPGB11] Eric Galin, Adrien Peytavie, Eric Guérin, and Bedrich Benes. Authoring Hierarchical Road Networks. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 30(7) :2021–2030, 2011.
- [GPMG10] Eric Galin, Adrien Peytavie, Nicolas Maréchal, and Eric Guérin. Procedural Generation of Roads. *Computer Graphics Forum (Proceedings of Eurographics)*, 29(2) :429–438, 2010.
- [GT05] Eric Guérin and Eric Tosan. *Fractal Inverse Problem : Approximation Formulation and Differential Methods*, pages 271–285. Springer London, London, 2005.
- [GTB01] Eric Guerin, Eric Tosan, and Atilla Baskurt. A fractal approximation of curves. *Fractals*, 09(01) :95–103, 2001.
- [Har96] John C. Hart. Sphere tracing : a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10) :527–545, Dec 1996.
- [HGA10a] Houssam Hnaidi, Eric Guérin, and Samir Akkouche. Multiresolution control of curves and surfaces with a self-similar model. *Fractals*, 18 :271–286, September 2010.
- [HGA<sup>+</sup>10b] Houssam Hnaidi, Eric Guérin, Samir Akkouche, Adrien Peytavie, and Eric Galin. Feature based terrain generation using diffusion equation. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 29(7) :2179–2186, 2010.
- [HSS03] Kai Hormann, Salvatore Spinello, and Peter Schröder. C<sub>1</sub>-continuous terrain reconstruction from sparse contours. In Thomas Ertl, editor, *VMV*, pages 289–297. Aka GmbH, 2003.
- [IFMC03] T. Ito, Tadahiro Fujimoto, K. Muraoka, and N. Chiba. Modeling rocky scenery taking into account joints. In *Computer Graphics International, 2003. Proceedings*, pages 244–247, July 2003.
- [JBPS11] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4) :78 :1–78 :8, July 2011.
- [KBKv09] Peter Krištof, Bedřich Beneš, Jaroslav Křivánek, and Ondřej Štáva. Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum*, 28(2) :219–228, 2009.
- [KG14] Çetin Koca and Uğur Güdükbay. A hybrid representation for modeling, interactive editing, and real-time visualization of terrains with volumetric features. *International Journal of Geographical Information Science*, 28(9) :1821–1847, 2014.
- [KMN88] Alex Kelley, Michael Malin, and Gregory Nielson. Terrain simulation using a model of stream erosion. In *Proceedings of SIGGRAPH*, pages 263–268, 1988.
- [LHo4] Frank Losasso and Hugues Hoppe. Geometry clipmaps : Terrain rendering using nested regular grids. *ACM Transactions on Graphics*, 23(3) :769–776, 2004.

- [MdCPBS14] D. Michelon de Carli, C.T. Pozzer, F. Bevilacqua, and V. Schetinger. Procedural generation of 3d canyons. In *Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on*, pages 103–110, Aug 2014.
- [MDH07] Xing Mei, Philippe Decaudin, and Baogang Hu. Fast hydraulic erosion simulation and visualization on gpu. In *Pacific Graphics*, 2007.
- [MGG<sup>+</sup>10] Nicolas Marechal, Eric Guérin, Eric Galin, Stephane Merillou, and Nicolas Merillou. Heat Transfer Simulation for Modeling Realistic Winter Sceneries . *Computer Graphics Forum*, 29(2), 2010.
- [MGGA10] Nicolas Maréchal, Éric Galin, Éric Guérin, and Samir Akkouche. Component-based model synthesis for low polygonal models. In *Proceedings of Graphics Interface 2010, GI '10*, pages 217–224, Toronto, Ont., Canada, Canada, 2010. Canadian Information Processing Society.
- [Mil86] Gavin Miller. The definition and rendering of terrain maps. In *Proceedings of SIGGRAPH*, pages 39–48, 1986.
- [MKM89] F. K. Musgrave, C. E. Kolb, and R. S. Mace. The synthesis and rendering of eroded fractal terrains. In *Proceedings of SIGGRAPH*, pages 41–50, 1989.
- [MPD14] I. Martin, S. Parkes, and M. Dunstan. Modeling cratered surfaces with real and synthetic terrain for testing planetary landers. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(4) :2916–2928, October 2014.
- [MZ93] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12) :3397–3415, 1993.
- [Nag98] K. Nagashima. Computer generation of eroded valley and mountain terrains. *The Visual Computer*, 13(9-10) :456–464, 1998.
- [NLP<sup>+</sup>13] Mattia Natali, Endre M. Lidal, Julius Parulek, Ivan Viola, and Daniel Patel. Modeling terrains and subsurface geology. In *EuroGraphics 2013 State of the Art Reports (STARs)*, pages 155–173, 2013.
- [NWD05] B. Neidhold, M. Wacker, and Olivier Deussen. Interactive physically based fluid and erosion simulation. In *Eurographics Workshop on Natural Phenomena*, pages 25–32, 2005.
- [OBW<sup>+</sup>08] Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. Diffusion curves : A vector representation for smooth-shaded images. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, volume 27, 2008.
- [Olso4] Jacob Olsen. Realtime procedural terrain generation - realtime synthesis of eroded fractal terrain for use in computer games, 2004.
- [Per85] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3) :287–296, 1985.
- [Pey09] Gabriel Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1) :17–31, 2009.
- [PGMG09] Adrien Peytavie, Eric Galin, Stephane Merillou, and Jerome Grosjean. Arches : a Framework for Modeling Complex Terrains. *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2) :457–467, 2009.
- [PH93] Przemyslaw Prusinkiewicz and Marc Hammel. A fractal model of mountains with rivers. In *Graphics Interface*, pages 174–180, 1993.

- [PI13] Vladimir Alves dos Passos and Takeo Igarashi. LandSketch : A First Person Point-of-View Example-Based Terrain Modeling Approach. In Levent Burak Kara and Cindy Grimm, editors, *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. ACM, 2013.
- [RME09] Brennan Rusnell, David Mould, and Mark G. Eramian. Feature-rich distance-based terrain synthesis. *The Visual Computer*, 25(5-7) :573–579, 2009.
- [RPP93] P. Roudier, Bernard Peroche, and M. Perrin. Landscapes synthesis achieved through erosion and deposition process simulation. *Computer Graphics Forum*, 12(3) :375–383, 1993.
- [SICS<sup>+</sup>14] Aitor Santamaría-Ibirika, Xabier Cantero, Mikel Salazar, Jaime Devesa, Igor Santos, Sergio Huerta, and PabloG. Bringas. Procedural approach to volumetric terrain generation. *The Visual Computer*, 30(9) :997–1007, 2014.
- [SS05] S. Stachniak and W. Stuerzlinger. An algorithm for automated fractal terrain deformation. In *In Proceedings of Computer Graphics and Artificial Intelligence*, pages 64–76, 2005.
- [Teo09] Soon Tee Teoh. Riverland : An efficient procedural modeling system for creating realistic-looking terrains. In *Proceedings of the 5th International Symposium on Advances in Visual Computing : Part I*, pages 468–479, 2009.
- [TGM12] F. P. Tasse, J. Gain, and P. Marais. Enhanced texture-based terrain synthesis on graphics hardware. *Computer Graphics Forum*, 31(6) :1959–1972, 2012.
- [vBBK08] Ondřej Štáva, Bedřich Beneš, Matthew Brisbin, and Jaroslav Křivánek. Interactive terrain modeling using hydraulic erosion. In *ACM Siggraph/Eurographics Symposium on Computer Animation*, pages 201–210, 2008.
- [VBHS11] J. Vanek, B. Benes, A. Herout, and O. Stava. Large-scale physics-based terrain editing using adaptive tiles on the gpu. *IEEE Computer Graphics and Applications*, 31(6) :35–44, Nov 2011.
- [vLvPJ96] Joost van Lawick van Pabst and Hans Jense. Dynamic terrain generation based on multifractal techniques. In M. Chen, P. Townsend, and J.A. Vince, editors, *High Performance Computing for Computer Graphics and Visualisation*, pages 186–203. Springer London, 1996.
- [WGG99] Brian Wyvill, Andrew Guy, and Eric Galin. Extending the csg tree - warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphic Forum*, 18(2) :149–158, 1999.
- [XZZ<sup>+</sup>14] Shiyao Xiong, Juyong Zhang, Jianmin Zheng, Jianfei Cai, and Ligang Liu. Robust surface reconstruction via dictionary learning. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 33, 2014.
- [ZDL<sup>+</sup>14] Juyong Zhang, Bailin Deng, Zishun Liu, Giuseppe Patanè, Sofien Bouaziz, Kai Hormann, and Ligang Liu. Local barycentric coordinates. *ACM Trans. Graph.*, 33(6) :188 :1–188 :12, November 2014.
- [ZKT05] Z. Zhang, K. Konno, and Y. Tokuyama. 3d terrain reconstruction based on contours. In *Computer Aided Design and Computer Graphics, 2005. Ninth International Conference on*, pages 6 pp.–, Dec 2005.
- [ZSTR07] Howard Zhou, Jie Sun, Greg Turk, and James M. Rehg. Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics*, 13(4) :834–848, 2007.
- [ZT96] Chems Eddine Zair and Eric Tosan. Fractal modeling using free form techniques. *Computer Graphics Forum*, 15(3) :269–278, 1996.

## COMPARATIF DES PUBLICATIONS SUR LES TERRAINS

Article	1 Étendue	2 Multimat	3 Contrôle	4 Réalisme	Modèle
[PI13]			●		Heightmap
[ASA07]				●	Heightmap
[BdBG10]				●	Heightmap
[BFO <sup>+</sup> 07]		●	●	●	Voxel
[BA05a]			●	●	Heightmap
[Bel07]			●	●	Heightmap
[Ben13]	●	●	●	●	Couches
[BF01]		●		●	Couches
[BF02]				●	Couches
[BA05b]		●		●	Couches
[BTHB06]		●		●	Couches
[BMV <sup>+</sup> 11]	●		●		Multires Heightmap
[BPC <sup>+</sup> 10]		●		●	Maillage
[BCA <sup>+</sup> 14]			●		Heightmap
[BSP <sup>+</sup> 05]	●	●			G-cartes
[BSS07]					Heightmap
[BN08]	●	●	●	●	Vectoriel
[CBPd11]					
[MdCPBS14]				●	Heightmap
[CMF98]				●	Heightmap
[CGG <sup>+</sup> 03]					BDAM
[CM06]			●		Heightmap
[CBC <sup>+</sup> 16]			●	●	Heightmap
[BCS <sup>+</sup> 14]		●		●	G-cartes
[CCZ11]				●	
[DS04]					Heightmap
[DDGG01]				●	
[dCB09]			●		Heightmap
[DGGK11]				●	Heightmap
[FFC82]	●		●		Algorithmme

Article	1 Étendue	2 Multimat	3 Contrôle	4 Réalisme	Modèle
[GMS09]			●		Heightmap
[GMM15]			●		Vectorel + Heightmap
[GM01]	●		●		Heightmap + déplacement
[GGG <sup>+</sup> 13]	●		●	●	Heightmap + déplacement
[HSS03]	●				Heightmap
[SICS <sup>+</sup> 14]		●			Voxels + hiérarchie
[IFMC03]				●	Voxels
[KMN88]				●	Heightmap
[KG14]	●				Voxels + heightmap
[KBKv09]		●		●	Heightmap
[LH04]	●				LOD Grid
[MPD14]				●	Heightmap
[MDH07]				●	Heightmap
[MKM89]				●	Heightmap
[Nag98]				●	Heightmap
[NLP <sup>+</sup> 13]					
[NWD05]				●	Heightmap
[Olso4]				●	Heightmap
[vLvP]96]	●				Heightmap
[PGMG09]		●	●	●	Couches
[PH93]				●	Heightmap
[RPP93]		●		●	Voxel
[RME09]			●		Heightmap
[SS05]			●		Heightmap
[vBBK08]		●		●	Couches
[TGM12]			●		Heightmap
[Teo09]			●	●	Heightmap
[VBHS11]				●	Heightmap
[ZKT05]	●				Heightmap
[ZSTR07]			●		Heightmap
[GGP <sup>+</sup> 15]	●	●	●		Fonctions + hiérarchie
[GDGP16]	●		●		Sparse + heightmap
[HGA <sup>+</sup> 10b]	●		●	●	Vectorel → heightmap
Total	15	15	25	36	

TABLE 6 – Comparaison des méthodes de l'état de l'art sur les quatre critères.