



**HAL**  
open science

# Génération procédurale d'effets atmosphériques

Antoine Webanck

► **To cite this version:**

Antoine Webanck. Génération procédurale d'effets atmosphériques. Modélisation et simulation. Université de Lyon, 2019. Français. NNT : 2019LYSE1109 . tel-02454037

**HAL Id: tel-02454037**

**<https://theses.hal.science/tel-02454037>**

Submitted on 24 Jan 2020

**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.



N° d'ordre NNT : 2019LYSE1109

## THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

**l'Université Claude Bernard Lyon 1**

**École Doctorale ED512 InfoMaths**

**Spécialité de doctorat : Informatique**

Soutenue publiquement le 16/07/2019, par :

**Antoine Didier WEBANCK**

---

# Génération procédurale d'effets atmosphériques

---

Devant le jury composé de :

M<sup>me</sup> Céline Loscos, Professeure, Université Reims Champagne-Ardenne

Président(e)

M. Loïc BARTHE, Professeur, Université de Toulouse

Rapporteur

M<sup>me</sup> Saida BOUAKAZ, Professeure, Université Lyon 1

Examinatrice

M. Jean-Michel DISCHLER, Professeur, Université de Strasbourg

Rapporteur

M<sup>me</sup> Céline Loscos, Professeure, Université Reims Champagne-Ardenne

Examinatrice

M. Éric GALIN, Professeur, Université Lyon 1

Directeur de thèse

M. Éric GUÉRIN, Professeur, Université Lyon 1

Co-directeur



# UNIVERSITE CLAUDE BERNARD - LYON 1

## **Président de l'Université**

Président du Conseil Académique

Vice-président du Conseil d'Administration

Vice-président du Conseil Formation et Vie Universitaire

Vice-président de la Commission Recherche

Directrice Générale des Services

**M. le Professeur Frédéric FLEURY**

M. le Professeur Hamda BEN HADID

M. le Professeur Didier REVEL

M. le Professeur Philippe CHEVALIER

M. Fabrice VALLÉE

Mme Dominique MARCHAND

## ***COMPOSANTES SANTE***

Faculté de Médecine Lyon Est — Claude Bernard

Directeur : M. le Professeur G. RODE

Faculté de Médecine et de Maïeutique Lyon Sud — Charles Mérieux

Directeur : Mme la Professeure C. BURILLON

Faculté d'Odontologie

Directeur : M. le Professeur D. BOURGEOIS

Institut des Sciences Pharmaceutiques et Biologiques

Directeur : Mme la Professeure C. VINCIGUERRA

Institut des Sciences et Techniques de la Réadaptation

Directeur : M. X. PERROT

Département de formation et Centre de Recherche en Biologie Humaine

Directeur : Mme la Professeure A-M. SCHOTT

## ***COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE***

Faculté des Sciences et Technologies

Directeur : M. F. DE MARCHI

Département Biologie

Directeur : M. le Professeur F. THEVENARD

Département Chimie Biochimie

Directeur : Mme C. FELIX

Département GEP

Directeur : M. Hassan HAMMOURI

Département Informatique

Directeur : M. le Professeur B. SHARIAT

Département Mathématiques

Directeur : M. le Professeur G. TOMANOV

Département Mécanique

Directeur : M. le Professeur H. BEN HADID

Département Physique

Directeur : M. le Professeur J-C PLENET

UFR Sciences et Techniques des Activités Physiques et Sportives

Directeur : M. Y. VANPOULLE

Observatoire des Sciences de l'Univers de Lyon

Directeur : M. B. GUIDERDONI

Polytech Lyon

Directeur : M. le Professeur E. PERRIN

Ecole Supérieure de Chimie Physique Electronique

Directeur : M. G. PIGNAULT

Institut Universitaire de Technologie de Lyon 1

Directeur : M. le Professeur C. VITON

Ecole Supérieure du Professorat et de l'Education

Directeur : M. le Professeur A. MOUGNIOTTE

Institut de Science Financière et d'Assurances

Directeur : M. N. LEBOISNE

# Remerciements

En premier lieu, il convient de préciser que ces travaux de thèse ont principalement été financés par les Investissements d'Avenir dans le cadre du projet PAPAYA, en collaboration avec Ubisoft, le Commissariat à l'énergie atomique et aux énergies alternatives (CEA), l'Université Paris-Est Marne-la-Vallée (UPEM) et Télécom ParisTech. Le but de ce projet était d'imaginer de nouvelles méthodes procédurales afin de générer d'énormes mondes digitaux pour la prochaine génération de jeux vidéo. Cela incluait la génération de terrains, d'effets atmosphériques et l'animation de rivières. Merci donc aux investisseurs et aux acteurs du projet PAPAYA.

Merci ensuite aux membres de mon jury d'avoir participé à l'évaluation de mes travaux, d'avoir pris le temps de vous y plonger, que ce soit pour le manuscrit ou la soutenance. Merci en particulier à mes deux rapporteurs, Loïc BARTHE et Jean-Michel DISCHLER, pour vos analyses très détaillées de mon manuscrit.

Merci beaucoup à mes deux encadrants, Éric GALIN et Éric GUÉRIN, de m'avoir accepté sur ce sujet de thèse et de m'avoir épaulé durant ces quatre années sans étouffer mon autonomie. Vous avez le même prénom et les mêmes initiales, et écrire vos noms complets pour éviter les ambiguïtés dans nos échanges par mail a toujours été un peu perturbant, mais vos personnalités différentes sont complémentaires. Si je n'ai toujours pas appris à me servir d'Inkscape, je suis maintenant formé aux primitives vectorielles de PowerPoint.

Petite pensée à mes anciens enseignants de l'Université de Strasbourg et aussi à mes anciens collègues de labo et amis durant mon stage au Laboratoire d'Informatique Graphique de l'Université de Montréal (LIGUM). Merci à Nicolas VIBERT, Joël POLARD-PERRON, Melino CONTE, Gilles-Philippe PAILLÉ, Olivier MERCIER, David QUIROZ MARIN, David MILAENEN, Aude GIRAUD, Cynthia BEAUCHEMIN, Adrien DUBOUCHET, Arnaud SCHOENTGEN, et Pierre POULIN d'avoir créé l'enthousiasme nécessaire pour que je me lance dans une thèse en informatique graphique. J'espère vous revoir un de ces jours pour collaborer ou même vous croiser au détour d'une conférence.

Merci à tous mes collègues du Laboratoire d'Informatique en Image et Systèmes d'information (LIRIS) ; vous avez fait de la vie du laboratoire celle que j'ai appréciée et beaucoup d'entre vous ne sont maintenant plus seulement des collègues, mais sont même devenus des amis.

Merci au directeur du laboratoire, Mohand-Saïd HACID, pour le soutien porté aux doctorants en général et en particulier aux représentants doctorants.

Merci à Guillaume BOUCHARD pour tes explications passionnées concernant la programmation fonctionnelle, le langage Haskell et Nix.

Merci à Xavier URBAIN de t'être proposé de relire mon manuscrit ; de tout temps, l'homme a mordu la tétine, et nous perpétuerons la tradition ! Allez, je reste encore un petit moment au labo et je file, c'est promis.

Un merci particulier à nos secrétaires Sylvie OUDOT, Catherine LOMBARDI, Isabelle BUISSON et Brigitte GUYADER. Merci non seulement pour votre efficacité mais aussi votre gentillesse qui en a presque rendu toutes les démarches administratives agréables ; sans vous, je serais certainement encore perdu dans un formulaire.

Merci à Françoise CONIL pour ta curiosité et tes initiatives de partage des connaissances pratiques, notamment par l'intermédiaire des cafés développeurs.

Merci aux membres de l'équipe Modélisation Géométrique, Géométrie Algorithmique, Génération Procédurale (GeoMod) pour les présentations scientifiques et les goûters.

Merci à Thierry EXCOFFIER pour nos échanges concernant Linux et surtout le langage Python. Cela ne m'aura pris que quatre ans pour commencer à écrire des scripts Python plutôt que des scripts shell.

Merci à Nicolas BONNEEL pour tes explications sur le transport optimal et pour tes questions désinhibées qui repoussent les incompréhensions jusque dans leurs derniers retranchements.

Merci à Guillaume LAVOUÉ, Julie DIGNE et à Raphaëlle CHAINE pour votre soutien à Eurographics 2018, les répétitions et les soirées.

Merci à Jean-Claude LEHL pour nos discussions au coin café, sur la terrasse [des fumeurs ?] ou devant mon code bogué. Tu ne faisais pas partie de mon encadrement, mais j'ai toujours pu compter sur ton aide et ta passion contagieuse concernant le rendu.

Merci à mes collègues du département informatique de l'Université Claude Bernard (Elo-die DESSERÉE, Marie LEFEVRE, Nathalie GUIN et Thierry EXCOFFIER) et de l'INSA Lyon (Mihailoan POPESCU, Yann CORTIAL, Pierre-Edouard PORTIER, Mathieu MARANZANA et Christine SOLNON). Merci pour votre confiance et pour notre collaboration.

Merci à Éric DUCHÈNE et Aline PARREAU, et aux autres acteurs de la cellule médiation pour tous les événements que vous organisez et pour m'avoir donné l'occasion d'y participer moi aussi.

Merci aux doctorants des années précédentes (François GROSBELLET, Jean-David GÉNEVAUX, Hélène PERRIER, Marie-Neige CHAPEL et Joseph GARNIER) pour votre accueil et votre aide. Je n'oublierai pas la convivialité de vos goûters de fin de journée. Merci aussi à Hélène et Joseph pour vos actions en tant que représentants des doctorants, j'ai été fier de reprendre votre flambeau.

Merci à Marc HEINRICH, le plus doué d'entre nous, à Valentin GLEDEL, le maître des cookies et de la trinquerie, à Agathe HERROU, la gardienne des hérissons, à Gabrielle PARIS, l'épicène, ainsi qu'à Vincent JAILLOT, Beatrix-Emőke FÜLÖP-BALOGH, Tong FU, Rémy DELANAUX, Axel PARIS, Oscar ARGUDO, Loïs PAULIN, Gabriel MEYNET, Yana NEHMÉ et ceux que ma mémoire rationnelle aurait oublié, pour leur présence au quotidien, nos discussions

partagées au cours des séminaires, conférences, repas, et pauses mais aussi lors des soirées et sorties.

Merci à Yann CORTIAL pour notre collaboration sur la génération de nuages et nos échanges sur le calcul GPU.

Merci à Basile FABRONI, le *renderer*, pour nos échanges et l'entraide en informatique graphique, et les jeux de plateau.

Merci à Julia SANCHEZ, la charitable, pour l'organisation du weekend de descente de l'Ardèche.

Merci à Charles BARNOUIN, le médecin, pour nos pauses/poses dessin ; je t'accompagne à un atelier quand tu veux !

Merci à Matthieu HEITZ et Julia LUKACS, les futurs Canadiens, pour les soirées et pour l'hébergement, l'étymologie et la philosophie ainsi que la compagnie de la douce Ortie.

Merci à Jocelyn BERNARD, le cinéphile et bon vivant, pour tes sélections cinématographiques, pour tes soupes, le saucisson, le fromage et les vins.

Merci à Yohann BEARZI, le géométricien ricain, pour les cafés, le bouchon, la sauce piquante et tes explications patientes de tes travaux à un néophyte.

Merci à Thomas CAISSARD, le jeune marié perpétuel, d'en avoir [r]amené plus d'un lors de nos soirées et pour des parties mémorables de Rencontre Cosmique.

Merci à Samuel CARENSAC, l'optimisateur poivré, pour les délicieux macarons, la tarte au citron et les parties de jeux vidéo et de plateau.

Merci à Simon PAGEAUD, l'agent autonome aux multiples personnalités, pour le jeu de rôle, nos parties de jeux de plateau, et l'avant-première où nous avons collaboré dans l'adversité ; je persiste à dire qu'un match nul est meilleur qu'une défaite !

Merci à Maxime SAINTLOT, le doctorant qui rêvait de vivre en camion, pour ta bonne humeur, ton entrain à découvrir de nouvelles choses, pour l'escalade et les randonnées ; j'attends ton futur *device*.

Merci à Rémi CASADO, mon ingénieur préféré, pour tes photos de nuages, de m'avoir demandé de te montrer mes dernières belles images de synthèse, pour tes critiques de films qui, bien que négatives, ont certainement participé à leur appréciation de ma part.

Merci à Arthur CRENN, l'animateur de service, pour les soirées, les crêpes et la découverte du Kouign Amann et du chouchen, les pauses café, les <sup>2</sup>, *ctm*, *cmb*, *ktl* et autres biais de langage, ainsi que pour une bonne partie des cadeaux.

Merci à Thibault DUPONT, avec qui j'ai commencé mes travaux exactement au même moment, dans le même projet, la même équipe, avec le même directeur de thèse, et dans le même bureau. Durant quatre ans, nous nous sommes entraidés pour résoudre les problèmes

que nous avons immanquablement rencontrés. Merci camarade, de m'avoir supporté tous les jours, que mon code fonctionne ou non, pour les innombrables débats que tu as initiés et tes meilleures blagues. Merci de m'avoir écouté palabrer de nombreuses fois devant notre tableau et sache qu'orlib attend toujours son heure de gloire, je reste persuadé que les *templates* vaincrons ! Merci aussi à toi et Mélanie pour le nouvel an à la neige et la sortie raquettes.

Merci à Antoine DAILLY, le Mexicain, toi aussi co-bureau de la première heure. Merci d'avoir fait ta thèse en maths et pas en informatique, histoire de varier un peu. Merci de m'avoir fait découvrir Brandon SANDERSON et son petit *Mistborn* ; ça faisait longtemps que je n'avais pas opté pour une œuvre de fantasy plutôt que de science-fiction. Merci aussi pour la découverte du jeu Seasons, qui est l'élément le plus joué de ma ludothèque. Merci de m'avoir laissé profiter de ton avance dans la majorité des procédures administratives ; tu sais bien que les formulaires et les systèmes informatiques ont toujours eu une aversion pour moi.

Merci à Alice JOFFARD, la chercheuse qui prend son métier à la lettre, toi qui a été ma troisième co-bureau. Merci pour les croissants, d'avoir porté une oreille attentive aux dilemmes de mon projet professionnel, pour les soirées, pour la cuisine méditerranéenne (on sait tous que tu as remporté le concours ce soir là), et pour la plupart des cadeaux. *Murcu ussu pur nu duscussuus du lu lugu du ductur pululu, du Urthur su duluctu tu.*

Merci à Alexis LEBIS, l'écrivain, toi avec qui j'ai partagé le mandat de responsable des doctorants. Merci d'avoir encadré le Congrès des Enfants avec moi, pour ta foi en la recherche et en nos actions en tant que représentants des doctorants. Merci pour Nex Machina, pour la découverte de ProtonMail et d'une pléthore de mots inusités, et pour les annonces sur Telegram.

Merci à mes amis et à toute ma famille, d'avoir supporté mes absences autant physiques que psychiques lorsque j'avais la tête dans les nuages. Merci à mes parents d'avoir pris le temps d'écouter mes explications sur mes travaux de thèse et sur le monde académique.

À mes grands-parents. Merci pour votre amour et le temps passé ensemble, en montagne, dans la forêt ou à la mer.

Et merci à ma copine, pour m'avoir supporté et soutenu dans les moments les plus tendus des *deadlines* et de la rédaction du manuscrit notamment ; merci d'enrichir ma vie au quotidien et au delà du travail.

# Abstract

This thesis focusses on the synthetization of natural landscapes, and more particularly on their celestial part. The aspect of the sky is governed by plenty of atmospheric phenomena, among which clouds play a major role for they are recurrent and widespread. Even without directly considering the sky, the density of the clouds allows them to intensely modify the global illumination of a landscape. The work of this thesis thus focuses mainly on the editing, modelling and animation of cloud areas of landscape dimensions.

Because simulating the thermodynamics behind the formation of clouds hardly provides any control and the details of the simulated volume are quickly limited, we propose instead a procedural generation method. We build a lightweight cloudscape model as a hierarchy of functions. The finest details are obtained by composing procedural noises and reproduce the specific shapes of different kinds of clouds. The large-scale cloud presence is described at a high level and at different times by maps drawn by the user. These discrete maps are transformed into implicit static primitives and then interpolated by morphing, accounting for relief and winds in order to produce coherent trajectories. The implicit field obtained by mixing the interpolating primitives represents the spatiotemporal field of cloud density. Images are finally synthesized by rendering of the atmospheric participative medium according to our own implementation, executed in parallel on a graphic card.

**Key Words:** computer graphics, atmosphere, clouds, cloudscape, modeling, animation, morphing, implicit surfaces, procedural methods, rendering, generated images, participating media.



# Résumé

Cette thèse s'intéresse à la synthétisation de paysages naturels, et plus particulièrement, à leur portion céleste. L'aspect du ciel est gouverné par de nombreux phénomènes atmosphériques parmi lesquels les nuages jouent un rôle prépondérant car ils sont fréquemment présents et couvrent de grandes étendues. Même sans considérer directement le ciel, la densité des nuages leur permet de modifier intensément l'illumination globale d'un paysage. Les travaux de cette thèse se concentrent donc principalement sur l'édition, la modélisation et l'animation d'étendues nuageuses aux dimensions d'un paysage.

Comme la simulation thermodynamique de la formation des nuages est difficilement contrôlable et que les détails du volume simulés sont rapidement limités, nous proposons plutôt une méthode par génération procédurale. Nous érigeons un modèle léger de paysage nuageux sous forme d'une hiérarchie de fonctions. Les détails les plus fins sont obtenus par composition de bruits procéduraux et reproduisent les formes de différents genres de nuages. La présence nuageuse à grande échelle est quant à elle décrite à haut niveau et à différents instants par des cartes dessinées par l'utilisateur. Ces cartes discrètes sont transformées en primitives implicites statiques ensuite interpolées par métamorphose en prenant en compte le relief et les vents pour produire des trajectoires cohérentes. Le champ implicite obtenu par mélange des primitives interpolantes constitue le champ spatiotemporel de densité nuageuse. Des images sont finalement synthétisées par rendu du milieu participatif atmosphérique selon notre propre implémentation exécutée en parallèle sur carte graphique.

**Mots clefs** : informatique graphique, atmosphère, nuages, paysage nuageux, modélisation, animation, métamorphose, surfaces implicites, méthodes procédurales, rendu, images de synthèse, milieux participants.





# Table des matières

<b>Remerciements</b>	<b>3</b>
<b>Abstract</b>	<b>7</b>
<b>Résumé</b>	<b>9</b>
<b>Introduction</b>	<b>15</b>
<b>1. Etat de l'art</b>	<b>19</b>
1.1. Météores	19
1.1.1. Électrométéores	19
Foudre et arcs électriques	19
Aurores polaires	21
1.1.2. Hydrométéores	21
Nuages	22
Précipitations	22
1.1.3. Photométéores	23
Arcs-en-ciel	23
Halos solaires, arcs blancs, gloires et parhélies	25
Rayons crépusculaires et trouées de lumière	26
Mirages	27
1.2. Rendu atmosphérique	28
1.2.1. Modèles atmosphériques sans nuages	29
1.2.2. Simulation de transport radiatif atmosphérique	30
Méthodes aux éléments finis	30
Méthodes de MONTE CARLO	37
Rendu atmosphérique temps réel	39
1.2.3. Synthèse	43
1.3. Modélisation de nuages	44
1.3.1. Simulations physiques	44
Simulations eulériennes	46
Automates cellulaires	48
Simulations lagrangiennes	49
1.3.2. Génération procédurale	51
Esquisses	52
Méthodes basées exemple	53
Modélisations et animations procédurales <i>ex nihilo</i>	56
1.3.3. Synthèse	61
1.4. Conclusion	61
<b>2. Prérequis</b>	<b>65</b>
2.1. Modélisation implicite	65
2.1.1. Primitive implicite à squelette	65
2.1.2. Opérateurs de combinaison de champs de potentiel	66

2.1.3.	Fonctions de distance	68
2.1.4.	Fonctions d'atténuation	68
	Fonctions d'atténuation à support restreint	71
2.1.5.	Décomposition de primitive implicite en mélange équivalent	72
2.1.6.	Conclusion	73
2.2.	Bruits procéduraux	75
2.2.1.	Bruits à bande fréquentielle restreinte	75
	Bruit de valeur	78
	Bruit de gradient	79
	Bruit cellulaire	80
	Bruit de crêtes	81
2.2.2.	Bruits à multiples bandes fréquentielles	82
	Bruit fractal	83
	Bruit de turbulence	83
	Bruit fractal de crête	84
2.2.3.	Conclusion	84
<b>3.</b>	<b>Modèle de paysage nuageux</b>	<b>87</b>
3.1.	Fonction d'atmosphère	87
	3.1.1. Types de nuage	87
	3.1.2. Fonctions de couche	89
3.2.	Fonctions de présence	92
	3.2.1. Primitives de présence	94
	3.2.2. Génération des primitives de présence	95
	3.2.3. Animation et métamorphose	96
3.3.	Fonctions de texture nuageuse	96
	3.3.1. Motifs nuageux	96
	Motif stratiforme	96
	Motif cumuliforme	97
	Motif cirriforme	97
	3.3.2. Animation de texture nuageuse et dynamique de la turbulence	97
	Animation par le vent	97
	Animation propre turbulente	97
3.4.	Conclusion	102
<b>4.</b>	<b>Métamorphose de paysages nuageux</b>	<b>103</b>
4.1.	Calcul des trajectoires	105
	4.1.1. Environnement	105
	4.1.2. Contraintes des trajectoires	105
	4.1.3. Fonction de coût	107
	4.1.4. Calcul de plus court chemin spatio-temporel anisotrope	110
4.2.	Mise en correspondance optimale	114
	4.2.1. Heuristiques de mise en correspondance	114
	4.2.2. Heuristique optimale	116
4.3.	Ajout de primitives fantômes	119
	4.3.1. Seuillage du coût des trajectoires	120
	4.3.2. Extrapolation et primitives fantômes	121
	4.3.3. Cohérence et choix de la valeur de seuil	122
4.4.	Primitives interpolantes	122
	4.4.1. Interpolation entre primitives statiques	123

4.4.2. Interpolation entre primitive statique et primitive fantôme . . . . .	124
4.5. Résultats . . . . .	125
4.5.1. Contrôle . . . . .	125
4.5.2. Performances . . . . .	127
4.5.3. Validation . . . . .	128
4.6. Conclusion . . . . .	129
<b>Conclusion et perspectives</b>	<b>131</b>
<b>A. Rendu atmosphérique</b>	<b>135</b>
A.1. Milieu participatif . . . . .	135
A.2. Densité et probabilité d'interaction . . . . .	135
A.3. Diffusion et fonction de phase . . . . .	136
A.3.1. Diffusion de RAYLEIGH . . . . .	136
A.3.2. Diffusion de MIE . . . . .	136
Fonction de phase de HENYEV-GREENSTEIN . . . . .	137
A.4. Équation de transfert . . . . .	137
A.4.1. Transfert atmosphérique multimatériaux et spectral . . . . .	138
A.5. Lancer de rayon . . . . .	139
A.5.1. Approche par <i>single scattering et raymarching</i> . . . . .	139
Ordre zéro . . . . .	141
Ordre un . . . . .	141
Évaluation de la transmittance par quadrature . . . . .	142
A.5.2. Approche par échantillonnage de l'espace des chemins . . . . .	142
Échantillonnage de la profondeur . . . . .	142
Échantillonnage de la direction . . . . .	145
Évaluation de la transmittance par <i>ratio tracking</i> . . . . .	145
<b>Tables des symboles</b>	<b>147</b>
Contexte général . . . . .	147
Surfaces implicites . . . . .	147
Bruits . . . . .	147
Rendu . . . . .	148
Modèle d'atmosphère . . . . .	148
Métamorphose . . . . .	149
<b>Bibliographie</b>	<b>151</b>



# Introduction

Depuis des millénaires, nous étudions, nous nous interrogeons et cherchons à recréer le monde qui nous entoure, que ce soit sous forme de dessins, de peintures, de sculptures, ou plus récemment de mondes virtuels et d'images de synthèse. L'informatique graphique est un domaine de recherche jeune, entre la technique et l'art, qui s'inscrit dans ce processus créatif. Que ce soit pour des jeux-vidéos ou des films, une partie des objectifs en informatique graphique est de synthétiser des scènes [re]créant un environnement naturel réaliste. Un paysage naturel englobe généralement un relief plus ou moins prononcé, de la végétation et des cours d'eau, et un ciel plus ou moins nuageux. Cette thèse se concentre sur la synthèse de la portion céleste des paysages naturels et plus précisément, elle se concentre sur les phénomènes qui ont lieu au sein de l'atmosphère. La considération même de l'atmosphère lors du rendu d'une image de synthèse complexifie grandement l'opération car les différents constituants de l'atmosphère diffusent la lumière. La diffusion atmosphérique est notamment responsable de la couleur bleue du ciel durant la journée, et de la teinte rougeoyante du soleil à son lever et à son coucher. Il existe de nombreux autres phénomènes atmosphériques et météorologiques, tels que l'arc-en-ciel ou bien les tempêtes orageuses par exemple, qui s'ajoutent à la diffusion atmosphérique et que les membres de la communauté de l'informatique graphique se sont intéressés à synthétiser. Parmi les effets atmosphériques, les nuages se révèlent particulièrement importants en cela qu'ils sont intrinsèquement liés à un bon nombre de phénomènes atmosphériques qu'ils rendent possibles. Ainsi, pour chacun des exemples précédents de l'arc-en-ciel et de la tempête orageuse et les précipitations qui ne manquent pas de l'accompagner, les nuages sont indispensables à leur réalisation. De plus, les nuages se manifestent particulièrement souvent et jouent un rôle majeur dans l'apparence et l'*atmosphère* d'un paysage. En effet, un ciel couvert de sombres nuages est souvent annonciateur de mauvais temps aussi bien symboliquement que météorologiquement. À l'inverse, lorsque le soleil perce une épaisse couche nuageuse en traçant des rayons crépusculaires, l'effet majestueux produit est symbolique d'espoir. Au coucher ou au lever du soleil, les nuages se parent de couleurs flamboyantes, contribuant ainsi à la beauté du ciel et ajoutant des effets spectaculaires au paysage. Pour ces raisons, cette thèse s'est principalement concentrée sur la modélisation et l'animation de nuages à dimension d'un paysage.

Les formes complexes que revêtent les nuages sont causées par une succession de perturbations partiellement chaotiques régies par les lois de la thermodynamique. Une approche permettant d'obtenir des formes nuageuses réalistes consiste à simuler informatiquement les lois thermodynamiques qui permettent l'évaporation de l'eau, la convection et les vents qui brassent les masses d'air humide ainsi que la condensation de l'eau lorsque les masses d'air sont saturées. Ce type de simulation fonctionne bien lorsque la quantité de masses d'air simulées reste faible, c'est-à-dire si le volume simulé et la finesse de sa discrétisation sont restreints. Dans le cas contraire, le coût de calcul passe mal à l'échelle et devient très rapidement prohibitif. Les simulations météorologiques mobilisent ainsi des supercalculateurs pendant des jours afin de prédire grossièrement le comportement des masses d'air, avec une fiabilité limitée spatialement et temporellement. Le niveau de détail et les dimensions des nuages sont donc limités par la puissance de calcul, et leur comportement chaotique

dégénère rapidement avec les itérations de la simulation. De plus, la plupart des simulations ne permettent pas de contrôler les formes des nuages ni les trajectoires qu'ils suivent. Le contrôle est indirect et réside quasiment exclusivement dans la définition de l'état initial du système et dans l'implémentation des lois thermodynamiques. À l'inverse, il existe toute une classe de méthodes qui s'intéressent à la création plus directe des nuages, en donnant le contrôle à l'utilisateur : les méthodes procédurales. Plutôt que de simuler la formation et l'évolution des nuages, certaines méthodes procédurales se servent d'esquisses grossières de nuages qu'elles amplifient et détaillent, tandis que d'autres généralisent des exemples de nuages fournis par l'utilisateur afin d'en générer de nouveaux. Tout le problème des méthodes procédurales est de fournir un outil qui simplifie le travail de l'utilisateur sans aller à l'encontre de ses directives et qui produise un résultat réaliste. Bien qu'il existe de bonnes méthodes procédurales pour générer des nuages statiques, très peu de méthodes permettent de produire des nuages animés réalistes facilement.

Cette thèse s'inscrit dans cette problématique : au lieu de réaliser une simulation thermodynamique complexe, nous érigeons un modèle procédural de paysage nuageux animé qui se veut efficace et contrôlable. L'ensemble des volumes nuageux est modélisé directement par une fonction continue définie par une hiérarchie de fonctions de manière à rendre locale l'évaluation de la densité atmosphérique en tout point de l'espace et à tout moment. Afin d'assurer la localité spatiale, la fonction de densité nuageuse globale est séparée en plusieurs fonctions représentant les densités de couches nuageuses indépendantes. Dans une approche similaire à la classification des nuages employée en météorologie, chaque couche nuageuse correspond à un type de nuage qui existe à des altitudes données et présente un motif particulier. Ainsi, les nuages en couches quasiment homogènes sont qualifiés de stratiformes, les nuages convectifs concentrés sous l'apparence d'amas bulbeux sont dits cumuliformes, et les nuages les plus hauts en altitudes qui sont constitués de cristaux de glace et qui s'effilochent au vent sont qualifiés de cirriformes. Bien qu'il soit possible d'éditer manuellement les couches nuageuses, c'est une tâche d'autant plus fastidieuse que l'étendue spatiale et temporelle couverte par les nuages animés. C'est pourquoi nous séparons la définition de la fonction de chaque couche nuageuse en deux parties : une fonction contrôlant à haut niveau la présence et le placement à grande échelle de nuages, et une fonction apportant automatiquement les motifs haute fréquence et la texture propres au type de nuage associé à la couche. Chaque fonction de détails prend la forme d'une combinaison de fonctions de base qui génèrent des variations aléatoires mais autosimilaires d'un motif animé. Le type et l'échelle spatiotemporelle des fonctions de bases employées et la méthode de combinaison sont prédéterminés avec soin de manière à reproduire au mieux les motifs nuageux observés dans la nature. Afin de permettre à un utilisateur de contrôler la présence des nuages à grande échelle, nous lui demandons de dessiner grossièrement des cartes de répartition des nuages pour chaque couche et à des instants clefs de son choix.

Notre méthode se charge d'interpoler les cartes fournies afin de produire une animation cohérente des nuages d'après le terrain situé sous les nuages ainsi que les conditions de vent. Plutôt que d'interpoler directement les cartes dessinées, elles sont discrétisées sous la forme de primitives implicites. Le mélange des primitives obtenues pour chaque instant clef donné forme ainsi une fonction lisse qui contrôle spatialement où se situent les nuages et à quel point ils sont denses. Afin d'interpoler ces fonctions de présence entre les images clefs successives, nous proposons une méthode de métamorphose en quatre étapes. Tout d'abord, nous recherchons les plus courts chemins spatiotemporels entre toutes les primitives successives selon une fonction de coût prenant en compte les

conditions de vent et le terrain. D'après les coûts des plus courts chemins et la masse nuageuse des primitives, nous appliquons ensuite une nouvelle heuristique de mise en correspondance fondée sur un algorithme de transport optimal discret. Le plan de transport optimal fournit le meilleur compromis pour mettre en correspondance toutes les primitives successives. Dans les cas problématiques tels que des nuages isolés des autres par le relief ou les vents, il peut cependant arriver qu'il soit impossible de relier une primitive isolée aux autres sans violer certaines contraintes d'intersection avec le terrain ou de vitesse maximale. Il arrive parfois aussi que certains des plus courts chemins aillent à l'encontre du réalisme ou du souhait de l'animateur. La troisième étape de notre méthode permet de contourner les cas problématiques des masses nuageuses les plus isolées et non souhaitées en les retirant effectivement des relations de correspondance, après le calcul du plan de transport optimal. Le comportement des masses nuageuses isolées est alors indépendant et consiste à suivre directement les trajectoires préalablement calculées qui sont les plus adaptées aux conditions de vent et au relief. En résulte des effets de disparition et d'apparition progressifs des anciennes et des nouvelles primitives isolées. L'étape finale de notre méthode de métamorphose est l'instanciation des primitives animées associées aux primitives en correspondance et aux primitives isolées, et qui réalisent l'interpolation continue des fonctions de présence.

Avec cette thèse, nous présentons un cadre de travail original et unifié pour la génération efficace et le contrôle de grands paysages nuageux mettant en scène différents types de nuages. Les nuages sont modélisés en combinant des fonctions qui définissent leur forme à des fonctions de plus haut niveau qui contrôlent la présence de couverture nuageuse. Nous présentons une nouvelle méthode pour l'édition intuitive de paysages nuageux animés et nous introduisons notamment un algorithme de métamorphose efficace permettant de calculer une interpolation cohérente entre deux paysages nuageux donnés. À notre connaissance, notre approche est la première permettant à un utilisateur de contrôler la création de paysages animés en utilisant un algorithme de métamorphose de champ de densité continu, prenant en compte le relief du terrain et le champ de vent, tout en fournissant un traitement unifié de différents types de nuages. La motivation majeure de nos travaux est de définir le paysage nuageux complet sous la forme d'une fonction analytique variant dans le temps, qui ne nécessite pas de simulation. Ces travaux ont de nombreuses applications, en particulier pour l'industrie du jeu qui a besoin de méthodes capables de contrôler la météo sur de grandes étendues.





# Etat de l'art

De nombreux travaux de recherche en informatique graphique s'intéressent à recréer des effets atmosphériques dans des mondes synthétiques afin d'y ajouter du réalisme ou de la richesse visuelle.

Cet état de l'art est organisé en trois sections. La section 1.1 regroupe les travaux de synthèse de divers phénomènes atmosphériques aussi appelés météores tandis que la section 1.2 page 28 s'intéresse aux travaux porte sur le rendu de l'atmosphère et des nuages comme milieu participatif. Enfin, la section 1.3 page 44 se concentre sur la modélisation et l'animation des éléments les plus distinctifs de l'atmosphère terrestre : ses nuages.

## 1.1 Météores

Le terme météore désigne, au sens large, tout phénomène atmosphérique et pas seulement les corps célestes qui brûlent en entrant dans l'atmosphère comme le laisse supposer le langage courant. Il existe différentes classes de météores qui regroupent les phénomènes atmosphériques qui se manifestent de la même manière. Les manifestations électriques comme la foudre forment la famille des électrométéores. Les phénomènes qui font intervenir l'eau, tels que les précipitations, constituent les hydrométéores, les phénomènes optiques comme l'arc-en-ciel ou les mirages sont appelés photométéores. Enfin, les manifestations de particules solides telles que les aérosols ou les tempêtes de sable sont des lithométéores.

### 1.1.1 Électrométéores

Parmi les météores, les électrométéores regroupent les phénomènes liés à des décharges électriques dans l'atmosphère. La foudre qui accompagne les orages est un des électrométéores les plus connus mais ce n'est pas le seul type de décharge électrique liée aux orages, et les aurores polaires font aussi partie des électrométéores.

#### **Foudre et arcs électriques**

Un événement qui se produit en présence de nuages, et notamment lors d'orages et les précipitations qui les accompagnent, est la foudre. La foudre est un arc électrique géant créé par un différentiel électrique (tension) si intense que la décharge est conduite à travers

l'air, entre le sol et une cellule orageuse, ou entre différentes parcelles orageuses. Quand la foudre frappe, l'arc électrique formé éclaire (éclair) et fait vibrer (tonnerre) très puissamment l'atmosphère durant un court instant, c'est un phénomène transitoire. L'illumination engendrée est même visible depuis l'espace.



**FIGURE 1.1.** – Photographies du phénomène de foudre. De gauche à droite : foudre partant d'un *Cumulonimbus* vue depuis un avion (MUKHOPADHYAY [100]), foudre entre nuages d'un orage (WEBANCK [140]), et foudre frappant la surface (BRESSON [16]).

En informatique graphique, GLASSNER [46] et KIM *et al.* [77] se sont intéressés à la modélisation des arcs électriques, tandis que DOBASHI *et al.* [24] se sont concentrés sur l'éclairage diffus produit par la foudre avec les nuages environnants. La reproduction sonore du tonnerre a été couverte par GLASSNER [47].



**FIGURE 1.2.** – De gauche à droite : les images de foudre synthétisées par GLASSNER [46], DOBASHI *et al.* [24] et KIM *et al.* [77].

Il existe d'autres types de décharges appelées phénomènes lumineux transitoires, liées aux conditions orageuses, mais plus exotiques car elles ne se produisent que lors des orages les plus importants et se manifestent à bien plus haute altitude que les nuages. Les phénomènes lumineux transitoires regroupent notamment les farfadets (*red sprites*) qui ressemblent à des méduses rouges géantes ou à des carottes, les jets bleus et les jets géants qui ont une apparence de chalumeau électrifié, et les elfes (*ELVES for Emission of Light and Very low frequency perturbations due to Electromagnetic Pulse Sources*) qui sont des halos rouges diffus. Ces phénomènes sont encore mal compris de la science et font l'objet d'études comme par exemple la mission spatiale du satellite Taranis, dont le lancement est prévu pour 2019 par le CNES, consacrée à leur observation. Ainsi, bien qu'ils sont spectaculaires, aucun des phénomènes lumineux transitoires n'a encore été reproduit en informatique graphique, mais SURKOV *et al.* [134] fournissent des pistes pour y remédier sous la forme d'un aperçu des connaissances actuelles quant à leur formation.

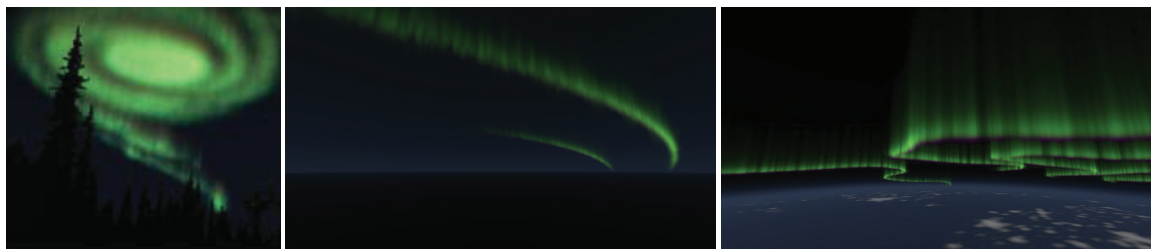
## Aurores polaires

Les aurores polaires – boréales dans l'hémisphère nord et australes dans l'hémisphère sud – sont des manifestations atmosphériques qui diaprent la haute atmosphère lorsqu'elle est excitée par les particules du vent solaire. Du fait de l'orientation et de la forme du champ magnétique terrestre, ce phénomène haut en couleur est surtout observé proche des pôles magnétiques.



**FIGURE 1.3.** – Photographies du phénomène d'aurores polaires. De gauche à droite : depuis la surface terrestre en Alaska (NICKEL [106]), et en Norvège (TIEDE [137]), et depuis l'orbite de la Station Spatiale Internationale (GERST [43]).

Quelques travaux ont été menés en informatique graphique par BARANOSKI *et al.* [2, 3, 4] pour simuler le phénomène des aurores boréales par lancer stochastique de particules dans la magnétosphère. Une autre méthode consiste à modéliser directement le flux des particules selon les lignes de champ de la magnétosphère [69]. Enfin, LAWLOR *et al.* [85] ont modélisé plus simplement les aurores en simulant un fluide dans un masque 2D à la manière de filaments de fumée qui se déforment en créant des vortex. Ces filaments 2D sont ensuite étirés verticalement pour reproduire le drapé des aurores avant de les rendre efficacement sur GPU.



**FIGURE 1.4.** – De gauche à droite : les images d'aurores synthétisées par BARANOSKI *et al.* [4], ISHIKAWA *et al.* [69] et LAWLOR *et al.* [85].

### 1.1.2 Hydrométéores

Les hydrométéores sont les phénomènes atmosphériques liés à la présence de particules d'eau sous toutes ses formes dans l'atmosphère. Ainsi, les nuages et les précipitations en font partie, de même que l'accumulation de particules d'eau par la condensation qui produit la rosée par exemple.

## Nuages

Les nuages sont un phénomène très visible au quotidien et ils font partie des hydrométéores car ils sont constitués d'eau. Dans la mesure où les travaux menés durant cette thèse se sont principalement concentrés sur la modélisation, l'animation et le rendu des nuages, l'état de l'art correspondant aux nuages en informatique graphique fait l'objet de deux sections complètes : la section 1.2 page 28 couvre le rendu atmosphérique des nuages tandis que la section 1.3 page 44 couvre leur modélisation et leur animation.

## Précipitations

Les nuages sont constitués de gouttelettes d'eau et de cristaux de glace en suspension dans l'atmosphère. Lorsque les phénomènes combinés de condensation et d'agglomération de ces particules d'eau les rend trop massives, elles chutent et se précipitent vers la surface, qu'une portion d'entre-elles finira par atteindre si elles ne s'évaporent pas avant. Ainsi, les précipitations qui accompagnent parfois les nuages peuvent prendre plusieurs formes telles que la pluie ou la neige par exemple.



**FIGURE 1.5.** – Photographies de différents types de précipitations. De gauche à droite : de la pluie en train de tomber (KLAJBAN [78]), un manteau neigeux qui recouvre le sol en montagne (WEBANCK [141]), et de gros grêlons de plus de 5 cm après un violent orage (NATIONAL OCEANIC *et al.* [104]).

Vues à grande distance, les précipitations s'apparentent aux nuages et leurs particules peuvent être modélisées et rendues de manière similaire. En revanche, pour un observateur au milieu des précipitations, les particules sont assez grosses pour distinguer leurs formes et leurs trajectoires. Un point important dans la perception de la pluie par exemple est la vitesse de chute des gouttes qui semblent s'étaler et tracer des traits du fait de la persistance rétinienne dans le cas de l'œil humain ou du temps d'exposition dans le cas d'une caméra.

LANGER *et al.* [84] s'intéressent au rendu des précipitations de neige. Ils modélisent le champ de densité animé des flocons dans l'espace écran par synthèse spectrale et le composent avec un système particules. GARG *et al.* [42] proposent une méthode de rendu photoréaliste de pluie. Ils modélisent précisément les oscillations que les gouttes d'eau manifestent durant leur trajectoire vers la surface et créent une base de données dont les exemples servent d'instances. WEBER *et al.* [143] proposent une méthode de rendu temps réel des précipitations de pluie et prennent en compte la vitesse de chute qui se traduit par un flou de mouvement des gouttes de pluie.





**FIGURE 1.6.** – De gauche à droite : de la pluie synthétisée par GARG *et al.* [42], et par WEBER *et al.* [143], et de la neige synthétisée par LANGER *et al.* [84].

Les précipitations qui rencontrent la surface du sol ou d'autres objets provoquent de nombreuses interactions à la limite du domaine des effets atmosphériques. Ainsi, TATARCHUK [135] se sont notablement intéressés à reproduire les interactions entre de la pluie et une scène synthétiques : les gouttes de pluie qui humidifient les surfaces, qui rebondissent en éclaboussant les alentours, et qui s'accumulent pour former des flaques et ruisseler, geler ou se réévaporer.

### 1.1.3 Photométéores

Les photométéores sont l'ensemble des phénomènes observables dans l'atmosphère et causés par une interaction optique particulière entre la lumière solaire, voire lunaire, et l'environnement atmosphérique. L'arc-en-ciel appartient aux photométéores, de même que d'autres types d'arcs et de halos moins connus, ainsi que les rayons crépusculaires et les mirages.

#### Arcs-en-ciel

L'arc-en-ciel est un phénomène optique qui se produit fréquemment dans la nature lorsque des averses sont éclairées par le soleil ou la lune par exemple. La lumière passe à travers les gouttes d'eau et rend apparentes les différentes composantes du spectre lumineux sous la forme d'une arche multicolore visible dans le ciel. Il arrive parfois que plusieurs arcs soient visibles au lieu d'un seul, voire même que des arcs se croisent. Certains de ces arcs secondaires présentent un dégradé de couleur dans le sens inverse du dégradé de l'arc principal.



**FIGURE 1.7.** – Photographies d’arcs-en-ciel. De gauche à droite : un arc-en-ciel et la pluie tombant depuis un *Cumulonimbus* (LALLI [83]) ; une série d’arcs surnuméraires à l’intérieur et un arc secondaire à l’extérieur d’un arc-en-ciel primaire (GRUBB [53]).

Le phénomène optique qui désigne cet étalement des différentes bandes spectrales est appelé la dispersion et s’observe notamment lorsqu’un faisceau de lumière blanche est dirigé à travers un prisme. Dans le cas des arcs-en-ciel, ce sont les gouttelettes d’eau qui jouent le rôle de prisme et qui dispersent la lumière et la redirigent vers l’observateur.

INAKAGE [68] reproduit le phénomène d’arc-en-ciel à arc unique sans simuler le phénomène de dispersion (voir Figure 1.8 page suivante). Il utilise simplement l’angle formé par la direction d’observation et l’orientation de l’illumination solaire pour interpoler les couleurs de l’arc-en-ciel qui ont été observées à des angles donnés dans la nature. Ainsi, des angles de  $42^\circ$ ,  $41^\circ$  et  $40^\circ$  correspondent au rouge, au vert et au bleu respectivement.

MUSGRAVE [102] détaille comment simuler le phénomène d’arc-en-ciel dans le cadre de l’optique géométrique en calculant les trajectoires que suivent des rayons lumineux en traversant une goutte d’eau. Des rayons parallèles sont lancés à travers une goutte d’eau : ils traversent l’interface (dioptré) air-eau en étant réfractés, ils sont réfléchis par la paroi de la goutte, et sont à nouveau réfractés en quittant la goutte. Après ces multiples déviations, la répartition angulaire de l’intensité lumineuse est variable : on parle de caustiques. Afin de simuler correctement la dispersion, des rayons sont lancés à différentes longueurs d’onde du spectre visible, donnant des indices et des angles de réfraction différents. Des longueurs d’onde uniformément réparties sur le spectre visibles sont choisie avant d’être perturbées stochastiquement afin de lisser les échantillons.

Le cadre de l’optique géométrique est trop limité dans le cas général de particules de petite taille par rapport à la longueur d’onde. La solution de MIE donne une expression analytique de la fonction de phase dans le cas général de particules sphériques. Il est donc normal qu’elle soit utilisée afin de reproduire le phénomène de diffusion de la lumière par des gouttelettes d’eau sphériques et de simuler des arcs-en-ciel synthétiques [71, 124].



**FIGURE 1.8.** – De gauche à droite : l'image d'arc-en-ciel directement synthétisée par INAKAGE [68] à partir des angles d'observation et d'illumination, les images d'arc-en-ciel synthétisées par JACKÉL *et al.* [71] et RILEY *et al.* [124] qui utilisent la théorie de MIE, et un arc-en-ciel avec un arc secondaire et des arcs surnuméraires synthétisés par SADEGHI *et al.* [126]

Cependant, plus une gouttelette d'eau grossit, plus sa forme réelle s'éloigne de la sphère parfaite en s'écrasant. Des travaux récents [126] généralisent la théorie de MIE pour des gouttelettes de forme écrasée. Considérer plus précisément le caractère ondulatoire de la lumière dans tout le spectre permet de modéliser finement les effets de réfraction, de dispersion, de polarisation, de diffraction et d'interférence de la lumière par les gouttelettes d'eau. Des types particuliers d'arcs-en-ciel observés dans la nature, avec de multiples arcs, peuvent ainsi être synthétisés.

### Halos solaires, arcs blancs, gloires et parhélies

Lorsque l'atmosphère est chargée de cristaux de glace, un phénomène de halo lumineux entourant le soleil ou la lune peut être observé. Il est aussi possible d'observer des sources lumineuses surnuméraires appelées parhélies dans le cas du soleil, et parasélènes pour la lune.



**FIGURE 1.9.** – De gauche à droite : photographie d'un halo solaire accompagné de parhélies (TENBERGEN [136]); photographie des phénomènes de gloire et d'arc blanc (INAGLORY [65]).

Dans une étude très complète, HONG *et al.* [62] explicitent comment les formes hexagonales des cristaux de glace présents dans l'atmosphère réfléchissent la lumière selon des angles particuliers pour créer le phénomène de halo atmosphérique. Ils détaillent aussi comment simuler le phénomène par méthode de MONTE CARLO en s'inspirant des travaux précédents menés en optique [52] et en informatique graphique [44, 45, 49].



Deux autres phénomènes optiques se manifestent sur les nuages s'ils sont observés dans la direction antisolaire : la gloire et l'arc blanc (*fogbow*). La gloire est une série d'anneaux colorés entourant l'ombre projetée par l'observateur sur le nuage, tandis que l'arc blanc est un anneau de plus grand rayon que ceux de la gloire. Une méthode de rendu atmosphérique du milieu nuageux considérant les hautes fréquences de la fonction de phase de  $M_{IE}$  permet de recréer ces phénomènes [13, 14, 124].



**FIGURE 1.10.** – De gauche à droite : gloire synthétisée par RILEY *et al.* [124], halo solaire synthétisé par RILEY *et al.* [124], et gloire et arc blanc synthétisés par BOUTHORS *et al.* [14]

### Rayons crépusculaires et trouées de lumière

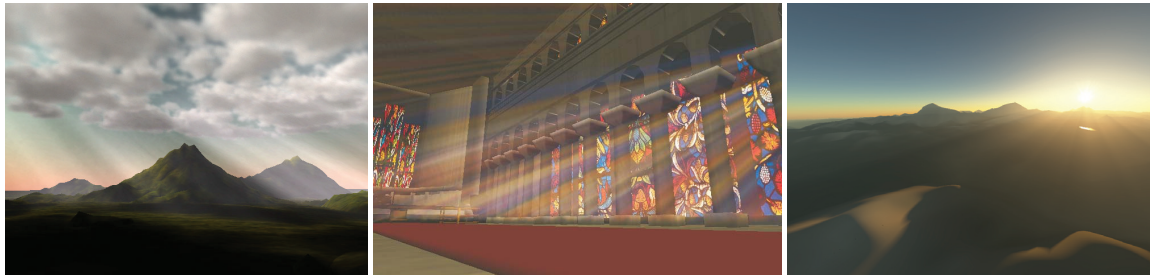


**FIGURE 1.11.** – Photographies de rayons crépusculaires. De gauche à droite : rayons crépusculaires tracés par des arbres et leur feuillage (RACHERLA [121]); rayons crépusculaires créés par des nuages qui occultent la lumière solaire (WEBANCK [139]).

Lorsqu'une source de lumière est partiellement obstruée et qu'une ombre est projetée dans un milieu participatif tel que l'atmosphère, elle n'est pas seulement visible sur la prochaine surface éclairée, mais dans tout le volume participatif qui se retrouve dans son ombre. Ce phénomène est particulièrement remarquable lorsqu'une couche nuageuse dense est trouée par endroit par la lumière solaire, à l'ombre d'une montagne, ou sous une canopée partiellement éclaircie (figure 1.11). Les trouées de lumière (*light shaft*) rendent notamment visibles les trajectoires des rayons solaires primaires qui arrivent parallèles, et sont appelés rayons crépusculaires (*god rays*).

MAX [94] et NISHITA *et al.* [108] s'intéressent au rendu des trouées de lumière rendues visibles par le milieu atmosphérique. Les méthodes qu'ils proposent consistent à calculer les volumes des ombres projetées par les polygones de la scène (*shadow volumes*) puis

à intégrer la portion de lumière diffusée par le milieu atmosphérique vers la caméra, selon l'approximation de la diffusion unique (*single scattering*). Par la suite, les méthodes plus génériques de rendu atmosphérique de BRUNETON *et al.* [17] et DOBASHI *et al.* [25, 29] permettent aussi de produire des effets de trouées de lumière.



**FIGURE 1.12.** – Rayons crépusculaires synthétisés par (de gauche à droite) : DOBASHI *et al.* [29], DOBASHI *et al.* [25] et BRUNETON *et al.* [17].

## Mirages

Certains travaux réalisés en informatique graphique considèrent la courbure de la trajectoire lumineuse par le phénomène de réfraction atmosphérique. Cela permet de simuler plus précisément la diffusion atmosphérique [58, 133] mais est surtout nécessaire à la simulation des effets de mirages.

Un exemple de mirage couramment observable est le mirage dit *de la route mouillée*. Dans des conditions d'ensoleillement intense, dans la direction d'une longue route, un observateur peut avoir l'impression de voir de l'eau miroiter sur la route et refléchir le décor environnant, bien qu'il n'y ait pas réellement d'eau.



**FIGURE 1.13.** – Photographies de différents types de mirages. De gauche à droite : mirage inférieur ou mirage chaud dit *de la route mouillée* (INAGLORY [66]) ; mirage supérieur ou mirage froid (SALLAZ-DAMAZ [129]) ; séquence de photographies du disque solaire montrant différentes déformations et le reflet vert (INAGLORY [67]).

Cela s'explique par le phénomène de réfraction subit par la lumière qui passe entre le ciel et l'observateur, au-dessus de la route chauffée. L'échauffement de la route crée un gradient de température important et par là même un gradient de l'indice de réfraction de l'air. Plutôt que de suivre une direction rectiligne, le chemin de la lumière est alors courbé, et la direction d'origine de la lumière perçue par l'utilisateur ne pointe pas vers la route mais un peu plus haut, donnant un effet de réflexion.

Le mirage de la route mouillée est un exemple de mirage inférieur, *id est* un mirage dont l'image réfléchie est située sous l'objet réel. Les nuages inférieurs sont généralement dus à un échauffement important du sol. À l'inverse, il existe des mirages dits inférieurs, dont l'image se situe au-dessus de l'objet réel. Des mirages inférieurs sont ainsi observables au-dessus de grandes étendues gelées par exemple.

Les mirages inférieurs et supérieurs sont des cas particuliers produits par une variation monotone de l'indice de réfraction. Il est tout-à-fait possible de considérer des profils de variation plus complexes [57, 131, 133]. Un exemple de mirage dû à un profil de variation plus complexe se produit parfois en regardant le soleil à travers l'atmosphère, qui est alors parfois très déformé, présentant une forme très éloignée du disque habituel : un aplatissement ou des discontinuités de courbure, voire même un dédoublement.

Un autre phénomène qui s'observe sur le disque solaire est le reflet vert [56, 57, 88]. Lorsque le soleil est bas dans le ciel, voire un peu sous la limite de l'horizon, il est parfois possible d'observer un reflet vert au-dessus du disque solaire, et inversement, un reflet rouge sous le disque solaire. Cela s'explique par le phénomène de dispersion de la lumière. La variation de densité de l'atmosphère en fonction de l'altitude crée un gradient de l'indice de réfraction, courbe la trajectoire de la lumière vers la surface et crée une faible dispersion de la lumière. Bien que l'effet de dispersion de la lumière par l'atmosphère soit très faible et inobservable dans le cadre des mirages évoqués précédemment [103], la bien plus importante longueur parcourue par la lumière dans le cas du reflet vert le permet [57].

## 1.2 Rendu atmosphérique

L'atmosphère terrestre joue un rôle prépondérant dans l'illumination des scènes naturelles, d'extérieur, ou étendues. En effet, lorsqu'une partie des rayons lumineux parcourent une grande distance dans l'atmosphère une partie non négligeable de l'énergie lumineuse est absorbée ou diffusée par l'atmosphère, ce qui influence notamment la couleur de la lumière qui arrive jusqu'aux objets de la scène et donne, entre autres, son bleu au ciel et son rouge au soleil couchant. Aussi, de nombreux travaux en informatique graphique ont été consacrés à la simulation de la diffusion atmosphérique, et cette section constitue l'état de l'art associé.

L'atmosphère est aussi le domaine des nuages qui influencent grandement l'illumination d'une scène. Une chape de nuages stratiformes agit par exemple comme un diffuseur qui transforme l'éclairage solaire directe en une lumière ambiante blafarde. Des îlots de *Cumulus* qui se déplacent au gré des vents peuvent au contraire donner un effet majestueux à l'éclairage d'un paysage en projetant des rayons crépusculaires, ou en se révélant la toile céleste embrasée au coucher du soleil. Cette importance des nuages dans l'éclairage d'une scène s'explique notamment parce qu'ils sont beaucoup plus denses que le reste de l'atmosphère et diffusent fortement la lumière. De plus, les formes complexes que prennent les nuages qui se déforment et se déplacent au gré des vents rend leur modélisation un problème qui demeure non résolu, même dans le domaine météorologique.

Du fait de leur complexité, de nombreuses méthodes de rendu atmosphérique ignorent les nuages. Les modèles d'atmosphère ainsi privée de nuages qui ont été élaborés avant

toute simulation de rendu sont couverts dans la section 1.2.1. Différentes méthodes de calcul permettant de simuler le transport radiatif atmosphérique en présence ou en l'absence de nuages sont ensuite développées dans la section 1.2.2 page suivante. Quant à la modélisation de nuages, elle constitue le cœur des travaux de cette thèse et pour cette raison, l'état de l'art associé est traité séparément dans la section 1.3 page 44.

### 1.2.1 Modèles atmosphériques sans nuages

La première étape avant de simuler un transport radiatif atmosphérique consiste à définir un modèle de l'atmosphère et des différents types de particules qui la composent et influencent le transport radiatif de la lumière. Parmi les particules atmosphériques visibles figurent notamment les molécules des gaz présents dans l'air et de la couche d'ozone, la brume sèche qui contient des aérosols tels que des poussières, du pollen et des spores, ainsi que les nuages qui sont composés de gouttelettes d'eau et cristaux de glace.

À petite échelle, la densité atmosphérique varie assez peu et l'atmosphère peut être représentée de manière simplifiée par un mélange homogène de particules. Ainsi, KLASSEN [79] modélise l'atmosphère par deux strates superposées de densités constantes : une couche à basse altitude contenant de la brume sèche et de l'air, et une couche haute contenant seulement de l'air, mais plus raréfié. De plus, l'auteur néglige la courbure du globe et le domaine des strates est donc parallélipédique, ce qui n'est pas le cas dans les travaux de HABER *et al.* [58] qui discrétisent eux aussi l'atmosphère selon l'altitude en strates homogènes, mais dans le domaine sphérique.

Bien que ces modèles atmosphériques considèrent des strates homogènes, ils comprennent plusieurs strates dont les densités décroissent avec l'altitude car globalement, l'atmosphère se raréfie exponentiellement avec l'altitude. Pour l'air, cela s'explique par un état d'équilibre entre la diffusion moléculaire et l'attraction gravitationnelle, et c'est assez similaire pour les aérosols qui sont arrachés de la surface par les vents mais ont tendance à retomber progressivement. KANEDA *et al.* [76] et NISHITA *et al.* [110] sont parmi les premiers en informatique graphique à avoir mis en œuvre de tels modèles faisant décroître la densité exponentiellement avec l'altitude, et ces modèles sont aujourd'hui très communs dans le domaine.

Cependant, l'atmosphère se révèle en réalité fortement hétérogène, même à altitude constante. En effet, il ne faut pas ignorer que l'atmosphère est continuellement brassée par des vents qui soulèvent et charrient les poussières de la surface. Combinés à l'évaporation et à la condensation, les vents facilitent aussi la formation et la déformation des nuages qui viennent encore enrichir les variations atmosphériques locales. Aucune des publications de cet état de l'art ne considère de variation plus complexe que la décroissance exponentielle pour les densités en aérosols ou même en air car cela n'aurait qu'un impact négligeable sur l'aspect de la scène. En revanche, les variations de densité en gouttelettes d'eau et en cristaux de glace conditionnent les formes des nuages que de nombreux travaux s'attachent à modéliser (voir Section 1.3 page 44).



## 1.2.2 Simulation de transport radiatif atmosphérique

Simuler le transport radiatif de la lumière dans le milieu participatif qu'est l'atmosphère pose un certain nombre de défis techniques :

- L'étendue spatiale du volume atmosphérique. Pour un point de vue situé en son sein, l'atmosphère englobe l'ensemble du champ de vision et agit à la fois comme obstacle et comme vecteur pour le passage de la lumière. En plus de déterminer si la lumière interagit avec l'ensemble des surfaces d'une scène, il faut alors aussi déterminer si la lumière interagit avec l'ensemble du volume ambiant.
- L'étalement énergétique sur de multiples ordres de diffusion. L'importance de l'éclairage direct est amoindrie par l'atmosphère et il devient nécessaire de simuler de nombreux rebonds pour obtenir une bonne approximation de l'illumination globale.
- Le caractère spectral de la diffusion. Comme l'air qui diffuse plus fortement la composante lumineuse bleue que la rouge, les particules qui constituent l'atmosphère ne diffusent pas la lumière de manière constante sur le spectre lumineux et les chemins empruntés par les rayons lumineux dépendent alors de leur longueur d'onde. Pour reproduire fidèlement les couleurs du ciel et de l'atmosphère, le rendu atmosphérique doit ainsi considérer le domaine spectral.
- L'hétérogénéité de la densité atmosphérique. La densité atmosphérique décroît exponentiellement avec l'altitude, mais différemment pour tous ses composants, et c'est encore pire en présence de nuages qui ajoutent des variations importantes et à haute fréquence de densité. L'hétérogénéité de l'atmosphère complexifie la définition de précalculs et empêche notamment le calcul analytique de la transmittance, qui doit alors être calculée par échantillonnage du domaine atmosphérique.

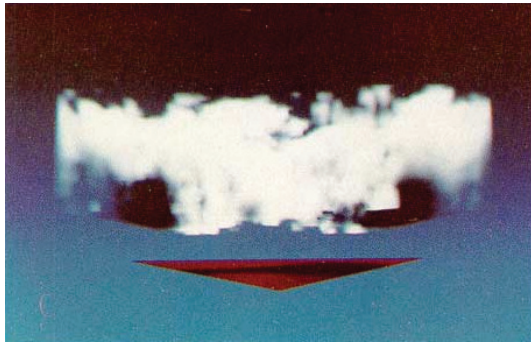
Afin de répondre à ces défis, différentes méthodes de rendu atmosphérique ont été développées en informatique graphique. Elles sont groupées et passées en revue dans la suite de cette section.

### Méthodes aux éléments finis

Les méthodes aux éléments finis cherchent à discrétiser le problème de la diffusion atmosphérique dans l'espoir d'en précalculer une partie ou du moins de factoriser les calculs répétitifs. Ces méthodes ont l'avantage de restreindre la complexité du rendu en fonction de la finesse de discrétisation, mais au prix de nombreuses approximations.

**Approximation de diffusion unique** BLINN [10] est un des premiers en informatique graphique à proposer une méthode de rendu d'un milieu participatif et de nuages en s'inspirant de l'étude des transferts radiatifs réalisés en physique des plasmas et en astronomie. Il définit les concepts de milieu participatif, de densité, de fonction de phase, de diffusion et d'albédo et donne les expressions de plusieurs fonctions de phase telles que la fonction de phase de HENYEU-GREENSTEIN ou la fonction de phase associée à la diffusion de RAYLEIGH. Il décrit l'approximation de diffusion unique (*single scattering*) qu'il juge correcte jusqu'à un albédo de 0.3 et applique cette approximation dans le cadre du rendu des anneaux de Saturne et d'une couche nuageuse simplifiée. Considérer les milieux des anneaux et de

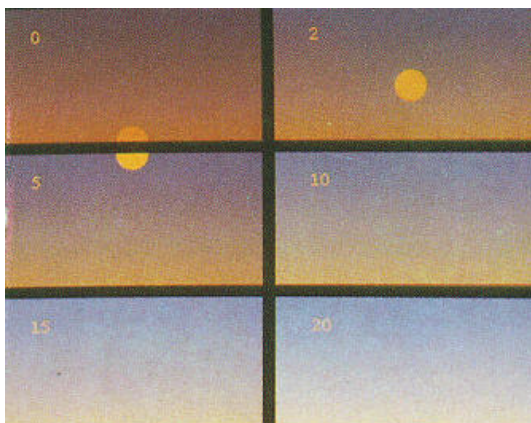
l'atmosphère comme uniformes lui permet d'exprimer analytiquement la variation d'intensité lumineuse due à la traversée de ces milieux par des rayons lumineux. BLINN évoque même comment calculer le rendu dans le cas plus complet de multiples diffusions (*multiple scattering*), par la méthode des ordonnées discrètes [18], en utilisant des chaînes de MARKOV (MONTE CARLO), ou encore la méthode d'approximation à deux faisceaux (*doubling or adding method*).



KAJIYA *et al.* [74] vont plus loin en généralisant la méthode de BLINN [10] à des milieux participatifs homogènes, éclairés par de multiples sources lumineuses, et présentant un albédo élevé. Le volume nuageux est discrétisé en cellules de densités données, puis le rendu est réalisé par deux passes de lancer de rayon. À la première passe, les rayons sont lancés depuis chaque source lumineuse afin d'obtenir l'intensité lumineuse perçue par chaque cellule pour

chacune des sources. La seconde passe consiste en un lancer de rayon depuis les pixels rendus afin d'accumuler la réduction de transmittance, et d'accumuler l'intensité lumineuse retransmise par les cellules le long de chaque rayon et multipliée. Cette méthode en deux passes constitue une évaluation de la diffusion de la lumière monochrome par les nuages, selon l'approximation de la diffusion unique (*single scattering*). La diffusion spectrale de la lumière solaire par l'atmosphère n'est pas considérée et les nuages sont rendus sur fond de dégradé bleu. Enfin, deux points sont obscurs dans l'article : la complexité en nombre de rayons lancés durant chaque étape de la méthode qui n'est pas précisée par les auteurs, ainsi que la précision de l'échantillonnage le long de chaque rayon.

MAX [95] considère le rendu de nuages de densité constante de manière similaire à BLINN [10] mais remplace la décroissance exponentielle de la transmittance en fonction de la profondeur optique par un polynôme quadratique, ce qui revient à choisir une distance limite à partir de laquelle toute lumière est absorbée par les nuages en fonction de leur densité.



KLASSEN [79] est le premier en informatique graphique à proposer une méthode de rendu du ciel prenant en compte la diffusion spectrale de la lumière solaire par l'atmosphère. L'atmosphère est modélisée par deux couches superposées de densité constante : une couche à basse altitude contenant de la brume sèche et de l'air, et une couche haute contenant seulement de l'air. L'auteur explique bien que pour les molécules d'air (domaine de la diffusion de RAYLEIGH), la fonction de phase ne dépend pas de la longueur d'onde, mais que c'est bien le cas de la section

efficace de diffusion (*scattering cross section*) qui est inversement proportionnelle à la puissance quatre de la longueur d'onde. Autrement dit, la répartition angulaire de la diffusion est indépendante de la longueur d'onde, mais pas la probabilité de déviation d'un rayon. Pour les particules composant la brume sèche qui sont beaucoup plus grandes que les molécules d'air (domaine de la diffusion de MIE), la fonction de phase dépend au contraire

du ratio entre la taille des particules et la longueur d'onde. Quant à leur section efficace de diffusion, elle dépend aussi du ratio entre taille et longueur d'onde mais beaucoup plus faiblement, et KLASSEN précise même que le continuum de tailles des particules composant la brume sèche vient encore amortir cette dépendance. En prenant en compte ces spécificités spectrales, l'intensité lumineuse et la transmittance sont intégrées indépendamment selon l'approximation de diffusion unique (*single scattering*) à chaque longueur d'onde nécessaire.



**Ray marching** En 1993, NISHITA *et al.* [110] traitent du rendu d'une scène représentant la Terre et son atmosphère vues depuis l'espace ou tout autre position, pas seulement depuis la surface. Un tel point de vue possiblement très haut en altitude, voire dans l'espace, empêche l'utilisation d'un unique dôme lumineux (voir section 1.2.2 page suivante), mais les auteurs mettent à profit les symétries de la scène afin d'optimiser la complexité du rendu. En effet, en considérant les rayons solaires comme parallèles et le globe terrestre comme sphérique, les conditions d'éclairage

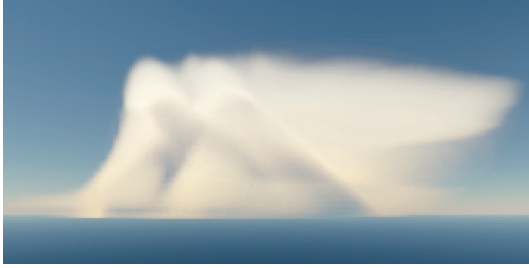
sont symétriques autour de l'axe Terre-soleil. Ainsi, la transmittance entre un point et la surface ne dépend que de l'angle au zénith et de l'altitude, et elle peut donc être précalculée en deux dimensions. Puisque le modèle d'atmosphère utilisé par les auteurs est celui d'une fonction de densité qui décroît exponentiellement avec l'altitude, il n'existe plus d'expression analytique de la transmittance d'un rayon traversant la couche atmosphérique selon une orientation quelconque. La transmittance est alors évaluée numériquement par *ray marching* en choisissant un pas qui n'est pas régulier mais plutôt inversement proportionnel à l'altitude et à la densité de l'atmosphère afin de limiter les erreurs d'intégration. Une fois les valeurs de transmittance précalculées, la symétrie de la scène peut encore être exploitée dans le cas de la radiance à la surface du globe terrestre, car elle ne dépend que de l'angle entre le zénith et l'axe Terre-soleil.



**Découpage selon la bissectrice** KNISS *et al.* [80, 81] dissèquent le fonctionnement du rendu volumique dans le contexte de la visualisation de données médicales. Les auteurs innovent en proposant une méthode ingénieuse de rendu lorsque seul l'éclairage direct est considéré, ce qui correspond à l'approximation de la diffusion unique (*single scattering*). Leur méthode qu'ils

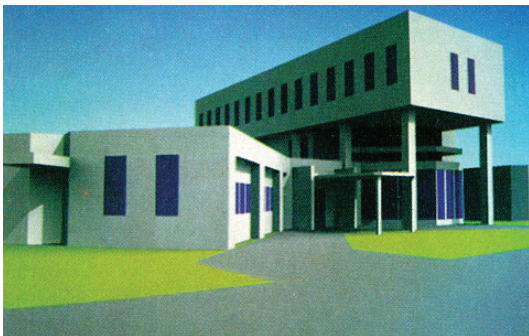
nomment découpage selon la bissectrice (*half angle slicing*) consiste à intégrer de concert la transmittance à la fois depuis la source lumineuse et depuis la caméra en une seule et même passe. La motivation principale est d'éviter de répéter inutilement les mêmes calculs de transmittance sur les mêmes segments. Pour ce faire, des tranches successives du volume sont rendues le long de la bissectrice de l'angle formé par la direction d'observation et la direction d'éclairage qui sont toutes deux préalablement placées dans le même quadrant (la direction d'observation peut être inversée).





RILEY *et al.* [122, 123, 124] mettent en œuvre la technique de découpage selon la bissectrice (*half angle slicing*) pour estimer la diffusion nuageuse et proposer un rendu atmosphérique hybride entre diffusion unique (*single scattering*) et multiples diffusions (*multiple scattering*). L'illumination céleste est calculée simplement dans l'approximation de diffusion unique (*single scattering*)

comme le produit entre l'intensité solaire et la transmittance en ignorant la courbure du globe afin d'exprimer analytiquement la transmittance de l'atmosphère en fonction de l'altitude. Afin d'appliquer le découpage selon la bissectrice (*half angle slicing*) dans le cas des nuages, le volume nuageux est discrétisé le long de la bissectrice en voxels de densité constante. De plus, afin d'estimer de multiples diffusions (*multiple scattering*) nuageuses, les premiers ordres de diffusion de la fonction de phase de MIE sont estimés par auto-convolution en ignorant toute diffusion arrière (*back scattering*). À chaque pas d'intégration dans les voxels nuageux, la transmittance depuis le point de vue est intégrée comme dans la méthode de découpage selon la bissectrice (*half angle slicing*) classique. En revanche, depuis la source lumineuse, ce n'est plus seulement la transmittance qui est intégrée, mais aussi l'ouverture angulaire d'après la fonction de phase de MIE pré-convoluée. L'ordre de diffusion de la fonction de phase est quant à lui estimé comme égal au nombre moyen d'événements de diffusion qui se produisent à la traversée du voxel de transmittance donnée, *id est* proportionnellement à l'albédo et à la profondeur optique du voxel. L'éclairage solaire initialement focalisé s'étale ainsi sur un angle qui augmente progressivement du fait des multiples diffusions au sein du nuage.



**Dôme céleste** NISHITA *et al.* [109] proposent de remplacer l'atmosphère par un dôme représentant le ciel comme une source lumineuse hémisphérique centrée sur le point de vue. Comme l'intensité lumineuse du ciel varie suivant la direction d'observation et la position du soleil, la surface du dôme est découpée en bandes comme des quartiers d'une orange, selon l'angle formé entre la direction du soleil et le zénith. L'intensité moyenne de chacune des bandes est estimée

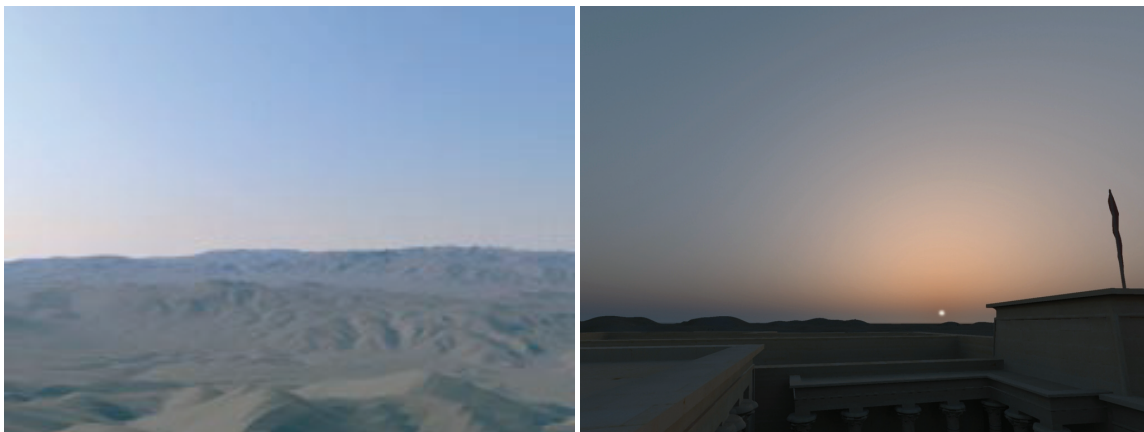
en échantillonnant la fonction de luminance du ciel le long des portions de sa ligne centrale qui ne sont pas obstruées par les objets de la scène. La fonction de luminance du ciel qui est échantillonnée est la fonction standard de luminance du ciel proposée en 1973 par la Commission Internationale de l'Éclairage. Les auteurs évoquent bien les interrélaxions entre les surfaces diffuses de la scène mais l'ordre maximum des réflexions considérées semble être de un pour les scènes en extérieur, tandis que de multiples rebonds sont illustrés dans le cas d'une pièce illuminée à travers sa fenêtre. De plus, seule l'intensité lumineuse du dôme est évaluée, jamais son spectre, ne permettant pas de rendre correctement la couleur du ciel ni de prendre en compte la hauteur du soleil dans le ciel. Enfin, la diffusion atmosphérique ambiante au sein de la scène n'est pas simulée. À la place, l'intensité lumineuse émise par une bande du dôme est plus simplement considérée comme ponctuelle et décroissant selon le carré de la distance entre le point d'observation et le centre de la bande.





KANEDA *et al.* [76] réutilisent le principe du dôme lumineux et l'adaptent au rendu spectral initié par KLASSEN [79]. Pour évaluer l'illumination globale de la scène, le principe du dôme lumineux de NISHITA *et al.* [109] est réutilisé, mais au lieu d'être calculée grâce à la fonction de luminance de la CIE, la radiance du dôme est évaluée de manière similaire à KLASSEN [79], *id est* par intégration de l'intensité lumineuse selon l'approximation de diffusion unique (*single scattering*). De plus, les couches homogènes de molécules d'air et de brume sèche sont remplacées par des fonctions qui décroissent exponentiellement avec l'altitude.

**Modèles analytiques** Plutôt que de recalculer l'illumination atmosphérique globale à chaque image et à chaque changement de conditions atmosphériques, PEREZ *et al.* [117] définissent un modèle atmosphérique générique et analytique décrivant la luminance du ciel perçue par un observateur en fonction de sa latitude, de sa longitude, de la position du soleil et de la densité nuageuse ambiante. Leur modèle est obtenu en appliquant une méthode de régression sur les relevés des conditions d'ensoleillement réalisés par la Commission Internationale de l'Éclairage.



**FIGURE 1.14.** – Rendus atmosphériques avec modèles analytiques par PREETHAM *et al.* [120] (gauche) et HOSEK *et al.* [63] (droite).

Mais la luminance ne donne pas d'information colorimétrique. C'est pourquoi PREETHAM *et al.* [120] simulent la diffusion atmosphérique en suivant la méthode spectrale de NISHITA *et al.* [111] (figure 1.14). Des simulations de diffusion atmosphérique sont calculées dans de nombreuses configurations d'illumination et une méthode de régression est appliquée aux résultats afin de définir un modèle analytique.

HOSEK *et al.* [63] proposent une variante au modèle analytique d'illumination globale de PREETHAM *et al.* [120] (figure 1.14). Ils implémentent leur propre simulation de diffusion atmosphérique par lancer de rayon en ajoutant notamment la prise en compte de l'albédo du terrain.

D'autres travaux font varier les conditions d'illuminations de la planète. Ainsi WILKIE *et al.* [144] considèrent un ou plusieurs soleils avec des spectres lumineux différents, tandis que COLLIENNE *et al.* [21] s'intéressent à des planètes dont la composition atmosphérique diffère telles que Mars par exemple.



**Méthode de la radiance** RUSHMEIER *et al.* [125] généralisent la méthode de la radiance [50] (*radiosity*) – qui est à l'origine de la scène très connue de la boîte de Cornell – pour l'appliquer aux milieux participatifs en s'inspirant de la méthode zonale (*zonal method*) provenant de l'étude des transferts de chaleur. La méthode de la radiance consiste à estimer l'illumination diffuse globale d'une scène en calculant les échanges entre l'ensemble des éléments de surface jusqu'à un ordre de diffusion donné. Les aires et la distance entre deux éléments de surface permettent d'exprimer le ratio de radiance échangée appelé facteur de forme. Pour appliquer cette méthode aux milieux participatifs, les auteurs

ajoutent deux nouveaux types d'échanges : les échanges entre volumes, et entre volumes et surfaces, ainsi que les facteurs de forme associés. Selon la méthode zonale, le volume participatif est discrétisé en cellules ou *zones* dont les coefficients d'émission, d'absorption et de diffusion sont homogènes, et les fonctions de phase associées sont isotropes.

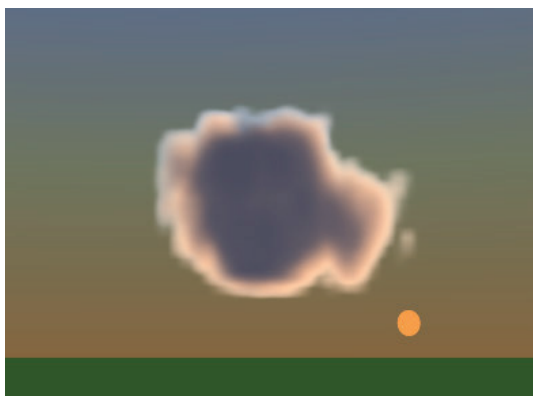


NISHITA *et al.* [107] proposent une méthode de rendu de nuages prenant en compte les multiples diffusions (*multiple scattering*) de la lumière ainsi que l'éclairage global par le soleil, le ciel et la surface. Comme dans la méthode de la radiance [50] (*radiosity*), les auteurs discrétisent le milieu participatif en voxels avant de leur associer des facteurs de forme. En revanche, à la différence de la méthode initiale de la radiance,

la diffusion de la lumière par le milieu n'est pas restreinte à être uniforme, mais est généralisée à des fonctions de phase hautement anisotropes. Pour ce faire, l'influence de la fonction de phase au deuxième et au troisième ordre de diffusion est incorporée aux facteurs de forme dont une carte est pré-calculée en faisant l'approximation d'une densité moyenne, *id est* d'un milieu participatif homogène. Depuis un voxel d'origine, il existe une infinité de voxels atteignables par des chemins composés de deux ou trois rebonds et dont les facteurs de forme sont non nuls. Aussi, un seuillage est appliqué afin de restreindre la carte aux facteurs de forme conservant le plus d'énergie et qui serviront ensuite de masque de pondération pour intégrer les participations lumineuses au deuxième et au troisième ordre de diffusion. Pour le premier ordre de diffusion, les auteurs observent que les nuages sont évidemment éclairés par le soleil mais que le ciel et le terrain participent aussi à l'éclairage global de manière non négligeable. Le calcul des intensités lumineuses du ciel, du disque solaire et de la surface en raison de la diffusion atmosphériques ne sont pas détaillés, les auteurs renvoyant à l'article « Display of the Earth Taking into Account Atmospheric Scattering » [110].

L'éclairage global est estimé une fois pour l'ensemble des voxels nuageux en calculant les angles solides du disque solaire, du ciel et de la surface tels que perçus au centre du milieu nuageux. Ensuite, pour chaque voxel, la valeur de transmittance moyenne dans l'angle solide de chaque source d'éclairage est pré-intégrée.

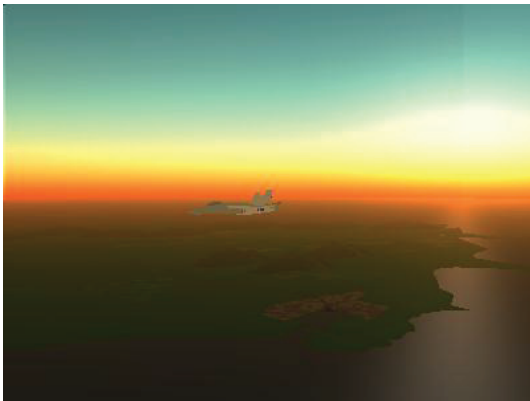
HABER *et al.* [58] s'intéressent au problème de la diffusion atmosphérique, non seulement lorsque le ciel est haut au-dessus de l'horizon, mais aussi dans les conditions de pénombre qui entourent le coucher et le lever de soleil. Pour rendre correctement le ciel depuis un point de vue à la surface dans de telles conditions de pénombre, il est nécessaire de simuler de multiples diffusions (*multiple scattering*) atmosphériques, contrairement à l'approximation de diffusion unique (*single scattering*). Pour ce faire, les auteurs s'inspirent de la méthode de la radiance (*radiosity*) appliquée aux volumes participatifs [125], et discrétisent l'atmosphère en couches de densité constante et en une grille dont les cellules stockent l'information de radiance reçue. La radiance à l'ordre zéro est stockée séparément suivant le type de particules qui la captent car les fonctions de phase des molécules d'air et des aérosols sont différentes, tandis que les ordres supérieurs de la radiance sont combinés. La radiance est initialisée pour chaque cellule par lancer de rayon en prenant en compte l'effet de réfraction dû à la variation de densité atmosphérique (voir Section 1.1.3 page 27), puis de multiples passes de diffusion entre chaque paire de cellules sont calculées, en considérant que la surface ne réfléchit pas de lumière. Pour calculer l'ordre suivant d'illumination d'une cellule, chacune des autres cellules contribue comme une source lumineuse proportionnellement à la radiance perçue à la passe précédente, et à la transmittance et au facteur de forme entre les deux cellules. Pour la première passe de diffusion, la radiance perçue est diffusée de manière anisotrope et séparément pour chaque type de particules, tandis que les passes suivantes de diffusion sont toutes isotropes. Une fois un nombre voulu de passes atteint, une texture du ciel est construite et plaquée sur un dôme pour rendre la scène en temps réel.



**Méthode des ordonnées discrètes** MAX [96] propose une approximation de la méthode des ordonnées discrètes [18] (*discrete ordinates*) évoquée par KAJIYA *et al.* [74] afin de simuler de multiples diffusions (*multiple scattering*) de la lumière dans les nuages. La méthode consiste à calculer la propagation de la lumière entre voxels nuageux voisins et dans un nombre fini de classes de directions (*directions bins*) d'après un schéma de pondération simplifié. La fonction de phase de HENYEV-GREENSTEIN est quantifiée

pour chaque paire de classes de directions. L'énergie lumineuse est initialisée par une passe de lancer de rayon depuis la source à travers le milieu sans déviation, mais en accumulant simplement la transmittance le long de chaque rayon et l'intensité perçue au niveau de chaque voxel. Ensuite, des vagues successives de propagation sont calculées dans chacune des classes de directions et le flux lumineux total est accumulé dans chaque classe de directions pour chaque voxel. Chaque vague correspond à un ordre de diffusion et réduit l'énergie lumineuse proportionnellement à l'albédo du milieu. Même dans le cas d'un albédo unitaire qui correspond à l'absence d'absorption, l'énergie lumineuse n'est pas conservée car une grande partie est perdue lorsque chaque vague de propagation atteint les limites du volume nuageux et que MAX ne considère aucun rebond en dehors du milieu. Le pré-calcul de la diffusion

converge donc rapidement. Au rendu final, les contributions des ordres zéro et un de diffusion sont calculées précisément pour chaque rayon tandis que les ordres supérieurs sont estimés d'après les flux des classes de directions des voxels traversés.



NISHITA *et al.* [111] appliquent une méthode similaire à MAX [96] dans le cas du rendu de l'atmosphère privée de nuages. Les auteurs calculent la diffusion unique (*single scattering*) de manière optimisée similairement à leurs travaux précédents [110] qui tirent profit des symétries du problème et précalculent notamment une table de transmittance. De multiples diffusions (*multiple scattering*) sont ensuite calculées pour un ensemble de voxels selon un nombre fini de directions.

## Méthodes de MONTE CARLO

Les méthodes de MONTE CARLO consistent à échantillonner stochastiquement l'espace des chemins (*path tracing*) parcourus par un très grand nombre de rayons lumineux dans la scène afin d'évaluer une solution à l'équation de rendu de KAJIYA [73], *id est* l'illumination globale perçue en chaque pixel de l'image finale. Un gros intérêt de ce type de méthodes est que l'erreur du résultat diminue avec le nombre d'échantillons pour converger vers zéro lorsque le nombre d'échantillons tend vers l'infini. Il est donc possible d'obtenir un aperçu rapide du résultat et de continuer les calculs pour s'approcher progressivement de la solution exacte. En revanche, le gros problème de ce type de rendu est de produire une image bruitée et de nécessiter beaucoup d'échantillons, et donc de temps de calcul, avant que le bruit ne deviennent imperceptible.

BLASI *et al.* [8] sont parmi les premiers en informatique à appliquer la méthode de MONTE CARLO au rendu atmosphérique en adaptant la méthode de KAJIYA *et al.* [74]. Les deux passes de lancer de rayon sont conservées mais la première passe ne se limite plus à l'approximation de la diffusion unique (*single scattering*). Elle évalue plutôt les multiples diffusions (*multiple scattering*) grâce à l'échantillonnage de chemin par la méthode de MONTE CARLO. La traversée du milieu par les rayons issus des sources lumineuses est faite par pas successifs uniformes (*ray marching*). Le complément à un de la transmittance associée à chaque pas donne la probabilité que le rayon soit dévié par un tirage aléatoire. Lorsqu'il y a déviation, la nouvelle direction du rayon est déterminée en échantillonnant par importance la fonction de phase du milieu et l'intensité lumineuse du rayon est accumulée dans le voxel où s'est produit la déviation. La fonction de phase de base utilisée par les auteurs est très proche de celle de HENY-STEIN. Elle dépend aussi d'un facteur d'anisotropie, mais son échantillonnage préférentiel s'exprime de manière analytique. La fonction de phase complète utilisée pour les rendus est la somme d'une composante isotrope et d'une composante anisotrope à dominante avant (*forward scattering*). Pour la seconde passe, un rayon est lancé pour chaque pixel d'après les conditions d'observation, et intègre la transmittance et l'intensité lumineuse au passage de chaque voxel qu'il traverse, puis récupère l'intensité de l'illumination globale à la sortie du milieu.





Novák *et al.* [112] présentent deux méthodes spécifiques au calcul de la transmittance dans un milieu participatif hautement hétérogène tel que l'atmosphère et les nuages : le *ratio tracking* et le *residual tracking*. Ces deux méthodes sont des variantes de l'algorithme de *delta tracking* qui permettent de limiter le coût et la variance du calcul de la transmittance par échantillonnage de MONTE CARLO. À la différence du *delta tracking*

classique qui estime de manière binaire si le rayon passe en absence de collision avec le milieu, ou pas du tout dès la première collision, le *ratio tracking* estime la probabilité que le rayon passe l'ensemble des événements stochastiques de collision sur tout l'intervalle où le rayon intersecte le milieu, et renvoie donc une estimation continue de la transmittance. Un gros avantage de cette méthode est aussi de permettre l'estimation de la transmittance spectrale pour chaque rayon, au lieu d'échantillonner le spectre en lançant des rayons monochromes. Entre l'air et les nuages, c'est principalement l'air dont la diffusion présente une variation spectrale, et qui colore la lumière. Les nuages diffusent uniformément tout le spectre, à moins de considérer les variations spectrales haute fréquence qui s'observent dans la fonction de phase de MIE pour de grosses gouttelettes. Le *residual tracking* applique le principe des *control variates* qui consiste quant à lui à séparer la fonction de densité du milieu hétérogène en deux : une fonction appelée fonction de contrôle et qui est constante ou assez simple pour être intégrée analytiquement, et un résidu hétérogène qui reste intégré par échantillonnage de MONTE CARLO. Si le principe des *control variates* est bien connu pour limiter la variance d'un estimateur de MONTE CARLO, un point intéressant que les auteurs soulignent est qu'il est possible de surestimer la fonction de contrôle, entraînant des valeurs négatives pour la portion résiduelle de la densité. Bien que la densité ne puisse physiquement être négative, ce n'est qu'un artifice de calcul et l'estimateur de MONTE CARLO reste valide : la transmittance est sous-estimée par l'intégration de la fonction de contrôle mais les zones à densité résiduelle négative compensent en augmentant la transmittance.



KUTZ *et al.* [82] expliquent comment échantillonner non seulement la transmittance mais aussi les chemins dans un milieu participatif hétérogène à l'aide de deux nouvelles variantes du *delta tracking* : le *decomposition tracking* et le *spectral tracking*. L'algorithme de *decomposition tracking* met aussi en œuvre les *control variates* et associe stochastiquement les événements de

collision à la fonction de contrôle de la densité, ou à son résidu hétérogène. Ainsi lorsque la collision est associée à la fonction de contrôle, il n'est pas nécessaire d'interroger la fonction de densité complète, ce qui limite le coût du rendu. L'algorithme de *spectral tracking* permet quant à lui d'échantillonner l'espace des chemins d'un milieu participatif sur toutes les longueurs d'onde souhaitées à la fois, au lieu de lancer un rayon monochrome par longueur d'onde. C'est un échantillonnage préférentiel du mélange spectral (*mixture importance sampling*) : chaque longueur d'onde considérée correspond à une stratégie possible pour échantillonner l'espace des chemins, mais une seule est choisie stochastiquement, et la contribution spectrale est pondérée en conséquence. Il est aussi intéressant de noter que les deux variantes peuvent être combinées en séparant la densité de probabilité spectrale en une partie commune qualifiable de blanche, et en un résidu spectral.



KALLWEIT *et al.* [75] mettent en pratique des techniques issues du domaine de l'apprentissage par réseaux de neurones dans le cadre du rendu de nuage par méthode de MONTE CARLO. Plutôt que d'évaluer stochastiquement la radiance après de multiples diffusions (*multiple scattering*), cette évaluation est remplacée par une requête à un réseau de neurones entraîné sur un ensemble varié de directions d'éclairage et de

couvertures nuageuses rendues au préalable par méthode de MONTE CARLO. L'éclairage dû à la diffusion unique (*single scattering*) reste quant à lui évalué par méthode de MONTE CARLO en utilisant l'algorithme du *delta tracking*. Cette méthode introduit un léger biais au résultat (indiscernable dans les exemples donnés), mais permet aux auteurs de gagner jusqu'à trois ordres de grandeurs en vitesse de convergence.

## Rendu atmosphérique temps réel

Si le rendu atmosphérique est en soi un problème qualitatif, avec l'évolution de la puissance de calcul, les méthodes les plus anciennes et celles faisant le plus d'approximations peuvent aujourd'hui s'exécuter en temps interactif, voire en temps réel. De telles performances constituent l'objectif de certains travaux plus récents. Pour y parvenir, le rendu est optimisé en réalisant d'une part des simplifications supplémentaires, et d'autre part en déportant un maximum de calculs dans une passe préalable dont les résultats sont réutilisés à bas coût.

**Illumination procédurale** GARDNER [41] définit la transmittance de la texture nuageuse comme une série de FOURIER composée de moins de quelques harmoniques sinusoïdales aléatoirement déphasées. Cette texture nuageuse est plaquée sur un plan horizontal pour modéliser les nuages en haute altitude et les couvertures nuageuses planes. Pour les nuages les plus proches de la surface ou présentant un important développement vertical, la texture nuageuse est appliquée aux surfaces d'un ensemble de primitives ellipsoïdales et détermine leur transmittance tandis que l'ombrage est simplement calculé à la surface de l'ellipsoïde. Afin d'éviter que les bords des ellipsoïdes soient trop visibles, un profil décroissant est appliqué à la texture nuageuse et augmente la transmittance de manière inversement proportionnelle à la distance du bord de l'ellipse dans le plan image. Pour modéliser des structures nuageuses complexes, plusieurs ellipsoïdes qui se chevauchent sont combinés et leurs intersections sont calculées et ordonnées pour déterminer comment combiner leurs transmittances.



**Imposteurs et fonctions de mélange** HARRIS *et al.* [59] s'intéressent au rendu temps-réel de nuages. Afin d'atteindre de telles performances, l'illumination atmosphérique globale n'est pas simulée, mais est remplacée par des sources lumineuses de couleurs, d'intensités et de positions choisies manuellement. Les nuages sont pré-calculés en amont sous la forme de particules sphériques statiques, puis le rendu est réalisé en deux passes exploitant les capacités de

mélange sur GPU. La première passe pré-calculé la quantité de lumière qui atteint les particules depuis le point de vue de chaque source lumineuse, en intégrant la transmittance et la fonction de phase de la diffusion de RAYLEIGH par mélanges successifs dans une texture. Cette première phase ressemble ainsi au principe des cartes d'ombres (*shadow maps*). La seconde passe est recalculée à la volée lors du rendu temps réel et procède de manière similaire en intégrant la transmittance des particules et leurs intensités lumineuses pré-calculées par mélanges successifs depuis le point de vue de la caméra. Afin d'accélérer encore plus le rendu, les auteurs construisent des imposteurs de nuages. Des cadres sont placés dans le champ de la caméra et des textures transparentes correspondant à la portion de rendu des nuages environnants sont plaquées sur les cadres. Les imposteurs sont recalculés en fonction du mouvement de la caméra, les imposteurs les plus proches nécessitant d'être mis à jour plus souvent. Dans le cas où un objet ou la caméra traverse un nuage, l'imposteur associé est scindé en couches séparant approximativement l'avant et l'arrière de l'objet. Bien que l'utilisation d'imposteurs permette aux auteurs d'atteindre des performances temps réel pour des centaines de milliers de particules nuageuses, aucune information n'est donnée sur le nombre d'imposteurs créés, le nombre de particules par imposteurs ou encore sur la méthode utilisée pour choisir les particules à grouper dans un même imposteur. L'utilisation de particules nuageuses sphériques permet aux auteurs d'obtenir des nuages cumuliformes, mais les autres types nuageux tels que les nuages stratiformes observés sur certains de ces mêmes rendus, ne sont pas évoqués.



**Placage de photons** En 2012, ELEK *et al.* [35] présentent une méthode de rendu interactif de nuage fondée sur le plaquage de photons. Le domaine géométrique du nuage est discrétisé en une grille régulière au sein de laquelle sont accumulées les informations de la carte de photons : un facteur d'anisotropie, un triplet RVB, et un facteur d'énergie. De nombreux photons sont lancés dans le volume depuis la source directionnelle solaire et sont suivis au cours de

leurs déplacements stochastiques afin de construire progressivement la carte de photons. L'échantillonnage préférentiel de la distance de libre parcours (*free path*) en fonction de la densité nuageuse est réalisé par l'algorithme de *delta tracking*, tandis que l'échantillonnage préférentiel de la fonction de phase de HENYEV-GREENSTEIN donne la déviation propre à chaque événement de diffusion. Lors de chacun de ces événements de diffusion, la carte de photons est mise à jour : l'anisotropie augmente proportionnellement à la déviation du photon vis-à-vis de la direction initiale du champ solaire, tandis que l'énergie portée par le photon diminue le long de son trajet proportionnellement à la transmittance. Le rendu

final est réalisé par *ray marching* à travers le champ de densité du nuage et la carte de photons depuis la caméra. L'intégration de la densité donne la transmittance et le stimulus lumineux est obtenu par autoconvolution analytique de la fonction de HENY-STEIN en choisissant le premier lobe dans la direction d'observation et le second dans la direction de l'éclairage solaire et selon le facteur d'anisotropie lu dans la carte de photons. Bien que la carte de photons soit grossière, elle est suréchantillonnée par un filtrage bilatéral joint guidé par le champ de densité du nuage, ce qui permet de prendre en compte les détails les plus fins du nuage tout en limitant la variance de la carte de photons. Enfin, les auteurs exploitent la cohérence temporelle grâce à un cache circulaire permettant d'amortir le coût de construction de la carte de photons en l'étalant dans le domaine temporel. Plutôt que de reconstruire entièrement la carte de photons à chaque image d'une séquence, seule une portion est ainsi invalidée et mise à jour.



### Précalcul de multiples ordres de diffusion

BOUTHORS *et al.* [14] proposent une méthode de rendu temps réel prenant en compte de multiples diffusions (*multiple scattering*) au sein d'une couche nuageuse horizontale infinie et homogène. Ils séparent le traitement des trois premiers ordres de diffusion et regroupent les ordres de diffusion plus élevés dans un terme diffus. De plus, les auteurs séparent les événements de diffusion avant (*forward scattering*) des autres car la fonction de phase de MIE possède un pic avant extrêmement prononcé. La déviation lors d'un événement de diffusion avant est alors négligée et la réduction d'intensité associée est incorporée à l'intégration d'extinction. Les interréllections diffuses entre la base nuageuse et la surface sont estimées sur GPU par la méthode de la radiance (*radiosity*).



BRUNETON *et al.* [17] poursuivent sur la voie de NISHITA *et al.* [110] et HABER *et al.* [58] en proposant une méthode de rendu interactif de paysage terrestre prenant en compte de multiples diffusions (*multiple scattering*) atmosphériques. L'innovation majeure de leurs travaux est de ne considérer la fonction de hauteur de la surface du globe qu'aux calculs de radiance à l'ordre zéro et à l'ordre un qui sont réalisés au rendu, permettant ainsi aux montagnes de projeter une ombre et les effets de trouées de lumière. La contribution à l'illumination globale des ordres supérieurs de diffusion est quant à elle précalculée en ignorant le relief, afin d'exploiter les symétries de la scène et de limiter les dimensions des tables précalculées, de manière similaire à NISHITA *et al.* [110]. La fonction de transmittance est précalculée, puis les ordres successifs de diffusion sont intégrés sur l'ensemble de l'hémisphère à la surface du terrain qui est pris en compte comme une surface diffuse et sur l'ensemble des directions dans le volume de l'atmosphère selon les diffusions de MIE et de RAYLEIGH. Le résultat est un rendu interactif avec trouées de lumière et approximation des multiples diffusions (*multiple scattering*) atmosphériques.





BOUTHORS *et al.* [15] proposent une autre méthode de rendu interactif prenant en compte multiples diffusions (*multiple scattering*), mais dans le cas particulier des nuages. Leur idée est de déterminer les principales zones du nuage, appelées collecteurs, qui contribuent à diffuser la lumière jusqu'à l'observateur. Ils précalculent la fonction de phase de MIE pour une distribution de taille de goutte choisie comme dans leur publication précédente [14]. Ils simulent ensuite

le transport de la lumière par méthode de MONTE CARLO, en précalculant la fonction de phase pour différents ordres de diffusion, au sein de dalles nuageuses homogènes (*slabs*) selon plusieurs paramètres : l'épaisseur de la dalle, la profondeur de l'observateur, l'angle d'observation, et l'angle d'élévation de la source lumineuse. Les résultats de la simulation encodent la transmittance, le centre et la largeur du collecteur, et permettent de définir ce que les auteurs appellent la *fonction canonique de transport*. La surface du nuage est représentée par un maillage grossier, agrémenté d'une hypertexture de bruit pour ajouter du détail tandis que le centre du nuage est considéré comme homogène. Au rendu, le maillage est utilisé pour calculer la carte de profondeur et de normales tandis qu'un processus itératif utilise ensuite la fonction canonique de transport pour trouver les collecteurs des différents ordres de diffusion et évaluer la transmittance.

Yusov [155] élabore une méthode de rendu temps réel de nuages en précalculant la transparence et la radiance après une seule ou de multiples diffusions (*multiple scattering*) de la lumière dans une particule nuageuse de référence, quelles que soient les directions d'éclairage et d'observation et la position de la caméra (figure 1.15). De multiples copies de la particule de référence sont instanciées à différentes échelles, positions et orientations afin de constituer une couverture nuageuse cumuliforme. Afin de réaliser le rendu de la couche nuageuse ainsi constituée, un rayon est lancé par pixel. Sa radiance est exprimée comme la somme entre les différents ordres de diffusion de la lumière calculé différemment : des contributions et les intersections entre ce rayon et les particules sont calculées et ordonnées par profondeur. À chaque intersection correspond une valeur pré-calculée de profondeur optique qui est multipliée par la densité de chaque instance afin d'obtenir la transmittance. La radiance est évaluée aussi bien dans le cas d'une diffusion unique (*single scattering*) que dans le cas de multiples diffusions (*multiple scattering*). Les ombres projetées sur une instance de la particule par les autres instances sont évaluées par la méthode de la *shadow map* : une liste ordonnée de particules est construite depuis le point de vue de la source lumineuse et l'occlusion est évaluée pour chaque particule visible.



FIGURE 1.15. – Nuages rendus par Yusov [155].

### 1.2.3 Synthèse

**TABLE 1.1.** – Synthèse des différentes méthodes de rendu atmosphérique : si elles sont appliquées au rendu de l’atmosphère, de nuages, à de multiples diffusions, et au rendu spectral.

	Auteurs	Ref.	Année	Atm.	Nuages	MD	Spectral
Méthodes aux éléments finis	BLINN	[10]	1982		X		
	KAJIYA <i>et al.</i>	[74]	1984		X	X	
	MAX	[95]	1986		X		X
	KLASSEN	[79]	1987	X			X
	NISHITA <i>et al.</i>	[110]	1993	X	X		X
	KNISS <i>et al.</i>	[81]	2003		X		X
	RILEY <i>et al.</i>	[122]	2003	X	X	X	X
	RILEY <i>et al.</i>	[124]	2004	X	X	X	X
	NISHITA <i>et al.</i>	[109]	1986	X			
	KANEDA <i>et al.</i>	[76]	1991	X	X		X
	PEREZ <i>et al.</i>	[117]	1993	X			
	PREETHAM <i>et al.</i>	[120]	1999	X		X	X
	HOSEK <i>et al.</i>	[63]	2012	X		X	X
	WILKIE <i>et al.</i>	[144]	2013	X			X
	COLLIENNE <i>et al.</i>	[21]	2013	X		X	X
	RUSHMEIER <i>et al.</i>	[125]	1987			X	X
	NISHITA <i>et al.</i>	[107]	1996			X	X
HABER <i>et al.</i>	[58]	2005	X		X	X	
MAX	[96]	1994			X	X	
NISHITA <i>et al.</i>	[111]	1996	X		X	X	
M.C.	BLASI <i>et al.</i>	[8]	1993	X		X	X
	NOVÁK <i>et al.</i>	[112]	2014		X		X
	KUTZ <i>et al.</i>	[82]	2017		X	X	X
	KALLWEIT <i>et al.</i>	[75]	2017		X	X	X
Temps réel	GARDNER	[41]	1985		X		
	HARRIS <i>et al.</i>	[59]	2001		X		X
	ELEK <i>et al.</i>	[35]	2012		X	X	X
	BRUNETON <i>et al.</i>	[17]	2008	X		X	X
	BOUTHORS <i>et al.</i>	[14]	2006		X	X	X
	BOUTHORS <i>et al.</i>	[15]	2008		X	X	X
YUSOV	[155]	2014		X	X	X	

Comme le synthétise le tableau 1.1, la plupart des méthodes de rendu évoquées dans cet état de l’art se focalisent soit sur des nuages isolés, soit sur l’atmosphère privée de tout nuage. Seules les méthodes de KANEDA *et al.* [76], NISHITA *et al.* [110] et RILEY *et al.* [122, 124] s’intéressent au problème combiné du rendu atmosphérique en présence de nuages. De plus, depuis les travaux de KLASSEN [79], et à l’exception du modèle analytique de luminance de PEREZ *et al.* [117], toutes les méthodes considèrent le rendu spectral au moins sous la forme d’un tristimulus RGB. De manière similaire, les méthodes les plus récentes s’attachent toutes à traiter de multiples diffusions (*multiple scattering*), sauf les travaux de HARRIS *et al.* [59] et KNISS *et al.* [81] qui les négligent par soucis de performances et les travaux de NOVÁK *et al.* [112] et WILKIE *et al.* [144] car ils s’intéressent exclusivement à l’estimation de la transmittance et à la construction d’un dôme céleste extrasolaire.

## 1.3 Modélisation de nuages

Les techniques existantes de modélisation et d'animation de nuages volumiques définissent une fonction de densité qui caractérise le contenu en eau des nuages en tout point de l'espace. Elles peuvent être classées en deux catégories : les simulations, et les méthodes procédurales. Des simulations physiques sont utilisées dans de nombreux domaines scientifiques pour reproduire numériquement le fonctionnement du monde réel afin de prédire l'évolution d'un système à partir d'un état initial donné. Elles permettent de réaliser virtuellement des expériences trop complexes ou trop coûteuses, voire impossibles ou dangereuses à mettre en œuvre dans le monde réel ainsi que de prédire l'évolution du monde réel avant qu'elle ne se produise comme c'est le cas en météorologie. Dans le cadre de la modélisation de nuages, les simulations physiques reproduisent les principaux phénomènes responsables de la formation, du déplacement et des déformations des masses nuageuses grâce aux lois de la mécanique des fluides et de la thermodynamique. La différence fondamentale de la génération procédurale avec la simulation est qu'elle s'abstrait partiellement ou totalement de la chaîne causale pour produire plus directement des représentations convaincantes de ce qui serait un unique état d'une simulation.

Du fait de la nature même du processus itératif d'évolution causale, les simulations présentent l'avantage de produire des animations. Au contraire, les méthodes à base de génération procédurales ne s'intéressent pas nécessairement à l'animation, et génèrent souvent directement un état statique.

Un autre point crucial qui oppose les simulations et les méthodes à base de génération procédurale est le contrôle utilisateur. Une force de ce type de méthodes est qu'elles proposent souvent un contrôle important à l'utilisateur par l'intermédiaire d'esquisses qui guident la génération ou l'édition manuelle des structures à plus ou moins haut niveau de la représentation générée. À l'inverse, une fois que le processus itératif d'une simulation strictement physique est lancé, le système n'est plus contrôlable. Chaque état successif dépend de l'état précédent et des lois de la physique considérées. La seule chose qu'il est possible de contrôler directement est l'état initial. Certaines méthodes de simulation proposent néanmoins d'intégrer une certaine forme de contrôle en modifiant légèrement les lois physiques modélisées pour influencer l'évolution du système.

Le point en quelque sorte dual au contrôle utilisateur est le réalisme. Si les méthodes à base de génération procédurale font la part belle au contrôle et à l'expressivité de l'utilisateur, c'est au risque de sacrifier le réalisme et la cohérence de ce qui est généré qui sont entre les mains du dit utilisateur. Dans le cas des simulations, le réalisme de l'évolution obtenue ne dépend pas de l'utilisateur mais de la qualité des lois modélisées et, dans une moindre mesure, de la précision des opérations numériques associées.

### 1.3.1 Simulations physiques

**Fluides** La simulation de fluides est un champ de recherche populaire en informatique graphique car elle a des applications évidentes sous la forme d'effets spéciaux impliquant des éléments tels que la fumée, le feu, les explosions, l'eau sous sa forme liquide ou gazeuse, et l'air. Le système d'équations de NAVIER-STOKES est alors communément utilisé pour décrire

l'évolution de l'état du fluide. Lorsque les équations de NAVIER-STOKES sont formulées dans un référentiel fixe, le plus souvent en discrétisant le domaine en une grille régulière et en s'intéressant à l'évolution de l'état de chaque cellule, il est question de simulation eulérienne. Une variante consiste à remplacer les cellules eulériennes par un ensemble de particules et à exprimer les équations de NAVIER-STOKES dans le repère de chacune de ces particules. Ce sont les simulations dites lagrangiennes.

Suivant la nature du fluide, la simulation est plus ou moins complexe à définir. Ainsi, parmi les cas les plus simples, la simulation de l'eau ne nécessite généralement que de simuler son écoulement en négligeant notamment ses changements d'état. De plus, l'eau est considérée comme très proche d'être incompressible. En comparaison, la compressibilité de l'air doit par exemple être prise en compte lorsqu'il s'agit de simuler ses interactions avec des éléments se déplaçant à des vitesses supersoniques, ce qui est le cas des avions de chasse par exemple. Même si de telles simulations sont plutôt de l'ordre de l'aéronautique ou même de l'aérospatial, la simulation des nuages reste plus complexe que celle de l'eau ou de la fumée car elle met en jeu le cycle de l'eau et nécessite donc de s'intéresser aux changements d'état associés. En effet l'eau est un élément qui nécessite un apport d'énergie thermique pour s'évaporer. Cette énergie est emmagasinée et conservée par la vapeur d'eau jusqu'à ce qu'elle se recondense, c'est l'énergie ou température latente de l'eau, qui est libérée à la condensation en échauffant le milieu.

**Formation de nuages** Afin de reproduire la formation de nuages par simulation, il faut au minimum reproduire les phénomènes d'évaporation et de condensation dont les nuages sont une manifestation. Le phénomène d'évaporation permet tout d'abord d'injecter de la vapeur d'eau dans l'air qui, contrairement à une idée reçue, est alors invisible et ne constitue donc pas encore de nuage. Suivant les conditions de pression et de température, l'air peut contenir une certaine quantité de vapeur d'eau, c'est l'humidité ambiante. Au delà du seuil de saturation en eau appelé point de rosée, de l'énergie latente est libérée en même temps que le surplus d'humidité se recondense en microgouttelettes qui sont, elles, visibles et assez petites pour rester en suspension et ne pas précipiter.

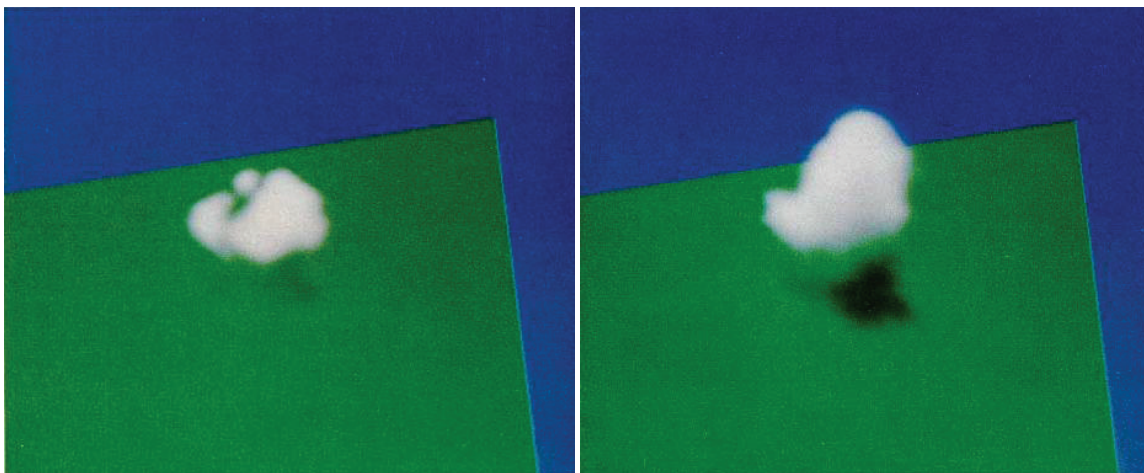
En plus de l'évaporation et de la condensation, il est aussi important de reproduire les changements de pression et les flux d'air, *id est* les vents. En effet, la diffusion de la vapeur d'eau dans l'air par le mouvement brownien de ses molécules est trop lente pour reproduire la formation soudaine des nuages et leurs formes observées dans l'atmosphère terrestre. Les nuages apparaissent plutôt lorsqu'une masse d'air humide et une masse d'air plus froide se rencontrent grâce au brassage plus violent des vents en général et de la convection. De plus, lorsqu'une part de vapeur se condense en nuage, la libération de chaleur latente favorise l'ascension par convection verticale.

Bien qu'il existe, notamment en météorologie, des modèles complexes de simulation physique des phénomènes responsables de la formation des nuages, ces simulations calculent généralement des moyennes de transferts de chaleur, d'indice d'ensoleillement et d'effet de serre par la couverture nuageuse ou des quantités de précipitations. Elles utilisent des représentations statistiques des différents types nuageux et ne calculent pas précisément leurs formes. Au contraire, en informatique graphique, la forme des nuages est cruciale afin de produire de belles et réalistes images de ciels nuageux. Depuis quelques années, certains travaux de météorologie font exception et s'intéressent tout de même aux formes précises que prennent les nuages, ce qui semble permettre de simuler plus précisément certains phénomènes. Bien qu'ils ne sont pas développés ici, ces travaux sont des pistes intéressantes

qui ouvrent la voie à d'éventuelles collaborations scientifiques entre l'informatique graphique et la météorologie.

## Simulations eulériennes

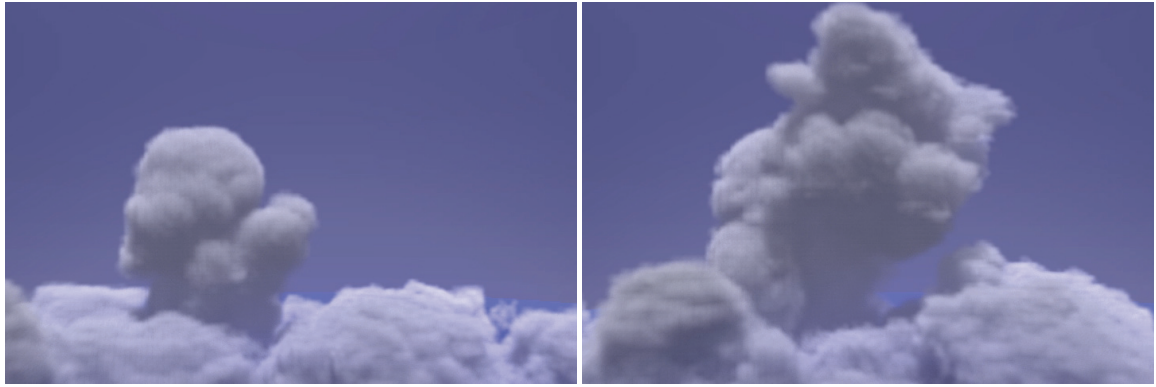
Plusieurs méthodes de simulation eulérienne ont été proposées en informatique graphique afin de reproduire la formation de nuages. KAJIYA *et al.* [74] sont les premiers en informatique graphique à proposer un modèle de simulation eulérienne de formation et évolution d'un nuage sur une grille eulérienne de  $10 \times 10 \times 20$  cellules (figure 1.16), où chaque pas de la simulation leur coûte 10s de temps CPU (figure 1.16). L'évaporation de l'eau est forcée par une source de chaleur externe et la convection et la libération de température latente à la condensation sont prises en compte dans le modèle tandis que la pression est définie comme constante à l'échelle considérée.



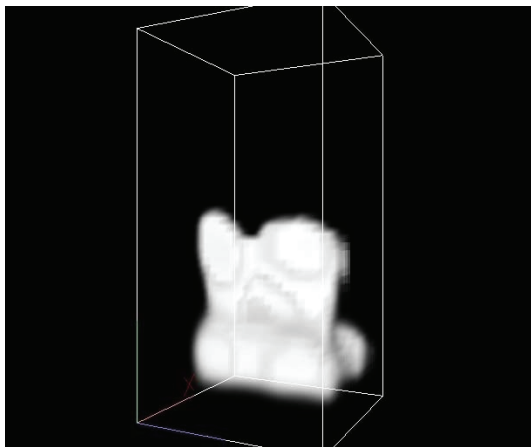
**FIGURE 1.16.** – Deux étapes de la simulation de *Cumulus* par KAJIYA *et al.* [74] sur une grille de  $10 \times 10 \times 20$  cellules.

MIYAZAKI *et al.* [98] font de même en ajoutant au modèle physique la prise en compte des changements d'altitude responsables du refroidissement adiabatique de l'air lorsqu'il s'élève dans l'atmosphère (figure 1.17 page suivante). Leur simulation intègre aussi la méthode de confinement de vortécité pour limiter la diffusion des vortex de la simulation eulérienne. La plus grande grille utilisée est constituée de  $150 \times 120 \times 100$  cellules, et est utilisée pour simuler la formation d'un *Cumulonimbus* (figure 1.17 page ci-contre).

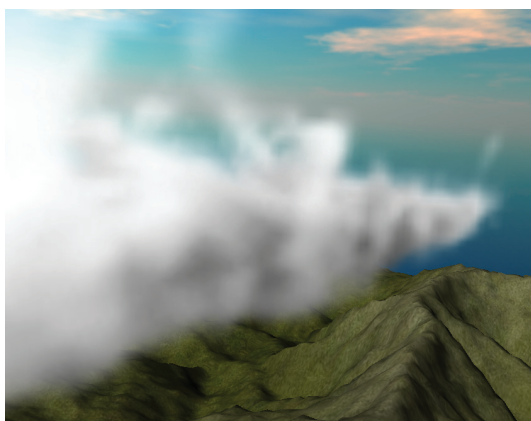




**FIGURE 1.17.** – Deux étapes de la simulation de *Cumulonimbus* par MIYAZAKI *et al.* [98] sur une grille de  $150 \times 120 \times 100$  cellules.



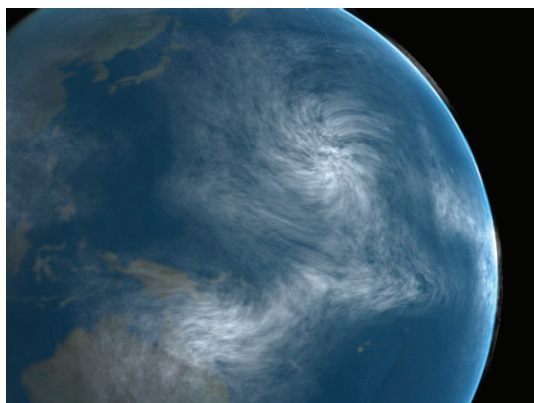
OVERBY *et al.* [114] proposent une simulation de formation de nuage qui prend aussi en compte le refroidissement adiabatique en considérant que la pression décroît exponentiellement par rapport à l'altitude (équilibre hydrostatique). Leur interprétation du phénomène de condensation ajoute la prise en compte d'une valeur de concentration en particules hygroscopiques qui favorisent la condensation de l'eau en servant de noyaux de condensation. Aucune information sur la définition ou l'advection du champ de densité de ces noyaux de condensation n'est cependant donnée. En utilisant directement l'expression exponentielle de la pression, la partie la plus complexe des équations de NAVIER-STOKES est résolue, ce qui simplifie grandement la simulation mais force le gradient de pression à être vertical et néglige les effets physiques qui pourraient causer des différentiels de pression horizontaux. Afin de compenser cette approximation, le champ de vitesse verticale est perturbé horizontalement en ajoutant une force centrifuge à l'ascension qui reproduit la dilatation d'air, et inversement, une force centripète à la descente pour reproduire la compression de l'air. La plus grande grille utilisée pour les résultats présentés est constituée de  $50 \times 15 \times 15$  cellules.



HARRIS *et al.* [60] publient de nouveaux travaux dont la contribution majeure est de porter la simulation de nuages sur GPU afin d'accélérer les calculs. À ce moment, la programmation GPU ne permet pas encore d'utiliser simplement des textures volumiques et il leur est alors nécessaire d'encoder la grille de simulation dans des textures planes. Leur implémentation GPU a le mérite supplémentaire de ne pas sacrifier la résolution de la contrainte d'incompressibilité par l'évaluation du gradient de pression et les performances de plusieurs techniques de réso-



lution de l'équation de Poisson sont comparées sur GPU. Les plus grandes grilles simulées comportent  $64 \times 32 \times 64$  cellules.



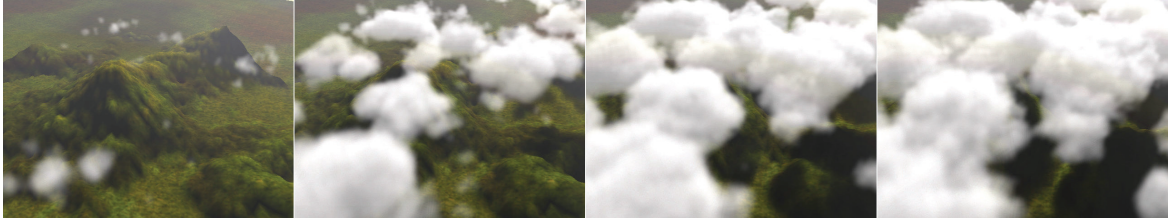
DOBASHI *et al.* [26] s'attaquent à la simulation de l'atmosphère à l'échelle planétaire. Pour ce faire, ils généralisent la grille eulérienne du domaine parallélépipédique au domaine de la coque sphérique que constitue l'atmosphère. La coque sphérique est décomposée uniformément sur les intervalles de coordonnées sphériques en  $160 \times 80 \times 4$  cellules. Afin de prendre en compte la rotation du référentiel terrestre, la force fantôme de CORIOLIS est ajoutée à la simulation. Les champs de différents coefficients physiques

sont définis à la surface sous forme de cartes : les coefficients de pression, de friction, de température et d'évaporation. Le champ de pression est défini comme la combinaison de primitives implicites qui représentent les zones de surpression et de dépression et qui sont animées par l'utilisateur indépendamment de la simulation. Les équations de NAVIER-STOKES sont alors résolues en injectant les forces extérieures de Coriolis, du champ de pression utilisateur ainsi que de friction en notant que cette dernière s'oppose proportionnellement à l'advection. Les autres champs de température et d'évaporation permettent quant à eux de contrôler l'injection de vapeur d'eau. Il est important de noter qu'une simulation simple de précipitations est réalisée. Une portion des nuages proportionnelle à leur densité précipite, et une partie de ces précipitations s'évapore avant d'atteindre la surface qui absorbe le reste, permettant de maintenir un budget constant d'eau dans l'atmosphère. Le ruissellement n'est cependant pas pris en compte, et l'eau absorbée par la surface rejoint simplement la réserve globale pour l'évaporation depuis la surface. Concernant les conditions thermiques, la température est définie comme décroissante proportionnellement à l'altitude et les phénomènes de convection thermique et adiabatique sont négligés, les déplacements verticaux étant donc uniquement générés par le champ de pression utilisateur.

## Automates cellulaires

Les publications subséquentes de DOBASHI *et al.* [22, 29] utilisent le cadre des automates cellulaires afin de permettre une simulation simplifiée mais hautement parallélisable de la formation et l'évolution de nuages (figure 1.18 page suivante). L'état de chaque cellule est encodé par trois variables booléennes indiquant si assez de vapeur est présente pour qu'il y ait condensation, si le changement d'état peut se produire, et la présence courante de nuage. À chaque pas de temps, ces variables d'états sont successivement mises à jour d'après des règles simples d'apparition de nuage, de disparition de nuage, et d'advection. L'apparition de nuage se produit si la cellule contient de la vapeur et qu'un certain motif de voisinage favorisant l'élévation et la croissance verticale est apte au changement d'état. La disparition de nuage et l'apparition d'humidité ainsi que la possibilité de changement d'état se produisent aléatoirement d'après des champs de probabilité modélisés par des sommes de primitives implicites ellipsoïdales. Quant à l'advection, les valeurs des trois variables booléennes sont simplement translatées entre cellules d'après la direction du vent qui varie selon l'altitude mais souffle toujours dans le premier axe horizontal de la grille. Un champ de densité lisse est construit à l'évaluation comme la combinaison de primitives implicites ponctuelles avec profil

de WYVILL *et al.* [151] (voir équation 2.13 page 71). La densité maximale de chaque primitive est calculée comme la moyenne des valeurs booléennes sur un voisinage spatio-temporel restreint. Les centres des primitives coïncident avec ceux des cellules et leur rayon unique est choisi de manière à couvrir plusieurs cellules. Les plus grandes grilles présentées dans le cadre de ces automates cellulaires font  $256 \times 256 \times 20$  cellules pour environ 160 kB de mémoire (figure 1.18).



**FIGURE 1.18.** – Cinq étapes de la simulation de *Cumulus* par DOBASHI *et al.* [29] sur une grille de  $256 \times 256 \times 20$  cellules.



Miyazaki *et al.* [99] publient une amélioration de la simulation de nuages par automates cellulaires, remplaçant les automates cellulaires par des *Coupled Map Lattices (CML)*. La principale différence est que les trois valeurs booléennes sont remplacées par des valeurs flottantes encodant un vecteur vitesse, la densité en vapeur d'eau, la densité nuageuse et la température de chaque cellule. Quant aux simples règles booléennes de changement d'état, elles sont remplacées par une adaptation des équations de NAVIER-STOKES sur le voisinage local de chaque cellule. Les plus grandes grilles de simulation pour lesquelles des résultats sont présentés sont constituées de  $256 \times 256 \times 40$  cellules, soit juste deux fois plus que dans la méthode précédente qui utilisait des automates cellulaires.

## Simulations lagrangiennes

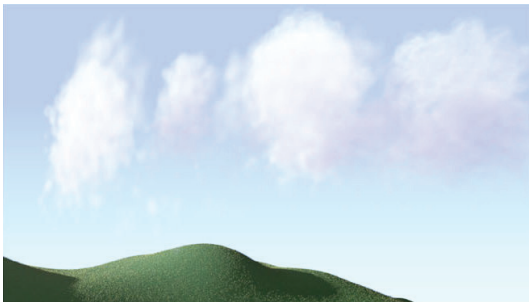
Si les simulations de nuages eulériennes sont les plus communes, elles présentent des défauts importants. En effet, le référentiel statique est partagé par l'ensemble des cellules et sa forme ne peut donc pas être adaptée dynamiquement. Au contraire, le domaine des simulations lagrangiennes à base de particules est totalement dynamique car il évolue directement avec les particules simulées et il est même possible de définir des règles adaptatives de subdivision et de fusion des particules en fonction de l'énergie cinétique locale par exemple.

Pour ces raisons, les simulations lagrangiennes connaissent un certain succès en informatique graphique, notamment dans la simulation d'eau, et quelques méthodes ont récemment été proposées dans le cadre de la simulation de nuages.



Bien que cela ne corresponde pas à une pure méthode lagrangienne, NEYRET [105] propose une méthode originale consistant à modéliser des nuages comme des amas de particules qu'il appelle des bulles et qui représentent chacune une parcelle nuageuse de température donnée. Pour éviter d'animer toutes les bulles appartenant à la partie intérieure et invisible des nuages, seules les bulles pour lesquelles il existe un important gradient de température sont animées. Le déplacement des bulles est défini comme la somme du gradient local de température et de la convection due notamment à la libération de chaleur latente par le phénomène de condensation. Afin de créer plus de mouvement et de faire croître les amas,

de nouvelles bulles sont créées à une altitude de condensation prédéfinie, *id est* au niveau de la base nuageuse, à la verticale des points chauds de la surface. D'autres primitives sont aussi créées dans les zones à fort différentiel thermique, mais les conditions qui régissent la quantité de primitives ainsi créées ne sont pas claires. Afin de limiter la quantité de bulles et de simplifier la représentation, les bulles dont le support est inclus par d'autres bulles sont supprimées et leur énergie thermique est redistribuée. Enfin, des détails animés sont produits en créant récursivement des sous-bulles plus petites qui roulent sur la surface de leurs mères.

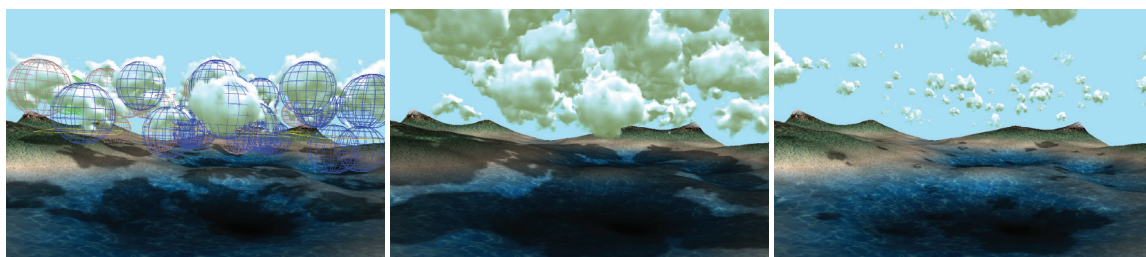


Plus récemment, BARBOSA *et al.* [5] proposent d'appliquer à la simulation de nuages les travaux de MACKLIN *et al.* [93] sur la simulation de fluides à base de contraintes sur les positions d'un système de particules qui assurent l'incompressibilité, et les travaux de ADAMS *et al.* [1] sur l'ajout-suppression adaptatif de particules dans les zones d'intérêt. Ils simulent ainsi sur GPU la formation d'une petite zone nuageuse

avec un système comprenant jusqu'à 930 000 particules, où chaque pas de temps nécessite environ pour 1 s et montrent des résultats obtenus après 500 ou 700 itérations de la simulation.

GOSWAMI *et al.* [51] proposent une méthode de simulation lagrangienne de type SPH dans laquelle quelques particules (jusqu'à 85 pour garder la simulation et le rendu interactifs) représentent de larges parcelles d'air auxquelles sont attachées des informations de température potentielle, de masse d'eau et de ratio de mélange entre eau condensée et vapeur (figure 1.19 page ci-contre). Les conditions environnementales de température et d'humidité sont modélisées indépendamment par des champs implicites afin d'éviter un stockage lourd et détaillé dans les zones vides. Sont alors simulés les phénomènes de convection prenant en compte la libération d'énergie latente par condensation, ainsi que la diffusion qui entraîne des échanges de masse entre les parcelles et entre les parcelles et l'environnement. Un certain contrôle est fourni à l'utilisateur qui peut notamment modifier le profil de température pour limiter l'altitude des nuages, l'altitude de condensation à partir de laquelle peuvent apparaître des nuages, et le rayon des parcelles nuageuses.





**FIGURE 1.19.** – Formation de nuages avec 75 parcelles par Goswami *et al.* [51].

ELHADDAD *et al.* [36] définissent un système de particules simplifié pour simuler la formation de nuages en temps réel en évitant de résoudre les équations de NAVIER-STOKES (figure 1.20). Le mouvement de chaque particule est simplement exprimé comme la somme de forces d'attraction-répulsion qu'exercent ses voisines en fonction de leur distance. Cette fonction est appelée fonction de potentiel de LENNARD-JONES et sert classiquement à modéliser les interactions entre atomes ou molécules neutres, mais les auteurs l'adaptent au cas de parcelles d'air qui supportent les nuages. La simulation est initialisée en plaçant jusqu'à 50 000 particules à la surface. La température de chaque particule est exprimée comme inversement proportionnelle à son altitude et une particule devient nuageuse lorsqu'elle dépasse l'altitude de condensation, tandis que sa température est alors considérée comme constante.



**FIGURE 1.20.** – Nuages obtenus par ELHADDAD *et al.* [36] par simulation d'un système de 48 000 particules.

### 1.3.2 Génération procédurale

En génération procédurale, l'accent est mis sur le processus de création. Il est primordial de pouvoir créer un résultat efficacement, que ce soit en matière de temps ou de maniabilité, de contrôle. À l'élaboration d'un outil procédural, s'opposent alors généralement l'automatisation, le contrôle et la variété de la génération. Ainsi, une simulation se positionne quasiment à l'extrême de l'automatisation totale, retournant un unique résultat, sans aucun contrôle si ce n'est la définition de l'état initial. D'un autre côté, les méthodes procédurales exploitent souvent des processus stochastiques afin de produire un assortiment de résultats. Mais là encore, un équilibre est à trouver car un outil au comportement trop aléatoire est difficile à contrôler, et les essais-erreurs nécessaires à l'utilisateur diminuent l'automatisation. Enfin, il est évident que la contrôlabilité s'oppose à l'automatisation et à la variété car le contrôle absolu est de définir manuellement et élément par élément ce qui est créé.

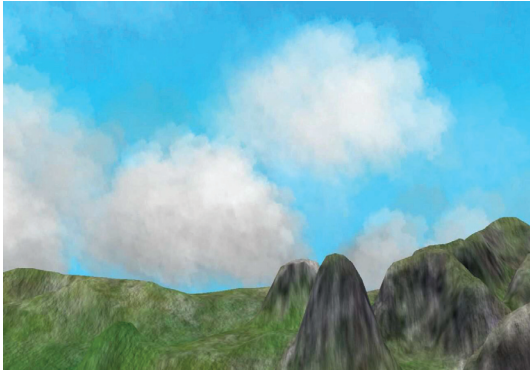
Contrairement aux simulations, la chaîne causale importe peu à la génération procédurale tant que le résultat généré correspond à ce que l'utilisateur souhaite. Les méthodes procédurales permettent de générer ce qui ne s'explique pas et ne serait donc pas simulable, ou ce qui serait trop coûteux à simuler. La génération procédurale permet par exemple de reproduire des phénomènes naturels tels que des nuages sans même savoir pourquoi et comment ils se forment physiquement, mais en sachant à quoi ils ressemblent, quels sont leurs motifs et leurs comportements habituels. Les méthodes de génération procédurale sont ainsi particulièrement communes en informatique graphique notamment dans les cas d'applications interactives et d'outils artistiques, pour lesquels la rapidité et le contrôle priment.

Des méthodes de simulation de nuages sont utilisées en météorologie ainsi qu'en informatique graphique (section 1.3.1 page 44). Cependant, les processus physiques qui régissent la formation et le comportement des nuages sont complexes, ce qui rend leur simulation coûteuse et limite la résolution qu'il est possible de calculer. Les détails les plus fins que les nuages exhibent ne peuvent donc pas être simulés, et s'il est courant de combiner une animation de nuage obtenue par simulation avec des détails procéduraux, il existe même tout un panel de méthodes entre la pure simulation physique et la génération procédurale directe.

**Modèle volumique animé** Les méthodes existantes de génération procédurale de nuages s'intéressent tout d'abord à modéliser l'aspect nuageux, leur texture. Suivant la distance des nuages à ajouter dans une scène, des effets convaincants peuvent être obtenus en plaquant une texture nuageuse sur une surface. Cela se justifie par exemple si le point de vue est très loin dans l'espace, la couche nuageuse pouvant être représentée par un ou plusieurs calques sphériques transparents de texture, ou si le point de vue est situé à terre et que le ciel est déjà représenté par des textures plaquées sur un dôme hémisphérique, ou une sphère ou un cube englobants (*skybox*). Il est aussi possible d'utiliser la technique des imposteurs (*billboards*) afin de créer des panneaux nuageux et un effet de parallaxe. Mais dans le cas général, les points de vue proches voire à l'intérieur des nuages nécessitent de considérer l'aspect volumique des nuages. La texture nuageuse peut ainsi être définie en tout point de l'espace ou être restreinte à un support compacte par une union de primitives géométriques telles que des sphères ou des ellipsoïdes. De manière similaire, certaines méthodes ne s'intéressent pas à la génération de nuages animés, ce qui est acceptable sur un laps de temps réduit. Dans le cas contraire, l'animation des nuages dans l'atmosphère permet d'enrichir une scène en ajoutant au réalisme d'une part, mais aussi en faisant évoluer son ambiance. Suivant le type de représentation utilisée, le support des nuages peut être animé, ou bien leur texture, voire les deux à la fois.

## Esquisses

Des modèles complexes peuvent être créés en éditant minutieusement tous les paramètres décrivant la représentation d'un nuage [130], mais cette méthode est lente et fastidieuse pour l'utilisateur. Les techniques à base d'esquisses visent à fournir à l'utilisateur un haut niveau de contrôle sur la modélisation d'un nuage. Les systèmes d'esquisse de nuage permettent à l'artiste de dessiner grossièrement la forme globale d'un nuage à partir de laquelle le système génère automatiquement un nuage tridimensionnel correspondant.



tions nuageuses passe donc directement par la définition des profils de distribution, produisant plutôt une animation procédurale qu'une simulation.



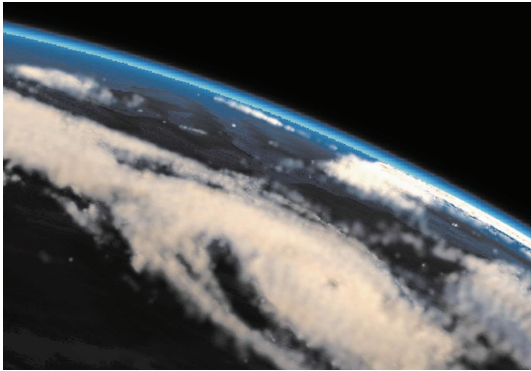
simulé et définit le sommet de la couche nuageuse de *Cumulus* vers laquelle est guidée la simulation. Afin de contraindre la simulation à prendre la forme dessinée, deux méthodes influencent indépendamment les déplacements verticaux et horizontaux des nuages. Verticalement, la chaleur latente et le ratio nuage/vapeur sont exprimés proportionnellement à l'écart d'altitude entre le sommet des nuages et le sommet de l'esquisse, ce qui favorise la convection et élève les parties nuageuses sous l'esquisse. Horizontalement, des forces externes sont ajoutées à la simulation d'après le champ de distance à la forme à atteindre.



WITHER *et al.* [145] proposent une autre méthode pour créer un nuage cumuliforme d'après des esquisses utilisateur qui ne contrôlent pas une simulation physique comme dans l'article de DOBASHI *et al.* [30], mais servent directement de contours à un nuage. De plus, l'utilisateur peut dessiner plusieurs esquisses selon différents points de vue. Une fois l'ensemble des esquisses dessinées par l'utilisateur, un squelette du volume nuageux est extrait et sert de support à la création d'un ensemble représentatif de primitives implicites ponctuelles. L'isosurface des primitives implicites est finalement discrétisée sous la forme d'un maillage classique qui est quant à lui rendu avec des détails procéduraux d'après la méthode de BOUTHORS *et al.* [15].

### Méthodes basées exemple





DOBASHI *et al.* [27, 28] proposent de construire des modèles nuageux composés de primitives implicites en utilisant les données recueillies par télédétection telles que des photos satellite de la couverture nuageuse terrestre. Afin de résoudre ce problème complexe de modélisation inverse, les auteurs utilisent un algorithme glouton qui isole répétitivement le pixel d'intensité maximale puis instancie une primitive implicite à l'intersection entre le rayon du plan image associé au pixel et un plan défini par l'utilisate

ur. Les paramètres de rayon et de densité de la primitive implicite instanciée sont déterminés par une boucle d'optimisation qui calcule un rendu et estime l'écart d'intensité des pixels couverts entre le rendu et la photo. Les auteurs proposent même d'animer les modèles nuageux obtenus d'après des photos satellite subséquentes. Pour ce faire, l'utilisateur définit le champ vectoriel des vents en dessinant un ensemble de courbes de BÉZIER d'après les flux observés sur les photos satellite. Les primitives sont ensuite animées en suivant les vents et en réalisant un fondu enchaîné des modèles subséquents par animation des paramètres de densité : les paramètres de densité des anciennes primitives diminuent et elles disparaissent progressivement en faisant place aux nouvelles dont les densités s'accroissent.

DOBASHI *et al.* [23] proposent cette fois d'obtenir des modèles nuageux en fournissant non plus des photos satellite mais plus simplement des photos de nuages prises depuis le sol (figure 1.21). Chaque photo de nuages est segmentée d'après les zones nuageuses au premier plan, les conditions d'éclairage sont estimées d'après le fond, et deux images représentant l'opacité nuageuse et l'intensité lumineuse due aux nuages sont calculées. Dans le cas des *Cirrus* qui sont très fins, une texture est extraite puis appliquée à un plan placé d'après l'angle de la prise de vue. Pour les couches nuageuses un peu plus épaisses telles que les *Alto cumulus* ou les *Cirrocumulus*, des primitives implicites sont créées et placées dans un plan. Quant aux nuages à développement plus vertical comme les *Cumulus* ou les *Cumulonimbus*, leur surface est estimée en considérant que les parties centrales de ces nuages sont les plus denses, ce qui permet de construire un squelette dans l'espace de la photo qui définit une forme tridimensionnelle ensuite perturbée par du bruit de PERLIN.



**FIGURE 1.21.** – Photographie de *Cirrocumulus* (gauche) et *Cirrocumulus* synthétisés (droite) par DOBASHI *et al.* [23].

Quelques années plus tard, les travaux subséquents de YUANG *et al.* [152, 153, 154, 156, 157] se concentrent aussi sur la modélisation de nuages d'après des photos prises depuis la surface ou par des satellites.



YUAN *et al.* [154] et ZHANG *et al.* [156] construisent la forme d'un *Cumulus* d'après une unique photo prise au sol. La surface du nuage est définie comme une carte de hauteur dans le plan image dont les valeurs sont déterminées d'après la direction d'ensoleillement et l'intensité lumineuse des pixels. Un algorithme glouton estime itérativement les hauteurs des zones les plus proches et les moins profondes du nuage du point de vue du soleil, ce qui correspond à un parcours dans l'ordre décroissant des intensités lumineuses. La forme globale du nuage est considérée comme symétrique et un effet miroir est appliqué au champ de hauteur pour définir l'arrière du nuage. Afin de limiter les aberrations géométriques due à l'effet miroir, les bords sont contraints à une hauteur nulle puis la forme est échantillonnée par des particules dont les tailles sont

proportionnelles à la distance de la surface pour obtenir un maximum de particules et de précision à la surface.



YUAN *et al.* [152] et ZHANG *et al.* [157] utilisent les jeux de données satellite de la mission de télédétection *Landsat8*. Ces jeux de données sont composés de relevés de la surface terrestre à onze longueurs d'ondes du spectre lumineux et à des résolutions entre 100 m et 15 m. Les auteurs utilisent les images du domaine infrarouge pour estimer les hauteurs de la cime nuageuse d'après les températures observées tandis que l'altitude de la base nuageuse est estimée aux

bord des nuages, où la base et la cime nuageuses se rejoignent. Les formes nuageuses sont ensuite enrichies procéduralement en ajoutant récursivement des sphères de tailles décroissantes au niveau de la surface de manière similaire à NISHITA *et al.* [107]



IWASAKI *et al.* [70] proposent de modéliser un nuage d'après une photographie et une base de données d'exemples nuageux obtenus par la méthode de simulation physique de DOBASHI *et al.* [30] sur des grilles de jusqu'à  $120 \times 60 \times 82$  cellules. De manière similaire aux travaux précédents, le volume nuageux global est construit à base de volumes élémentaires mais les primitives implicites sont remplacées par des colonnes de voxels tirées de la base d'exemples simulés. Chaque colonne de la base contient les

densités de ses voxels ainsi qu'une image correspondant à son rendu. La construction du volume global se fait alors en deux temps : l'initialisation du volume puis son optimisation.

Dans l'ordre de traversée par la lumière incidente, chaque colonne du volume est initialisée en recopiant les voxels de la colonne simulée dont les pixels ont le rendu le plus similaire à la photo. Une optimisation est ensuite menée en substituant successivement les colonnes simulées les plus proches en terme d'écart de densité de leurs voxels et d'écart entre les pixels rendus. Une variante est aussi proposée afin de combiner deux photos correspondant à deux points de vue orthogonaux plutôt qu'une seule et en adaptant les métriques de l'initialisation et de l'optimisation en conséquence.

Dans la publication récente de CHEN *et al.* [19], une base de données de photos de nuages isolés est construite et les modèles volumiques associés sont construits d'après la méthode de YUAN *et al.* [154]. L'utilisateur dessine ensuite une esquisse de forme nuageuse et le modèle nuageux le plus proche de la base est récupéré.

### Modélisations et animations procédurales *ex nihilo*



GARDNER [40, 41] définit des modèles de couches nuageuses composées de *Stratus* ou de *Cumulus* par placement et orientation stochastiques d'ellipsoïdes sur lesquels est plaquée une texture nuageuse. Les détails de la texture nuageuse sont animés en translatant le domaine de la série de FOURIER ou en animant ses paramètres et les ellipsoïdes sont étirés ou décalés verticalement pour produire un effet de croissance verticale. La texture nuageuse est exprimée par une série de FOURIER composée d'entre quatre et sept harmoniques sinusoidales aléatoirement déphasées. Cette texture nuageuse est plaquée sur un plan horizontal pour modéliser les nuages en haute altitude et les couvertures nuageuses planes. Pour les nuages les plus proches de la surface ou présentant un important développement vertical, la texture nuageuse est appliquée aux surfaces d'un ensemble de primitives ellipsoïdales et détermine leur transmittance tandis que l'ombrage est simplement calculé à la surface de l'ellipsoïde. Pour modéliser des structures nuageuses complexes, plusieurs ellipsoïdes qui se chevauchent sont combinés et leurs intersections sont calculées et ordonnées pour déterminer comment combiner leurs transmittances.

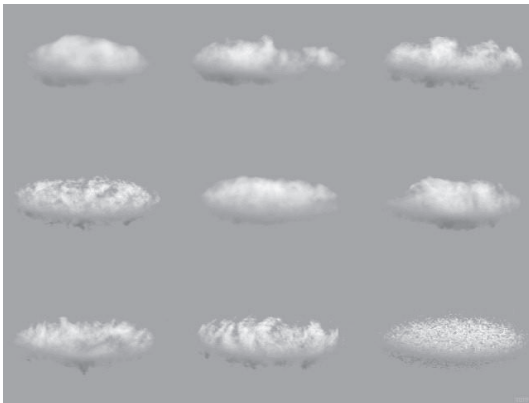
PEACHEY [116] et PERLIN [118] publient leurs travaux de synthèse de texture volumique qu'ils appellent texture solide. Si PEACHEY travaille avec les outils de synthèse spectrale par séries de FOURIER de manière similaire à GARDNER [41], PERLIN innove en définissant une fonction stochastique de bruit modélisant un signal spatial statistiquement invariant par rotation et translation et dont la fréquence est bornée ; le bruit de PERLIN est né. Il compose plusieurs octaves de ce bruit pour définir la fonction de turbulence qui lui permet de générer des textures de marbre et des vaguelettes à la surface d'un océan, mais aussi des nuages.





MAX [95] s'intéresse principalement au rendu de nuages mais en propose tout de même une modélisation sous la forme d'une carte de hauteur symétrique par rapport à un plan horizontal.

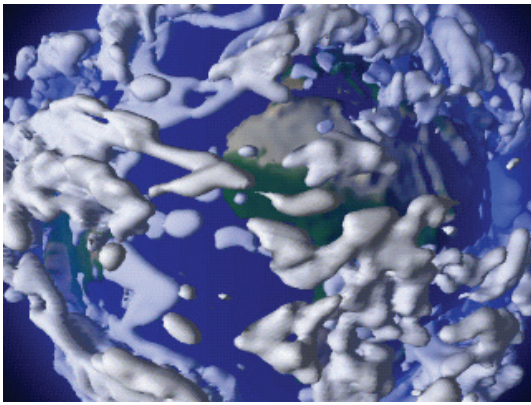
Le livre *The Science of Fractal Images* sort en 1988 et couvre l'utilisation de fractales pour modéliser procéduralement de nombreux éléments naturels tels des montagnes, des planètes, des plantes et des nuages par exemple. Les auteurs expliquent que le mouvement brownien est un cas particulier de fractale à une variable dont la dimension est 1.5 soit exactement entre la courbe de dimension un et la surface de dimension deux. Ils proposent plusieurs méthodes pour en construire la généralisation à toute dimension intermédiaire, *id est* le mouvement brownien fractionnaire, et s'en servent pour modéliser différents objets tels que des montagnes, des planètes ou des nuages. Si les auteurs qualifient de nuage toute fractale à trois variables, seuls quelques exemples de nuages représentant un champ d'eau condensée sont donnés, et si des astuces sont données pour modéliser des montagnes plus ou moins jeunes, aucune discussion n'est faite concernant la reproduction de types particuliers de nuages. Un point notable est apporté lorsque les auteurs citent le travail de Shaun LOVEJOY. LOVEJOY [92] a montré que le développement horizontal des nuages présente une dimension fractale de l'ordre de  $\frac{1}{3}$  quelle que soit l'échelle des nuages entre 1 km et 1000 km. De nombreux autres travaux ont été menés en météorologie afin de caractériser les différents types de nuages et leurs formes intermédiaires en mesurant leur dimension fractale, mais il n'y a pas encore consensus en la matière [7, 55].



PERLIN *et al.* [119] proposent d'enrichir la surface d'un objet en le couvrant d'une certaine épaisseur de texture solide que les auteurs appellent l'hypertexture. Aucune application à la modélisation de nuage n'est présentée mais une autre publication simultanée de LEWIS [86] présente différents résultats obtenus à partir du bruit de PERLIN, dont des nuages. Cependant, les paramètres utilisés pour obtenir les différents motifs nuageux ne sont pas détaillés.

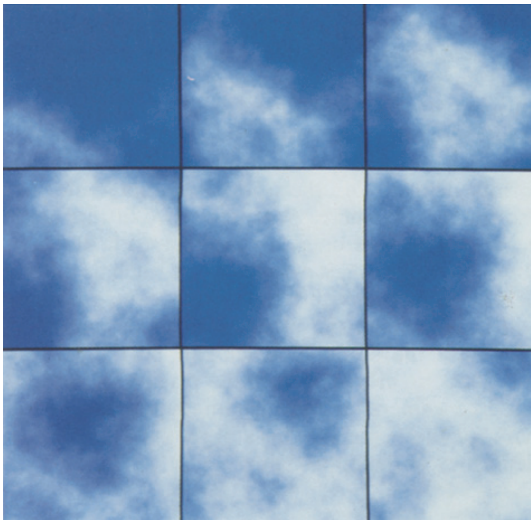


EBERT *et al.* [32] appliquent le bruit et la turbulence de PERLIN dans des formes simples pour y définir des champs de densité. Le champ de densité est animé en perturbant les coordonnées de texture par la fonction de turbulence et en leur appliquant un effet de rotation héli-coïdale. Des effets nuageux sont ainsi modélisés et animés, tels que du brouillard qui roule sur un échiquier et de la vapeur partiellement condensée qui s'élève au-dessus d'une tasse.



MAX *et al.* [97] publient une méthode développée afin de créer une animation continue et détaillée de nuages à l'échelle planétaire afin de visualiser les résultats de simulations météorologiques, et plus particulièrement le champ vectoriel des vents. Le champ de densité nuageuse grossier (grille atmosphérique de  $320 \times 160 \times 19$  cellules) qui est fourni par la simulation est enrichi par des détails procéduraux composés de séries de FOURIER d'après la méthode de GARDNER [41], mais l'originalité de leur approche consiste

à animer la couche nuageuse en advectant la texture nuageuse selon le champ de vents. Au rendu de chaque image de la séquence, chaque intersection avec la surface nuageuse est mise en correspondance avec une position d'origine dans l'espace de la texture qui est extrapolée en remontant le temps et les vents. Afin de limiter la complexité et les distorsions engendrées par l'intégration des trajectoires chaotiques du champ de vent sur un long intervalle temporel, la séquence complète est découpée en intervalles de temps plus courts. Sur chaque intervalle, les champs de densité aux deux instants clefs de la simulation sont mélangés avant d'y appliquer la texture procédurale détaillée. Puis, chaque point de la surface nuageuse est mis en correspondance avec deux valeurs de texture en suivant les deux champs de vent clefs, et la valeur de texture nuageuse finale est obtenue par interpolation d'après la progression dans l'intervalle de temps.

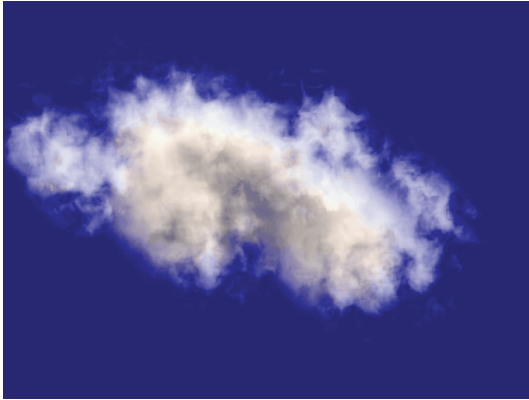


SAKAS [127] et SAKAS *et al.* [128] explicitent les liens entre la synthèse de texture solide à base de bruit de PERLIN et la synthèse spectrale à partir de séries de FOURIER. Les auteurs s'inspirent des travaux d'analyse statistique des phénomènes physiques à turbulence afin de proposer une méthode de synthèse de texture nuageuse animée plus cohérente et naturelle. En effet, un constat important de l'étude de la turbulence est que l'expression de l'intensité du mouvement chaotique d'une parcelle en fonction de sa taille n'est pas monotone. Au contraire, l'intensité du mouvement chaotique est quasiment nulle à petite et grande échelle tandis qu'entre les deux, l'intensité connaît un maximum local.

SAKAS *et al.* [128] appliquent cette observation en adaptant la somme récursive du bruit de turbulence de PERLIN tandis que SAKAS [127] l'appliquent dans le cadre de la synthèse spectrale par série de FOURIER.

Parmi les incontournables de la génération procédurale figure le livre *Texturing and Modeling* [33] dont la première édition, sortie en 1994, traite notamment de la modélisation procédurale de phénomènes gazeux en utilisant des textures solides procédurales comme fonction de densité. L'application aux nuages est illustrée par les exemples des *puffy clouds*, des *Cirrus* ainsi qu'une modélisation à plusieurs niveaux de la couche nuageuse planétaire avec des vortex.

NISHITA *et al.* [107] publient un article principalement focalisé sur le rendu de scènes nuageuses composées de primitives implicites et décrivent un processus récursif d'ajout de détails qu'ils qualifient de *fractal*. L'idée est de raffiner successivement la surface nuageuse définie par l'isosurface des primitives implicites en la discrétisant par la méthode de LORENSEN *et al.* [91], puis en y plaçant de nouvelles primitives implicites de plus en plus petites.

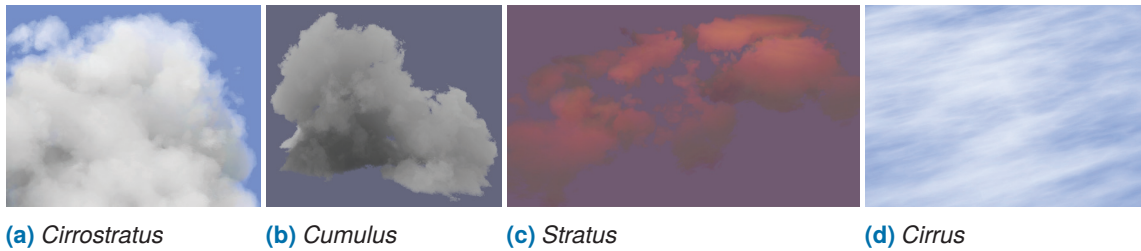


EBERT [31] propose une méthode de modélisation et d'animation de nuages volumiques fondée sur l'utilisation combinée de primitives implicites et de bruit procédural de type turbulence. Cette même représentation est ensuite utilisée dans la troisième édition de *Texturing and Modeling* [34] qui intègre plus d'éléments concernant la modélisation de nuages. Un chapitre entier est consacré à la modélisation et à l'animation procédurales de nuages volumiques sous la forme de mélanges de primitives impli-

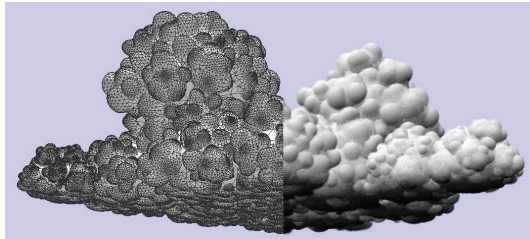
cites qui définissent la forme globale nuage puis sont combinées à du bruit procédural pour ajouter les détails les plus fins de la texture nuageuse. Les *Cumulus* y sont plutôt modélisés par des primitives implicites ponctuelles de forme sphériques qui correspondent aux bourgeons convectifs tandis que les *Cirrus* sont composés de primitives implicites beaucoup plus étirées. Dans le cas des *Cumulus*, les primitives sont nombreuses et permettent déjà de bien sculpter le volume et la texture nuageuse procédurale ne modélise que les plus fins détails. Pour les *Cirrus*, les primitives sont bien plus grandes, peu nombreuses et ne guident qu'assez peu la forme globale, rendant la texture procédurale plus importante pour reproduire les motifs caractéristiques des couches de nuages cirriformes. Un autre chapitre, qui traite des fractales, comporte une section dédiée à la modélisation de nuages à partir de fractales.

Et cela ne s'arrête pas là car la même combinaison de primitives implicites et de bruit procédural est réutilisée par SCHPOK *et al.* [130]. Les auteurs rappellent que les primitives implicites servent à sculpter intuitivement la forme globale des nuages. Ils donnent aussi quelques idées afin d'animer les nuages en associant aux primitives des trajectoires prédéfinies manuellement ou d'après une direction principale de vents, voire même en les animant par la méthode des images clefs, *id est* en définissant à quelques instants clefs les états des différents paramètres des primitives tels que leurs rayons ou densités par exemple. Les auteurs expliquent aussi comment animer les détails de la texture nuageuse. Afin de créer la sensation que ce sont les vents qui soufflent les primitives animées, des translations sont appliquées aux domaines des bruits procéduraux, tandis que des rotations sont appliquées pour créer un effet d'animation propre, *id est* indépendant des vents. Enfin, bien que les auteurs évoquent la possibilité de définir différents *tiers*, *id est* des types nuageux auxquels sont associées des textures et des dynamiques spécifiques (figure 1.22 page suivante), la description précise de ces *tiers* n'est pas donnée. L'exemple le plus détaillé est celui d'un *Cumulus* en train de se développer en *Cumulonimbus*. Les primitives implicites de ce nuage sont animées de manière à reproduire l'effet de convection tout en limitant l'altitude maximale et en étirant une primitive pour recréer l'enclume caractéristique des *Cumulonimbus*.





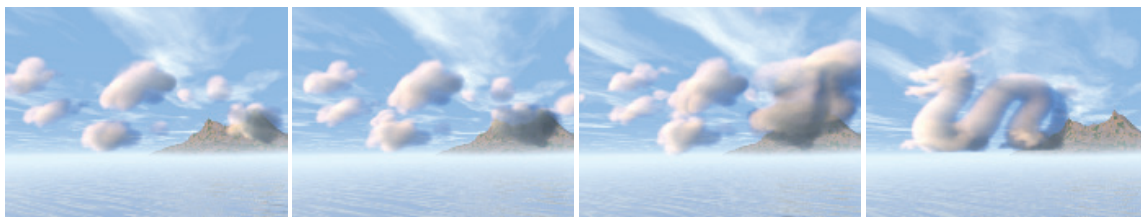
**FIGURE 1.22.** – Différents types de nuages modélisés par SCHPOK *et al.* [130].



BOUTHORS *et al.* [12] proposent une méthode procédurale afin de générer un nuage cumuliforme composé d'un ensemble de primitives implicites ponctuelles sphériques. L'ensemble des primitives est tout d'abord initialisé manuellement ou automatiquement par quelques primitives de grande taille. L'idée consiste ensuite à

ajouter récursivement du détail au niveau de l'isosurface de l'ensemble des primitives par adjonction stochastique de nouvelles primitives de plus en plus petites. Si ce processus de modélisation par raffinement récursif est très similaire à l'ajout de détails *fractals* de NISHITA *et al.* [107], il en diffère cependant par son application à l'ensemble de la modélisation, plutôt que seulement à l'ajout de détails. De plus, les auteurs ne discrétisent pas explicitement l'isosurface du mélange de primitives qui est plutôt échantillonnée par un système de particules attirées par l'isosurface, qui se repoussent entre elles et qui se séparent ou fusionnent de manière adaptative suivant la couverture de la surface. Bien qu'un tel système de particules semble être relativement instable et puisse par conséquent être stochastiquement perturbé pour obtenir une animation, aucune animation des nuages cumuliformes générés par cette méthode n'est cependant discutée.

Une autre manière de construire un modèle nuageux composé de primitives implicites est donnée par LIU *et al.* [89]. Une forme spécifiée sous forme d'un maillage polygonal est discrétisée par un algorithme glouton en un ensemble de primitives implicites de forme sphérique ou ellipsoïdale. De plus, deux méthodes simples d'animation des primitives nuageuses appelées l'agrégation et la diffusion sont décrites par les auteurs. L'agrégation consiste à spécifier les positions et formes des primitives nuageuses à différents instants clefs puis à interpoler linéairement les paramètres entre les images clefs (figure 1.23). Dans le cas de la diffusion, l'ensemble des primitives implicites est décomposé en plusieurs sous-ensembles selon les volumes des primitives, et une animation est obtenue en faisant graduellement apparaître les sous-ensembles de primitives. Les volumes sont de tailles décroissantes, raffinant progressivement la forme nuageuse.



**FIGURE 1.23.** – Différentes étapes d'une métamorphose par agrégation de nuages épars vers un dragon nuageux par LIU *et al.* [89].

Bien que la publication récente de PARGA *et al.* [115] soit principalement focalisée sur le rendu des nuages, les auteurs détournent de manière intéressante un outil classique de la génération procédurale de plantes – les L-systèmes – et proposent une grammaire qui génère des nuages comprenant des tours convectives.

### 1.3.3 Synthèse

Le tableau 1.2 page 63 synthétise les différentes méthodes de génération de nuages évoquées dans cet état de l'art. Si la plupart des méthodes s'intéressent à la modélisation de nuages cumuliformes (24/32), il y en a peu (17/32) qui parlent explicitement de *Cumulus* et encore moins (8/32) qui considèrent d'autres types de nuages. Certaines méthodes n'évoquent même aucun motif ni type nuageux. Ainsi, les *Stratus* ne sont couverts que par six méthodes [41, 54, 60, 114, 130], et les *Cirrus* par seulement trois [23, 40, 130]. Peut-être est-ce parce que les *Cumulus* ont plus d'importance dans l'imaginaire commun et qu'ils représentent un peu le modèle commun du nuage. Une autre explication serait que les fins filaments des *Cirrus* sont complexes à modéliser et à rendre et que les *Stratus* ressemblent plus à un brouillard ambiant informe qu'à un nuage. Concernant les *Cumulonimbus*, il est étonnant que seules trois méthodes [54, 98, 99] s'attachent à les modéliser car ce sont d'imposants nuages synonymes de tempêtes orageuses impressionnantes qui se prêtent bien aux effets spéciaux. Peut-être est-ce justement à cause de leur taille et des conditions météorologiques instables qui les accompagnent qui font de leur modélisation un problème plus complexe.

Les dimensions des nuages modélisés sont rarement données mais la plupart des méthodes se concentrent sur la modélisation d'un petit nombre de nuages isolés au-dessus d'un terrain de quelques centaines de kilomètres carrés. Seules trois méthodes [26, 33, 97] s'intéressent à la modélisation de la couverture nuageuse planétaire. À l'échelle planétaire, les types de nuage ne sont plus considérés, certainement parce que les détails nécessaires pour les représenter seraient trop fins.

Sans surprise, toutes les simulations génèrent des nuages animés mais seuls DOBASHI *et al.* [26] et GOSWAMI *et al.* [51] fournissent un certain contrôle utilisateur autre que la description des conditions initiales. Du côté des méthodes procédurales, les nuages modélisés ne sont pas tous animés mais le contrôle utilisateur est bien plus présent, notamment grâce aux méthodes à base d'esquisses. Parmi les méthodes à base d'exemples, DOBASHI *et al.* [27] sont les seuls à proposer une animation en laissant l'utilisateur éditer les vents principaux, plutôt que de se limiter à éditer les exemplaires nuageux photographiés. Dans le reste des méthodes procédurales, environ la moitié fournissent à la fois animation et contrôle utilisateur. L'animation continue obtenue par MAX *et al.* [97] à partir des résultats d'une simulation météorologique est entièrement automatisée et ne fournit aucun contrôle.

## 1.4 Conclusion

Il existe de nombreux phénomènes atmosphériques auxquels la communauté de l'informatique graphique s'est intéressé afin de les reproduire dans des scènes synthétiques. La qualité, l'efficacité et la simplicité des méthodes proposées peuvent cependant en-

core être améliorées, et cela sans même considérer les phénomènes encore inconnus ou inexplicables. Plus particulièrement, les nuages sont un phénomène atmosphérique majeur dont la modélisation, l'animation et le rendu restent des problèmes encore non résolus.

Tout d'abord, le rendu de nuages requiert une simulation du transfert radiatif de la lumière dans l'atmosphère qui se révèle très coûteuse (annexe A page 135). Un certain nombre de méthodes existantes cherchent à mitiger ce coût et à élaborer une méthode de rendu interactif, voire temps réel, grâce à des approximations choisies avec soin selon les conditions spécifiques de la scène synthétique. Dans cet esprit, nous avons tout d'abord implémenté une méthode de rendu par *ray marching* dans l'approximation d'une diffusion unique (*single scattering*) nous permettant de visualiser une scène de paysage nuageux en quelques secondes de calcul. Mais certaines conditions d'illumination telles que des couches nuageuses éclairées par le soleil couchant se révèlent trop particulières pour que les effets grandioses puissent être reproduits avec fidélité. Inspirés par les méthodes de lancer de rayon à base d'échantillonnage de MONTE CARLO, nous avons ensuite implémenté une méthode de rendu générique et qualitative, dont les performances ne se révèlent pas diamétralement opposées à notre première approximation, notamment car le rendu devient progressif.

Concernant la modélisation de nuages, la plupart des méthodes s'intéressent à modéliser quelques nuages isolés et peinent à reproduire les formes et animations caractéristiques des différents types de nuage. Les simulations ont l'avantage de produire des animations réalistes, mais leur coût prohibitif empêche le passage à l'échelle et limite la résolution spatiale et temporelle nécessaires aux lois de la thermodynamique pour reproduire les formations nuageuses les plus fines. À l'opposé, les méthodes procédurales se focalisent sur le contrôle de la forme des nuages, permettant la production artistique de nuages convaincants mais néanmoins principalement statiques, leur animation étant une tâche répétitive et requérant encore plus d'investissement. Afin d'obtenir des nuages animés, détaillés et caractéristiques, nous proposons une sélection de bruits procéduraux avec une animation propre et adaptés à chaque type de nuages. Pour faciliter la modélisation de grandes étendues nuageuses animées, nous proposons un contrôle à haut niveau aussi bien spatial que temporel qui repose sur un principe intuitif et bien connu des artistes : l'interpolation d'images clefs ; des cartes de la couverture nuageuse que l'utilisateur dessine aux instants qu'il choisit. Enfin, pour garantir la cohérence de l'animation, la méthode d'interpolation des images clefs que nous proposons permet de prendre compte l'environnement physique des nuages constitué d'un terrain et de vents. Le tout pouvant être réglé afin de suivre le plus fidèlement l'environnement physique ou adhérer au plus près aux images clefs fournies.

**TABLE 1.2.** – Synthèse des différentes méthodes de modélisation de nuages : quels types de nuages sont modélisés, si les nuages sont animés, si la modélisation est contrôlable, et l'échelle et la résolution en voxels, en particules, ou en primitives des nuages modélisés.

	Auteurs	Réf.	Année	Types de nuage	Anim.	Ctrl.	Dimensions	Résolution
Simulations	KAJIYA <i>et al.</i>	[74]	1984	<i>Cumulus</i>	X		?	10 × 10 × 20 c.
	MiyAZAKI <i>et al.</i>	[98]	2002	<i>Cumulus Cumulonimbus</i>	X		3 × 2.4 × 2 km <sup>3</sup>	150 × 120 × 100 c.
	OverBY <i>et al.</i>	[114]	2002	<i>Cumulus Stratus</i>	X		?	50 × 15 × 15 c.
	HARRIS <i>et al.</i>	[60]	2003	<i>Cumulus Stratus</i>	X		?	64 × 32 × 64 c.
	DOBASHI <i>et al.</i>	[26]	2006		X		planétaire	160 × 80 × 4 c.
	DOBASHI <i>et al.</i>	[29]	2000	<i>Cumulus</i>	X		?	256 × 256 × 20 c.
	MiyAZAKI <i>et al.</i>	[99]	2001	<i>Cumulus AltoCumulus Cumulonimbus Cirrocumulus</i>	X		5 × 5 × 0.4 km <sup>3</sup>	256 × 256 × 40 c.
	NEYRET	[105]	1997	<i>Cumulus</i>	X		?	
	BARBOSA <i>et al.</i>	[5]	2015	cumuliformes ?	X		?	930 000 particules
	GOSWAMI <i>et al.</i>	[51]	2016	<i>Cumulus</i>	X	~	?	85 particules
ELHADDAD <i>et al.</i>	[36]	2016	cumuliformes ?	X		?	50 000 particules	
Méthodes procédurales	GRUDZIŃSKI <i>et al.</i>	[54]	2007	<i>Cumulus Stratus Nimbostratus Cumulonimbus</i>	X	~	15 × 15 km <sup>2</sup>	
	DOBASHI <i>et al.</i>	[30]	2008	cumuliformes	X	X	?	320 × 80 × 100 c.
	WITHER <i>et al.</i>	[145]	2008	cumuliformes	X	X	?	1866 sphères
	DOBASHI <i>et al.</i>	[27]	1998		X	X	1000 km	
	DOBASHI <i>et al.</i>	[23]	2010	<i>Cumulus AltoCumulus Cirrus</i>	X	~	?	
	YUAN <i>et al.</i>	[152]	2015	<i>Cumulus</i>	X	~	20 × 20 km <sup>2</sup>	150 000 c.
	ZHANG <i>et al.</i>	[157]	2017	<i>Cumulus</i>	X	~	20 × 20 km <sup>2</sup>	
	IWASAKI <i>et al.</i>	[70]	2017	cumuliformes ?	X	~	?	120 × 60 × 82 c.
	GARDNER	[41]	1985	<i>Cumulus Stratus Cirrus</i>	X	X	?	
	MAX	[95]	1986	cumuliformes	X	X	?	
	BARNsLEY <i>et al.</i>	[6]	1988		X	X	?	
	EBERT <i>et al.</i>	[32]	1990		X	X	?	
	MAX <i>et al.</i>	[97]	1992		X	X	?	
	SAKAS <i>et al.</i>	[128]	1992		X	X	planétaire	320 × 160 × 19 c.
	EBERT <i>et al.</i>	[33]	1994		X	X	?	128 × 128 × 128 c.
	NISHITA <i>et al.</i>	[107]	1996		X	X	planétaire	
	EBERT	[31]	1997		X	X	?	
	SCHPOK <i>et al.</i>	[130]	2003	<i>Cumulus</i>	X	X	?	
BOUTHORS <i>et al.</i>	[12]	2004	<i>Cumulus Stratocumulus Stratus Cirrus</i>	X	X	?		
Liu <i>et al.</i>	[89]	2006	<i>Cumulus</i>	X	X	?		
PARGA <i>et al.</i>	[115]	2018	cumuliformes	X	X	?		
				<i>Cumulus</i>				20 × 20 × 20 c.





## Prérequis

Ce chapitre rappelle les fondements de la modélisation implicite, et des bruits procéduraux qui sont nécessaires à la modélisation des nuages, tandis que des éléments concernant le rendu sont donnés annexe A page 135.

### 2.1 Modélisation implicite

Dans cette section, nous rappelons les principes de la modélisation implicite à base de primitives à squelette, telles que le Blob de BLINN [9], combinées par une structure hiérarchique telle que le BlobTree [149], que nous utilisons par la suite pour modéliser (section 3.2 page 92) et animer (chapitre 4 page 103) la présence nuageuse à grande échelle.

**Définition 1 (Champ scalaire)** *Un champ scalaire est une fonction qui à tout point de l'espace associe une valeur scalaire. Un champ scalaire de température est ainsi une fonction qui donne la température en tout point de l'espace.*

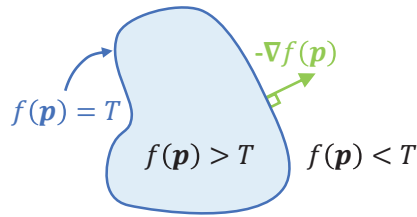
**Définition 2 (Isosurface, isovaleur, champ de potentiel et surface implicite)** *Une isosurface est la généralisation tridimensionnelle de la courbe de niveau qui est bidimensionnelle. C'est l'ensemble des points de l'espace en lesquels un champ scalaire prend une même valeur appelée isovaleur. Une surface implicite est l'ensemble des zéros d'un champ scalaire appelé champ de potentiel ou champ de distribution de densité (figure 2.1 page suivante). Soit  $f$  un champ de potentiel et  $T$  une isovaleur ; la surface implicite associée est :*

$$S_T = \{ \mathbf{p} \in \mathbb{R}^3 \mid f(\mathbf{p}) - T = 0 \}. \quad (2.1)$$

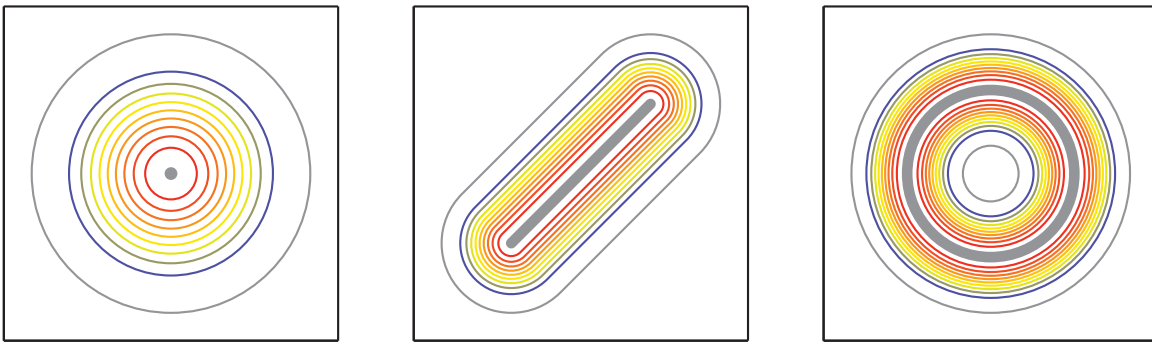
*Par convention, nous orientons les surfaces implicites selon la direction opposée du gradient de potentiel, id est l'intérieur est composé des points de potentiel positif et l'intérieur des points de potentiel négatif.*

#### 2.1.1 Primitive implicite à squelette

De nombreuses formes primitives peuvent être décrites implicitement par un champ de potentiel, comme une sphère, une ellipsoïde ou bien un tore par exemple. Une primitive implicite à squelette possède un champ de potentiel qui décroît selon la distance à un lieu appelé squelette. Le squelette le plus simple se limite à un point mais il est généralement



**FIGURE 2.1.** – Schématisation d’une surface implicite, avec son intérieur, son extérieur et l’orientation selon la direction opposée du gradient du champ de potentiel.



**FIGURE 2.2.** – Visualisation de primitives implicites à squelette pour différents types de squelettes. De gauche à droite : point, segment (équation 2.3), cercle.

plus complexe et est composé d’un ensemble de points épars, de courbes, de surfaces et de volumes (figure 2.2). Soit  $S$  un squelette,  $d$  une fonction de distance et  $g$  une fonction d’atténuation ; le champ de potentiel d’une primitive à squelette s’exprime comme la composition de la fonction d’atténuation avec la fonction de distance :

$$f(\mathbf{p}) = g \circ d_S(\mathbf{p}) \quad \text{où} \quad d_S = \min_{\mathbf{c} \in S} d(\mathbf{c}, \mathbf{p}). \quad (2.2)$$

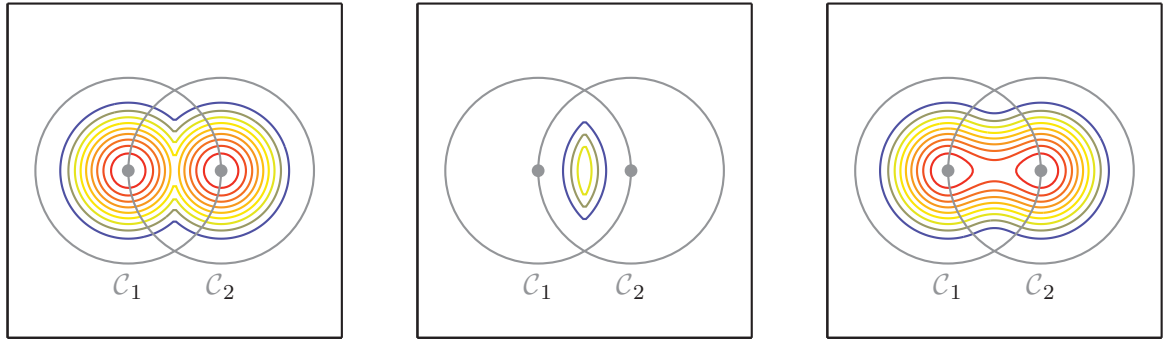
L’expression de la distance à un squelette réduit à un point est triviale : c’est simplement la distance au point. Pour un squelette  $S = [\mathbf{p}_1, \mathbf{p}_2]$  constitué d’un segment, la distance euclidienne au squelette s’exprime en fonction du ratio  $t$  de la projection orthogonale sur le segment :

$$d_S(\mathbf{p}) = \begin{cases} d(\mathbf{p}, \mathbf{p}_2) & \text{si } t \leq 0 \\ d(\mathbf{p}, \mathbf{p}_1) & \text{si } t \geq 1 \\ d(\mathbf{p}, \mathbf{p}_2 + t \cdot (\mathbf{p}_1 - \mathbf{p}_2)) & \text{sinon} \end{cases} \quad \text{où} \quad t = \frac{(\mathbf{p}_1 - \mathbf{p}) \cdot (\mathbf{p}_1 - \mathbf{p}_2)}{\|\mathbf{p}_1 - \mathbf{p}_2\|^2}. \quad (2.3)$$

Pour un squelette  $S = \mathcal{B}$  constitué d’une boule de centre  $\mathbf{c}$  et de rayon  $r$ , la distance euclidienne au squelette s’exprime par  $|r - d(\mathbf{p}, \mathbf{c})|$ .

### 2.1.2 Opérateurs de combinaison de champs de potentiel

De manière similaire à la géométrie de construction de solides (*Constructive Solid Geometry*), un modèle implicite complexe peut être construit comme une combinaison de primitives implicites à l’aide d’opérateurs booléens d’union, d’intersection et de différence [149].



**FIGURE 2.3.** – Visualisation de la combinaison de deux primitives à squelette ponctuel par différents opérateurs. De gauche à droite : opérateur d'union, opérateur d'intersection et opérateur de mélange (équations 2.4 à 2.6).

De multiples combinaisons successives forment un arbre de construction dont les feuilles sont des primitives implicites, dont chaque nœud interne correspond à une opération de combinaison de ses descendants en un modèle intermédiaire, et dont la racine est le modèle implicite complexe.

Il y a de nombreuses manières de définir les opérateurs de combinaison mais il faut prendre garde aux valeurs prises par le champ de potentiel résultant dans tout l'espace, pas seulement au niveau de la surface implicite. Dans ce qui suit, nous considérons  $n$  primitives  $\mathcal{C}_i$  et leurs champs de potentiel  $f_{\mathcal{C}_i}$ . La figure 2.3 illustre l'union et l'intersection de deux primitives sphériques.

**Définition 3 (Opérateur d'union)** Soit  $U = \bigcup \mathcal{C}_i$  l'union des primitives implicites, son champ de potentiel est exprimé comme le maximum des champs de potentiel des primitives

$$f_U(\mathbf{p}) = \max_i f_{\mathcal{C}_i}(\mathbf{p}). \quad (2.4)$$

**Définition 4 (Opérateur d'intersection)** De manière complémentaire à l'opérateur d'union, soit  $I = \bigcap \mathcal{C}_i$  l'intersection des primitives implicites, son champ de potentiel est exprimé comme le minimum des champs de potentiel des primitives

$$f_I(\mathbf{p}) = \min_i f_{\mathcal{C}_i}(\mathbf{p}). \quad (2.5)$$

À la définition d'un opérateur de combinaison, il faut aussi s'interroger sur le degré de continuité désiré. En effet, les opérateurs d'union par maximum et d'intersection par minimum introduisent des discontinuités au niveau des intersections entre les champs de potentiel, *id est* aux points où la primitive associée au champ extremum varie. Afin d'éviter de telles discontinuités, il existe un autre type d'opérateur de combinaison qui produit des jonctions lisses : l'opérateur de mélange aussi appelé opérateur de fusion.

**Définition 5 (Opérateur de mélange)** Soit  $M = \sum \mathcal{C}_i$  le mélange des primitives implicites, son champ de potentiel s'exprime comme la somme des champs de potentiel des primitives

$$f_M(\mathbf{p}) = \sum_i f_{\mathcal{C}_i}(\mathbf{p}). \quad (2.6)$$

Dans le cadre de la modélisation de la présence nuageuse (section 3.2 page 92), nous utilisons exclusivement l'opérateur de mélange.

### 2.1.3 Fonctions de distance

La fonction de distance notée  $d$  utilisée pour définir une primitive implicite à squelette n'est pas nécessairement euclidienne. La figure 2.4 page suivante illustre le comportement du champ de potentiel d'une primitive implicite à squelette ponctuel selon différentes fonctions de distance. Il est par exemple classique d'utiliser la distance euclidienne au carré par soucis de performance car cela évite de calculer des racines carrées. Il est aussi possible d'utiliser une fonction de distance qui n'accorde pas la même importance à tous les axes afin de dilater le squelette des primitives par un ellipsoïde plutôt que par une sphère par exemple [9, 150].

**Définition 6 (Distance euclidienne)** Soit  $\mathbf{v}$  le vecteur à  $n$  composantes séparant deux points  $\mathbf{p}_0$  et  $\mathbf{p}_1$ ; la distance euclidienne s'exprime par

$$\|\mathbf{v}\| = \sqrt{\sum_{i=0}^{n-1} \mathbf{v}[i]}. \quad (2.7)$$

**Définition 7 (Distance de Manhattan)** Soit  $\mathbf{v}$  le vecteur à  $n$  composantes séparant deux points  $\mathbf{p}_0$  et  $\mathbf{p}_1$ ; la distance de Manhattan s'exprime par

$$\sum_{i=0}^{n-1} |\mathbf{v}[i]|. \quad (2.8)$$

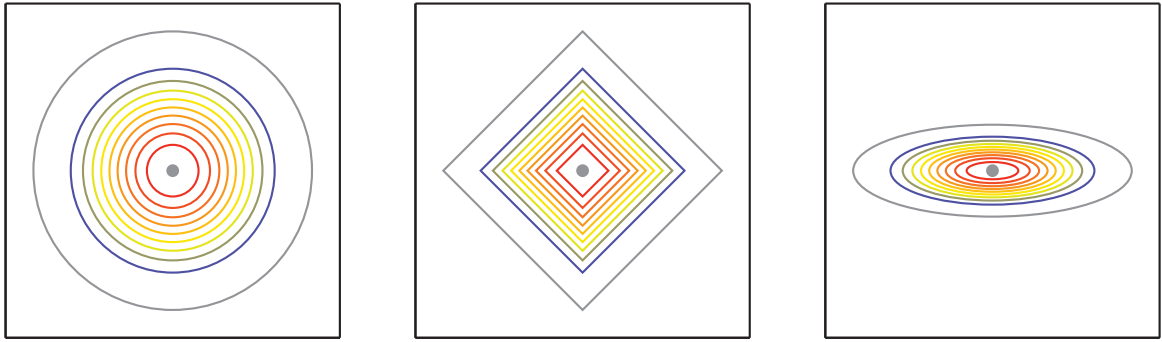
**Définition 8 (Distance ellipsoïdale)** Soit  $\mathbf{v}$  le vecteur à  $n$  composantes séparant deux points  $\mathbf{p}_0$  et  $\mathbf{p}_1$ , et  $\mathbf{e}_0$ ,  $\mathbf{e}_1$  et  $\mathbf{e}_2$  les vecteurs canoniques du repère cartésien et  $r_0$ ,  $r_1$  et  $r_2$  les rayons de l'ellipsoïde; la distance ellipsoïdale s'exprime par

$$\left| \mathbf{v} \cdot \left( \frac{\mathbf{e}_0}{r_0} + \frac{\mathbf{e}_1}{r_1} + \frac{\mathbf{e}_2}{r_2} \right) \right|. \quad (2.9)$$

Puisque les couches nuageuses que nous modélisons sont plutôt étendues horizontalement que verticalement, nous augmentons l'importance de la composante verticale de la fonction de distance. Nous obtenons ainsi des primitives ellipsoïdales aplaties comme des lentilles, ce qui nous permet de contrôler à grande échelle la présence nuageuse tout en limitant le nombre de primitives.

### 2.1.4 Fonctions d'atténuation

Une fonction d'atténuation permet de limiter l'influence du squelette d'une primitive en fonction de la distance à ce squelette. Il existe plusieurs fonctions d'atténuation qui présentent un intérêt dans le cadre du mélange de primitives implicites à squelette. Certaines permettent notamment d'assurer que la classe de continuité d'un mélange de primitives soit



**FIGURE 2.4.** – Visualisation d’une primitive implicite à squelette ponctuel selon différentes fonctions de distance. De gauche à droite : distance euclidienne, distance de Manhattan, distance euclidienne pondérée verticalement par un facteur trois (équations 2.7 à 2.9 page ci-contre).

d’un certain rang, tandis que d’autres limitent le rayon d’influence des primitives mélangées,  $id$  est la plus grande distance au squelette pour laquelle son champ de potentiel ne s’annule pas.

La figure 2.5 page suivante illustre le comportement du champ de potentiel cumuliforme d’un mélange de primitives implicites ponctuelles utilisant différentes fonctions d’atténuation. La fonction d’atténuation de BLINN assure la douceur du champ de potentiel du mélange mais son support est difficile à déterminer. En comparaison, le rayon d’influence de la fonction d’atténuation cubique permet d’exprimer le support du champ de potentiel du mélange comme l’union de boules de rayons donnés.

**Définition 9 (Fonction d’atténuation lisse de BLINN)** BLINN [9] propose une fonction d’atténuation inspirée de la modélisation du champ de densité des électrons gravitant autour d’atomes. Soit  $x$  la variable de distance au squelette et  $b$  un paramètre scalaire ; l’atténuation de BLINN est exponentielle en fonction de la distance au squelette :

$$g(x) = e^{-bx^2}. \quad (2.10)$$

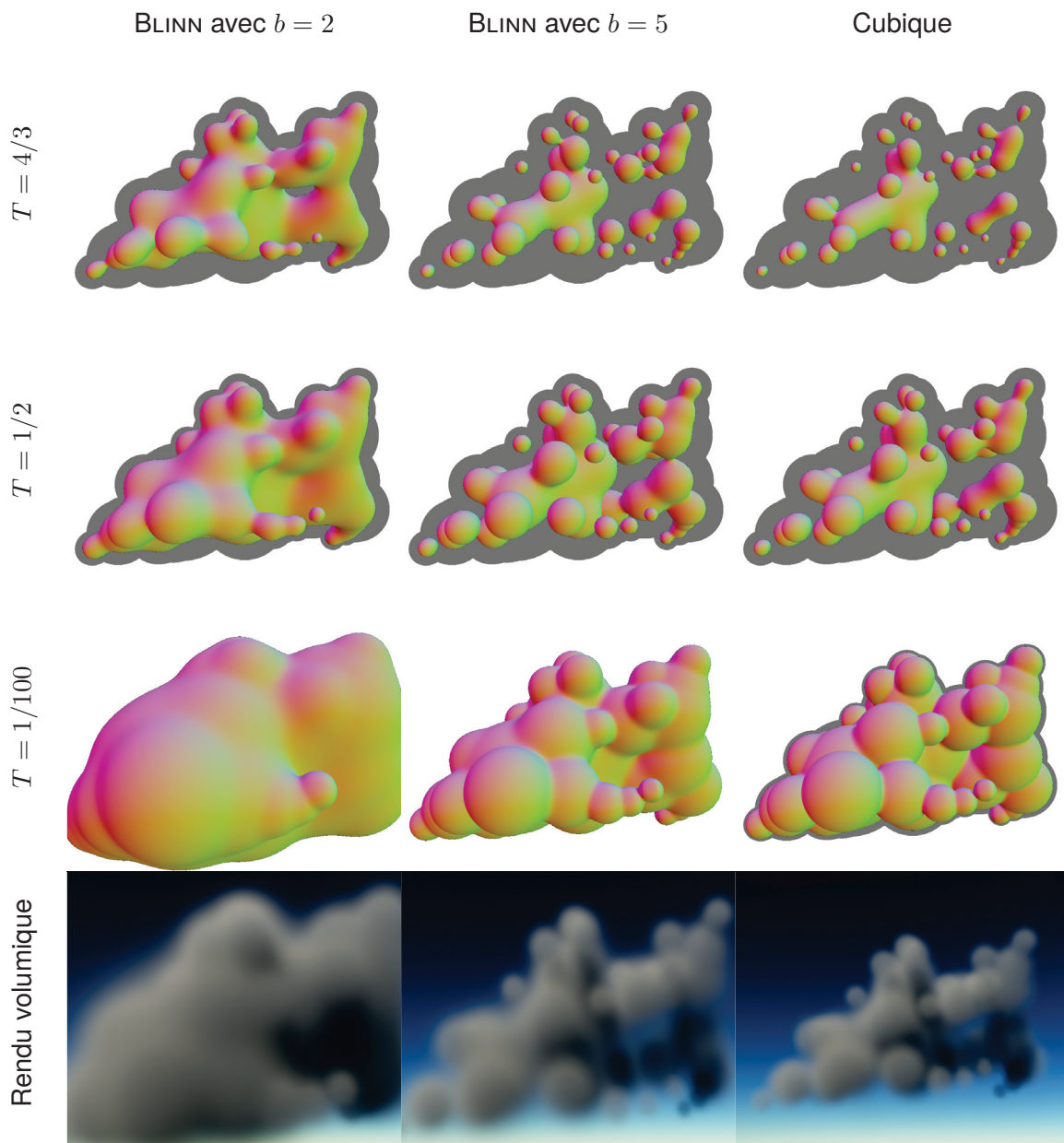
Le paramètre  $b$  contrôle la raideur du profil d’atténuation : plus  $b$  est faible, plus la pente est douce, et vice versa (figure 2.6 page 71).

**Définition 10 (Support d’une fonction)** Soit une fonction  $f : E \rightarrow \mathbb{R}$  ; son support  $\Omega$  est la partie de son domaine de définition  $\Omega \subseteq E$  constituée des points en lesquels la fonction ne s’annule pas

$$\Omega = \{\mathbf{p} \in E \mid f(x) \neq 0\}. \quad (2.11)$$

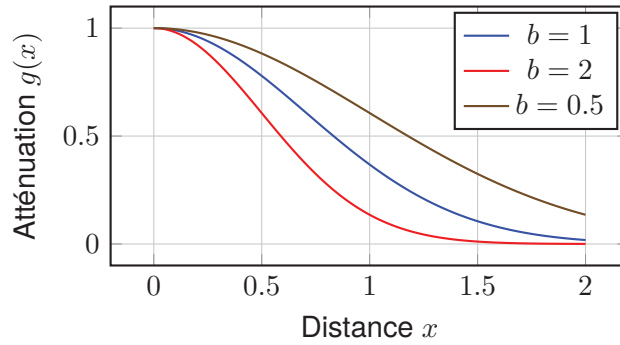
L’exponentielle qui intervient dans la fonction d’atténuation de BLINN lui confère un support infini. Toute primitive implicite à squelette qui utilise cette fonction d’atténuation a donc un rayon d’influence infini, même si son influence décroît exponentiellement avec la distance. En tout point, l’évaluation du champ de potentiel d’un mélange de primitives utilisant la fonction d’atténuation de BLINN requiert alors d’évaluer les champs de potentiel de toutes les primitives. En revanche, l’exponentielle garantit un mélange lisse de classe  $C^\infty$ .



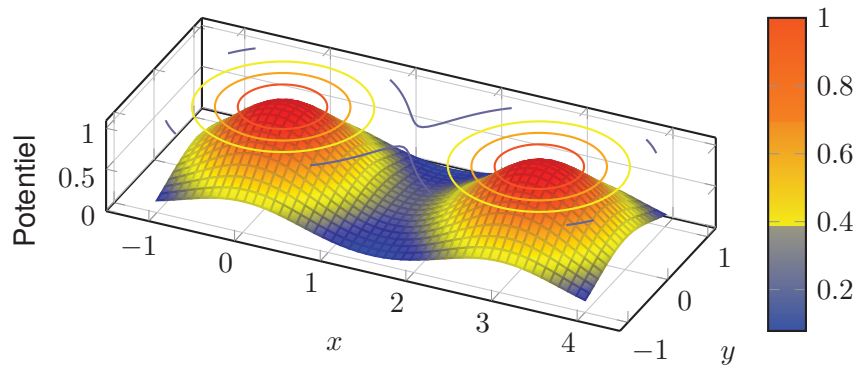


**FIGURE 2.5.** – Visualisation du comportement d'un mélange de primitives sphériques utilisant différentes fonctions d'atténuation. De haut en bas : normales des isosurfaces pour différentes isovaleurs (le fond représente l'union des primitives sphériques qui restreignent le support du mélange dans le cas de l'atténuation cubique), et un rendu du champ de potentiel comme densité nuageuse.

La figure 2.7 page suivante et la figure 2.5 montrent des exemples de mélanges de primitives implicites à squelettes ponctuels utilisant la fonction d'atténuation de BLINN et que BLINN appelle des *blobs*.



**FIGURE 2.6.** – Visualisation de la fonction d’atténuation de BLINN (équation 2.10 page 69) pour différentes raideurs  $b$ .



**FIGURE 2.7.** – Visualisation de l’influence de deux primitives ponctuelles utilisant l’atténuation de BLINN (équation 2.10 page 69). Sont illustrés le potentiel scalaire et la forme prise par la courbe implicite pour différentes isovalues. Le paramètre de raideur est  $b = 1$  et les primitives ont pour centres  $C_0 = (0, 0)$  et  $C_1 = (3, 0)$ .

### Fonctions d’atténuation à support restreint

Afin de limiter l’influence d’une primitive à squelette dans un mélange, il est possible d’utiliser des fonctions d’atténuation à support restreint. MURAKAMI *et al.* [101] et WYVILL *et al.* [151] proposent des fonctions définies par un morceau polynomial décroissant entre zéro et le seuil de distance du rayon d’influence, et nulles au-delà.

**Définition 11 (Fonction d’atténuation de MURAKAMI)** Soit  $x$  la variable de distance au squelette et  $R$  le rayon d’influence ; la fonction d’atténuation de MURAKAMI est :

$$g(x) = \begin{cases} (1 - r^2)^2 & \text{si } r \in [0, 1] \\ 0 & \text{sinon} \end{cases} \quad \text{où } r = \frac{x}{R}. \quad (2.12)$$

**Définition 12 (Fonction d’atténuation de WYVILL)** Soit  $x$  la variable de distance au squelette et  $R$  le rayon d’influence ; la fonction d’atténuation de WYVILL est :

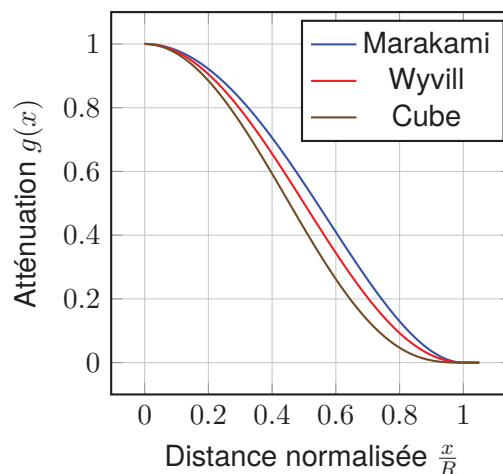
$$g(x) = \begin{cases} -\frac{4}{9}r^6 + \frac{17}{9}r^4 - \frac{22}{9}r^2 + 1 & \text{si } r \in [0, 1] \\ 0 & \text{sinon} \end{cases} \quad \text{où } r = \frac{x}{R} \quad (2.13)$$

et vérifie  $g(x) + g(1 - x) = 1$  sur son support, ce qui fait du point  $(0.5, 0.5)$  un centre de symétrie de la fonction.

Visuellement, les fonctions d'atténuation de MARAKAMI et WYVILL diffèrent légèrement (figure 2.8), mais sur leur support, la fonction d'atténuation de WYVILL est de degré six, soit deux degrés de plus que la fonction d'atténuation de MARAKAMI.

**Définition 13 (Fonction d'atténuation de puissance généralisée)** La fonction d'atténuation de MARAKAMI peut être généralisée à une puissance quelconque, afin notamment de garantir une meilleure classe de complexité. Soit  $x$  la variable de distance au squelette et  $R$  le rayon d'influence ; la fonction d'atténuation de puissance généralisée est :

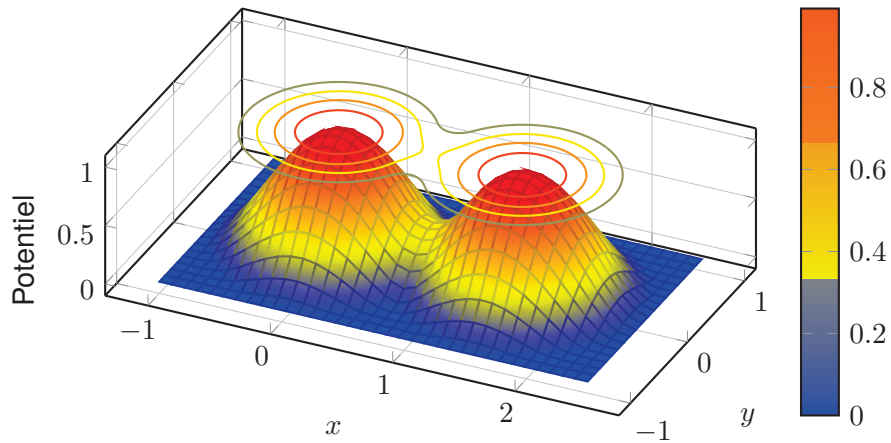
$$g(x) = \begin{cases} (1 - r^2)^n & \text{si } r \in [0, 1] \\ 0 & \text{sinon} \end{cases} \quad \text{où } r = \frac{x}{R}. \quad (2.14)$$



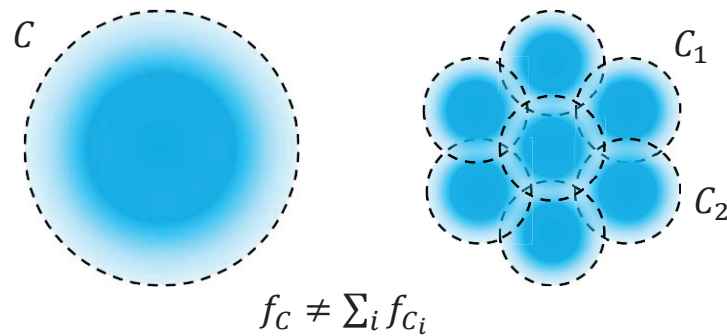
**FIGURE 2.8.** – Visualisation des fonctions d'atténuation de MARAKAMI (équation 2.12 page précédente), de WYVILL (équation 2.13 page précédente), et de puissance trois (équation 2.14).

## 2.1.5 Décomposition de primitive implicite en mélange équivalent

Dans le cadre de l'animation du champ de présence nuageuse (chapitre 4 page 103), nous transformons un mélange de primitives implicites en un autre par métamorphose. Pour ce faire, il est nécessaire de pouvoir décomposer certaines primitives en de multiples parties où chacune est animée indépendamment, tout en respectant les champs implicites initiaux et finaux.



**FIGURE 2.9.** – Visualisation de l’influence limitée de deux primitives ponctuelles utilisant la fonction d’atténuation de MARAKAMI (équation 2.12 page 71) avec rayon d’influence. Sont illustrés le potentiel scalaire et la forme prise par la courbe implicite pour différentes isovaleurs. Les primitives ont un rayon d’influence unitaire et pour centre  $C_0 = (0, 0)$  et  $C_1 = (1.5, 0)$ .



**FIGURE 2.10.** – Schématisation d’une approximation de discrétisation d’une primitive implicite sphérique en un mélange de plus petites primitives.

De nombreuses décompositions peuvent être envisagées. Une approche géométrique consisterait par exemple à discrétiser une primitive d’une certaine taille en plus petites primitives. Il se révèle cependant impossible de discrétiser une primitive sphérique en un mélange d’un nombre fini de plus petites primitives sphériques (figure 2.10) sans erreur d’approximation du champ de potentiel. D’un autre côté, nous pouvons simplement décomposer une primitive, quelle que soit sa forme, en un mélange de répliques pondérées par des facteurs d’intensité maximum dont la somme est unitaire. Soit  $f$  le champ de potentiel d’une primitive implicite, et soit  $(D_i)$  les intensités maximum d’une famille de primitives pondérées ; alors :

$$f = \sum_i (D_i \cdot f) \quad \text{si et seulement si} \quad \sum_i D_i = 1. \quad (2.15)$$

### 2.1.6 Conclusion

Les primitives implicites sphériques permettent de modéliser assez facilement le support d’un nuage cumuliforme (figure 2.5 page 70), mais utiliser le champ de potentiel directement comme densité nuageuse produit une forme aux bords flous qui manque de détails. Dans

notre modèle de paysage nuageux (chapitre 3 page 87), nous modélisons plutôt une couche nuageuse qu'un nuage cumuliforme, et nous remplaçons les primitives sphériques par des primitives ellipsoïdales et nous combinons leur champ de potentiel avec du bruit procédural pour ajouter du détail.

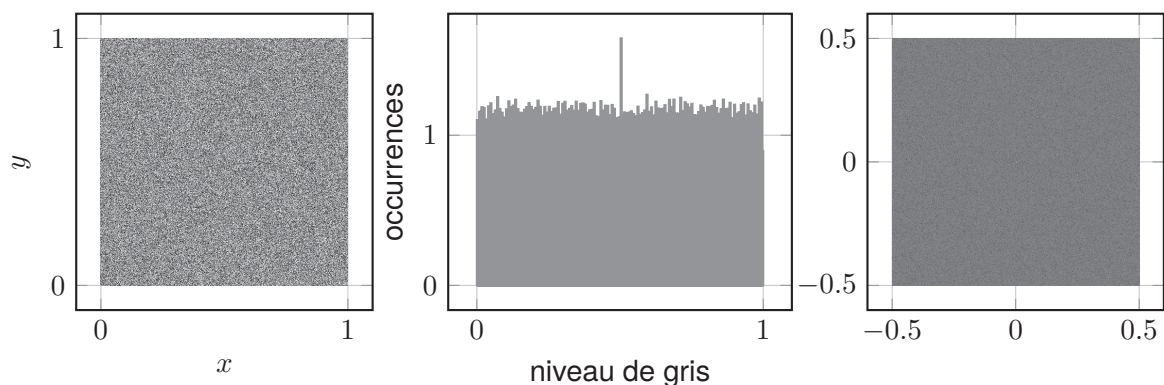


## 2.2 Bruits procéduraux

Dans le contexte de la génération procédurale, le bruit est une fonction  $\mathbb{R} \rightarrow [0, 1]$  construite à partir de valeurs aléatoires ou pseudo-aléatoires, afin d'assurer une certaine classe de continuité et de présenter des fréquences bornées. Le domaine d'un bruit n'est pas nécessairement le temps et n'est pas nécessairement ni continu, ni périodique comme l'est une fonction sinusoïdale avec ses oscillations à fréquence unique. Une image à niveau de gris peut par exemple être vue comme l'échantillonnage d'un bruit depuis le domaine spatial dont le support est limité par les dimensions de l'image, vers l'intervalle de niveau de gris. La notion de fréquence est alors plutôt spatiale que temporelle.

Nous rappelons la définition de plusieurs types de bruits classiques que nous combinons par la suite afin de modéliser les détails de différents aspects nuageux (section 3.3 page 96). La plupart des fonctions de bruits ont pour domaine de définition  $\mathbb{R}$ , mais leurs expressions sont aisément adaptables à un domaine de définition multidimensionnel comme  $\mathbb{R}^3$  pour un bruit volumique ou  $\mathbb{R}^4$  pour un bruit spatiotemporel.

**Bruit blanc uniforme** Un exemple trivial de bruit est le bruit blanc uniforme (figure 2.11). En tout point, un exemplaire de bruit blanc uniforme est défini comme une réalisation d'une variable aléatoire suivant la loi de probabilité uniforme sur  $[0, 1]$ . Ce bruit est qualifié de blanc car il ne présente pas de cohérence fréquentielle, *id est* il n'y a pas de corrélation spatiale des variables aléatoires. Il est aussi qualifié d'uniforme car l'intensité du signal suit la loi de probabilité uniforme.



**FIGURE 2.11.** – Un exemplaire de bruit blanc uniforme 2D (gauche), son histogramme (milieu) et son spectre de fréquence spatiale (droite).

### 2.2.1 Bruits à bande fréquentielle restreinte

Le bruit blanc ne présente aucune cohérence fréquentielle (figure 2.11), aucune structure exploitable afin de [re]produire un motif particulier. Pour cette raison, son intérêt est plutôt limité en synthèse procédurale et la plupart des bruits sont plutôt construits de manière à contrôler leurs fréquences, guidant ainsi la taille des éléments des motifs. Pour ce faire, il est commun de construire une fonction continue par morceaux entre les sommets d'un treillis tel qu'une grille régulière dont la maille guide la taille des éléments du bruit. La continuité de la fonction de bruit est obtenue par convolution avec une fonction d'interpolation.

**Produit de convolution** Le produit de convolution est un opérateur mathématique qui à deux fonctions définies sur le même domaine, fait correspondre une troisième fonction sur le même domaine qui est l'intégrale de leur produit. Soit  $f$  et  $g$  deux fonctions de  $E \rightarrow F$ ; leur convolution est définie par

$$(f * g)(x) = \int_E f(t)g(x-t) dt = \int_E f(x-t)g(t) dt. \quad (2.16)$$

En tout point, le produit de convolution de deux fonctions peut se visualiser comme l'aire commune sous la courbe des deux fonctions dont les domaines sont décalés d'autant que le point l'est de l'origine. Si l'une des deux fonctions d'un produit de convolution a un support limité, l'intégrale du produit peut être simplifiée. Cela revient à faire glisser une fenêtre de taille réduite, *id est* limitée au support de la fonction, sur le support de l'autre fonction.

Dans le cas de fonctions définies sur un domaine discret, les intégrales sont simplement remplacées par des sommes, et la fenêtre glissante est appelée matrice de convolution ou noyau de convolution. L'application partielle de l'opérateur de convolution au noyau est appelé filtre de convolution, et son application finale à une fonction discrète telle qu'une image est une opération de filtrage, *id est* la transformation d'un signal discret en un autre signal discret.

**Interpolation sur la grille régulière** Il est possible d'exprimer différentes méthodes d'interpolation de points de contrôle placés selon les sommets d'une grille sous la forme d'un produit de convolution en généralisant le peigne de DIRAC unidimensionnel à une grille de DIRAC n-dimensionnelle. L'idée consiste à construire une distribution qui, à la convolution, transmette les valeurs des points de contrôle aux fonctions de la base interpolante. Pour ce faire, une impulsion de DIRAC est translatée en chaque point de contrôle de la grille régulière et pondérée par sa valeur associée  $v_i$  :

$$\mathcal{V}(\mathbf{p}) = \sum_i \delta(\mathbf{p} - \mathbf{p}_i) \cdot v_i. \quad (2.17)$$

Et la grille de DIRAC pondérée est finalement convoluée avec une fonction  $\mathcal{I}$  d'interpolation par convolution pour produire la fonction interpolante ( $\mathcal{I} * \mathcal{V}$ ).

Ainsi, la fonction de convolution pour l'interpolation constante par morceau s'exprime par :

$$\mathcal{I}_0(t) = \begin{cases} 1 & \text{si } t \in [-0.5, 0.5[ \\ 0 & \text{sinon} \end{cases}. \quad (2.18)$$

Et la fonction de convolution pour l'interpolation linéaire s'exprime par :

$$\mathcal{I}_1(t) = \begin{cases} 1+t & \text{si } t \in [-1, 0[ \\ 1-t & \text{si } t \in [0, 1[ \\ 0 & \text{sinon} \end{cases}. \quad (2.19)$$

La fonction d'interpolation linéaire permet ensuite d'exprimer la fonction de convolution pour l'interpolation  $n$ -linéaire :

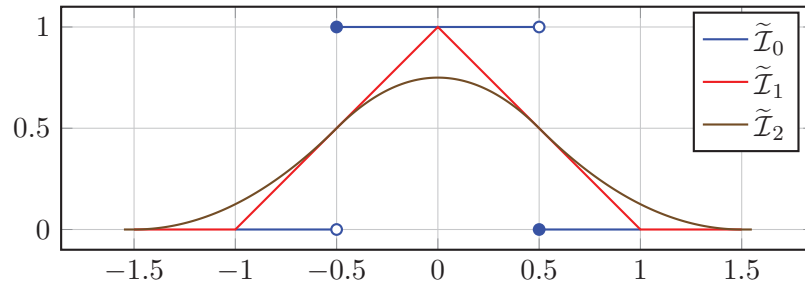
$$\mathcal{I}_1^n(\mathbf{t}) = \mathcal{I}_1(\mathbf{t}[0]) \cdots \mathcal{I}_1(\mathbf{t}[n-1]). \quad (2.20)$$

**Approximation par convolution** Il est possible de continuer d'augmenter le degré de la fonction de convolution en l'intégrant récursivement sur  $[-0.5, 0.5]$  d'après le schéma des B-splines centrées. Cela a pour conséquence d'augmenter la taille du support et de couvrir un point de la grille en plus. En augmentant le degré ainsi après l'interpolation linéaire, la convolution n'est plus à proprement parler une interpolation mais plutôt une approximation, car la fonction qui résulte de la convolution ne passe plus nécessairement par les points de contrôle.

$$\tilde{\mathcal{I}}_0^n(\mathbf{t}) = \mathcal{I}_0^n(\mathbf{t}), \quad (2.21)$$

$$\tilde{\mathcal{I}}_c^n(\mathbf{t}) = \left( \tilde{\mathcal{I}}_{c-1}^n * \mathcal{I}_0^n \right) (\mathbf{t}) \quad (2.22)$$

$$= \int_{-0.5}^{0.5} \tilde{\mathcal{I}}_{c-1}^n(\mathbf{t} - \mathbf{p}) \, d\mathbf{p}. \quad (2.23)$$



**FIGURE 2.12.** – Visualisation des fonctions de la base B-splines pour l'approximation par convolution.

De manière similaire, il est possible d'interpoler avec une fonction de convolution utilisant la fonction cosinus :

$$\mathcal{I}_{\cos}(t) = \begin{cases} \frac{\cos(t \cdot \pi) + 1}{2} & \text{si } t \in [-1, 1] \\ 0 & \text{sinon} \end{cases}, \quad (2.24)$$

$$\mathcal{I}_{\cos}^n(\mathbf{t}) = \mathcal{I}_{\cos}(\mathbf{t}[0]) \cdots \mathcal{I}_{\cos}(\mathbf{t}[n-1]). \quad (2.25)$$

Ou encore avec le polynôme  $f_{ss}(x) = 3x^2 - 2x^3$  correspondant à la fonction très répandue *smoothstep* :

$$\mathcal{I}_{ss}(t) = \begin{cases} 1 - f_{ss}(-t) & \text{si } t \in [-1, 0[ \\ 1 - f_{ss}(t) & \text{si } t \in [0, 1[ \\ 0 & \text{sinon} \end{cases} \quad (2.26)$$

$$= \begin{cases} -2t^3 - 3t^2 + 1 & \text{si } t \in [-1, 0[ \\ 2t^3 - 3t^2 + 1 & \text{si } t \in [0, 1[ \\ 0 & \text{sinon} \end{cases}, \quad (2.27)$$

$$\mathcal{I}_{ss}^n(\mathbf{t}) = \mathcal{I}_{ss}(\mathbf{t}[0]) \cdots \mathcal{I}_{ss}(\mathbf{t}[n-1]). \quad (2.28)$$

Ainsi que la quintique  $f_q(x) = 6x^5 - 15x^4 + 10x^3$  :

$$\mathcal{I}_q(t) = \begin{cases} 1 - f_q(-t) & \text{si } t \in [-1, 0[ \\ 1 - f_q(t) & \text{si } t \in [0, 1[ \\ 0 & \text{sinon} \end{cases} \quad (2.29)$$

$$= \begin{cases} 6x^5 + 15x^4 + 10x^3 + 1 & \text{si } t \in [-1, 0[ \\ -6x^5 + 15x^4 - 10x^3 + 1 & \text{si } t \in [0, 1[ \\ 0 & \text{sinon} \end{cases}, \quad (2.30)$$

$$\mathcal{I}_q^n(\mathbf{t}) = \mathcal{I}_q(\mathbf{t}[0]) \cdots \mathcal{I}_q(\mathbf{t}[n-1]). \quad (2.31)$$

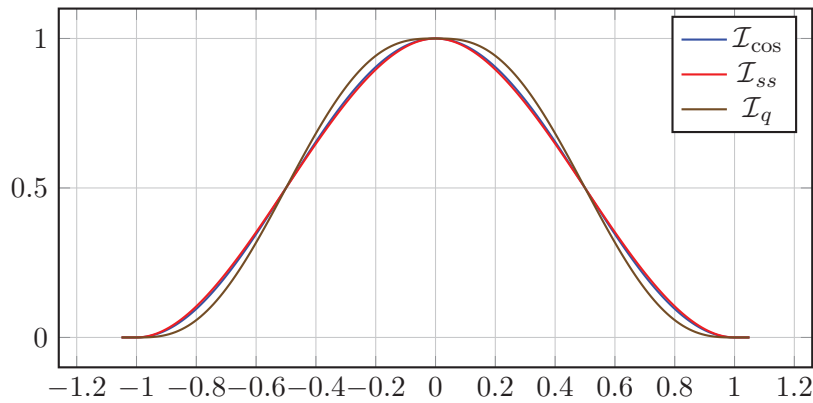


FIGURE 2.13. – Visualisation des fonctions d'interpolation par convolution.

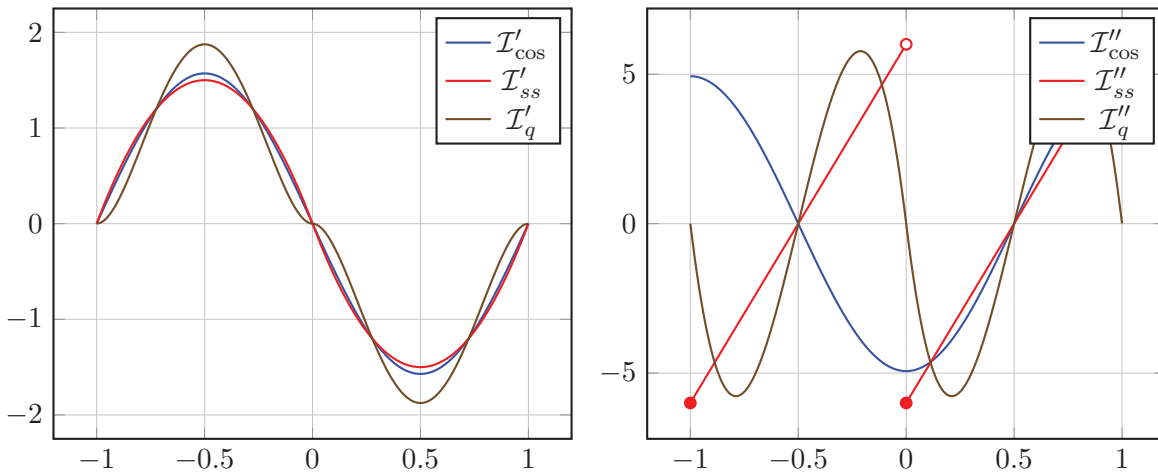


FIGURE 2.14. – Visualisation des dérivées des fonctions d'interpolation par convolution.

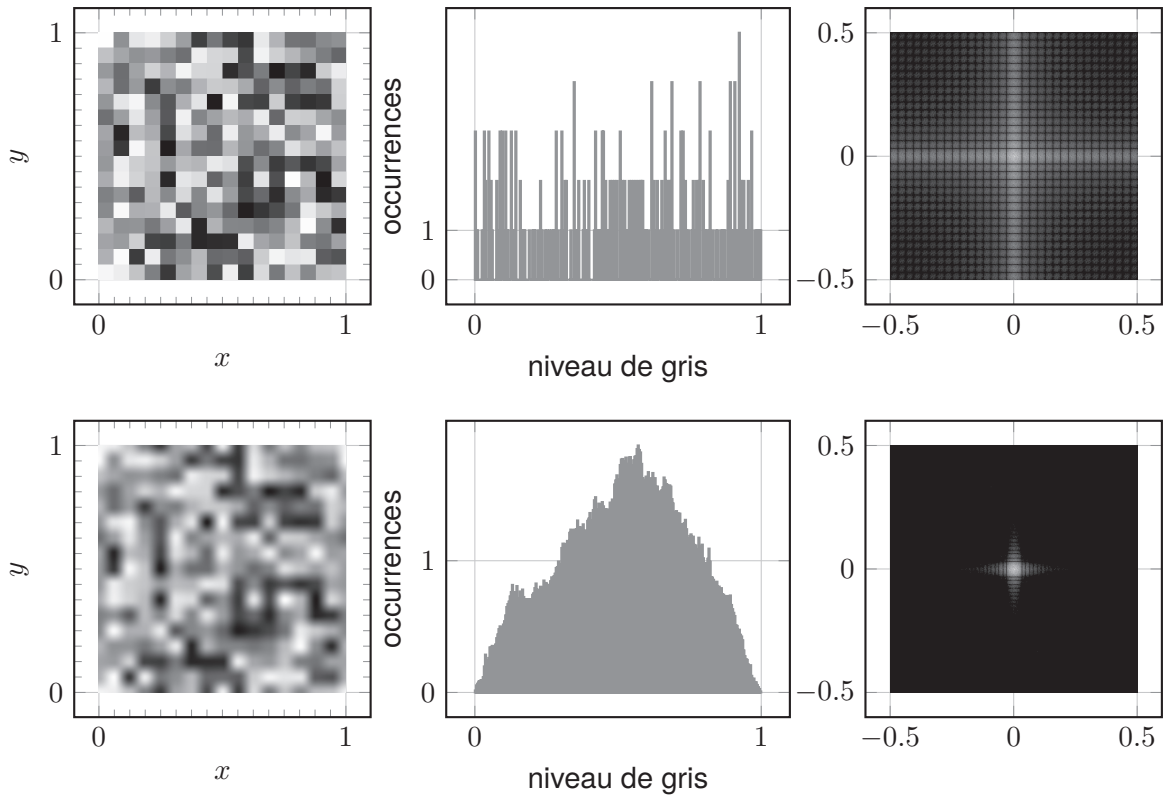
## Bruit de valeur

Le bruit de valeur est construit à partir de valeurs tirées aléatoirement, placées selon une grille régulière et interpolées. Soit  $n$  le nombre de dimensions,  $\mathcal{I}^n$  une fonction  $n$ -aire d'interpolation par convolution,  $r$  et  $\mathbf{p}_0$  la taille de la maille et l'origine de la grille, et  $v_i$  un ensemble de variables aléatoires suivant la loi uniforme  $U(0, 1)$  qui servent de pondération à

la grille de DIRAC ; une expression basique du bruit de valeur consiste alors en l'interpolation des valeurs placées sur la grille :

$$\mathbb{R}^n \rightarrow [0, 1] \quad (2.32)$$

$$\mathbf{p} \mapsto (\mathcal{I}^n * \mathcal{V}) \left( \frac{\mathbf{p}}{r} - \mathbf{p}_0 \right). \quad (2.33)$$



**FIGURE 2.15.** – Un exemplaire de bruit de valeur sur une grille 2D de fréquence égale à 16, sans interpolation (haut) et avec interpolation de type *smoothstep*  $\mathcal{I}_{ss}$  (bas).

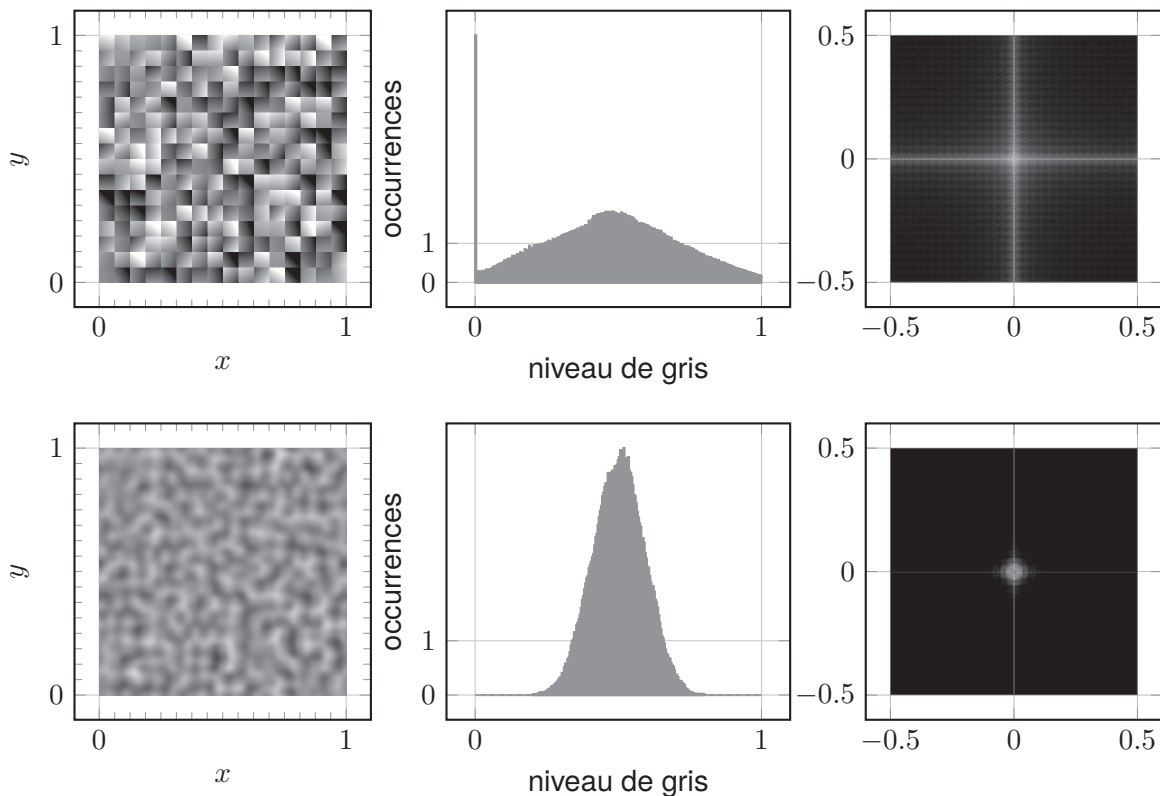
## Bruit de gradient

De manière similaire au bruit de valeur, le bruit de gradient est construit à partir de tirages aléatoires qui sont interpolés selon une grille régulière. En revanche, plutôt que de placer des valeurs scalaires sur la grille, ce sont des vecteurs unitaires (directions) aléatoires qui sont placés sur la grille. Afin d'obtenir une valeur scalaire en retour, les directions ne sont pas interpolées directement. La case de la grille associée au point est calculée et pour chaque coin de la case et la direction associée est multipliée par produit scalaire avec le vecteur entre le point et le coin. Pour finir, les valeurs scalaires obtenues sont interpolées, comme dans le cas du bruit de valeur. Soit  $n$  le nombre de dimensions,  $\mathcal{I}^n$  une fonction  $n$ -aire d'interpolation par convolution,  $r$  la taille de la maille,  $\mathbf{p}_0$  l'origine de la grille, et  $\omega_i$  des



directions tirées aléatoirement; les valeurs qui pondèrent la grille de DIRAC sont exprimées par  $v_i = \omega_i \cdot (\mathbf{p} - (\mathbf{p}_0 + r \cdot \mathbf{i}))$ , et le bruit de gradient peut s'écrire

$$\begin{aligned} \mathbb{R}^n &\rightarrow [0, 1] \\ \mathbf{p} &\mapsto (\mathcal{I}^n * \mathcal{V}) \left( \frac{\mathbf{p}}{r} - \mathbf{p}_0 \right). \end{aligned} \tag{2.34}$$



**FIGURE 2.16.** – Un exemplaire de bruit de gradient sur une grille 2D de fréquence égale à 16, sans interpolation (haut) et avec interpolation quintique  $\mathcal{I}_q$  (bas).

**Bruit simplexe** Le bruit simplexe est un bruit de gradient construit sur une grille dont la cellule n'est plus un hypercube mais un simplexe, *id est* un triangle en 2D et un tétraèdre en 3D. Ainsi, dans le cas d'un espace à  $n$  dimensions, l'interpolation à calculer est simplifiée. Dans l'interpolation  $n$ -aire, il n'y a plus besoin d'interpoler  $2^n$  éléments mais seulement  $n + 1$ , ce qui passe beaucoup mieux à l'échelle.

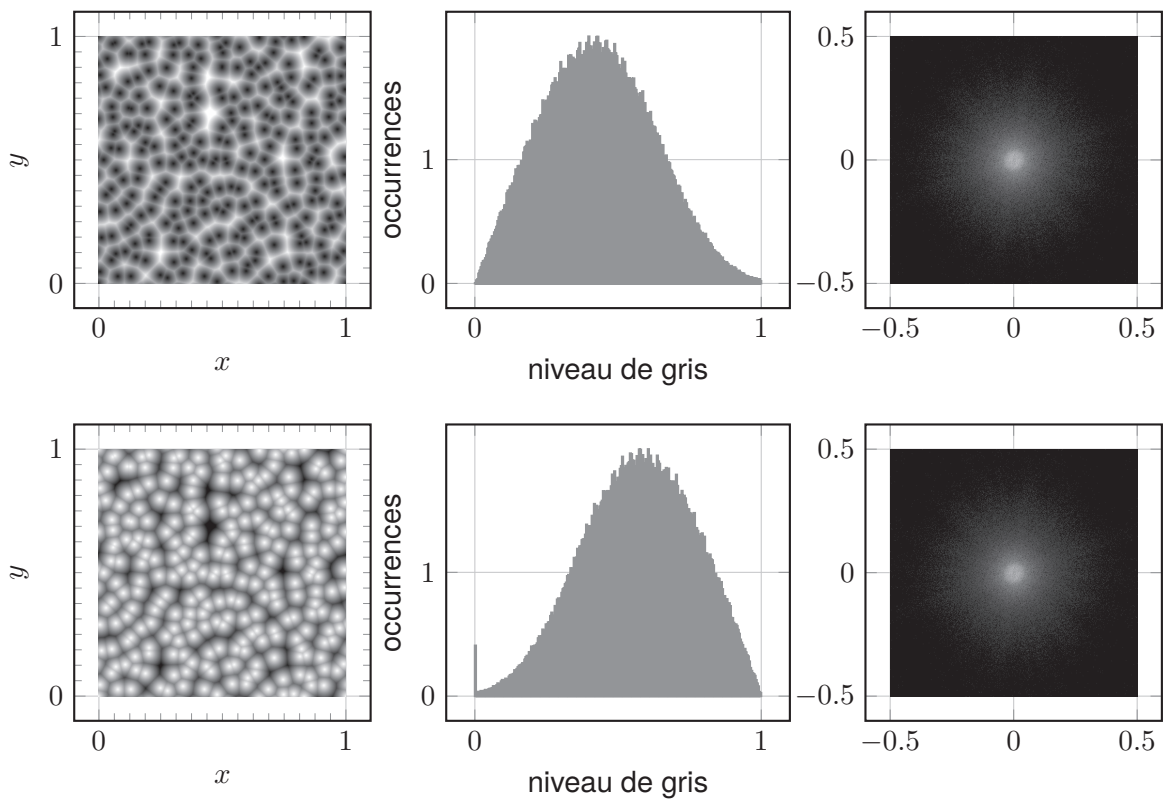
### Bruit cellulaire

Le bruit cellulaire, aussi appelé bruit de WORLEY en hommage à son créateur, est un bruit qui ne demande plus d'interpoler les valeurs d'une grille. Le bruit cellulaire peut être vu comme le champ de potentiel d'une union de primitives implicites à squelette ponctuel dont le nombre et la position sont tirés aléatoirement. La structure de grille n'est pas strictement nécessaire mais permet de limiter la complexité de l'évaluation du bruit cellulaire en ne considérant qu'une quantité limitée de primitives voisines dans l'expression du champ de distance.

Soit  $d$  une fonction de distance et  $\mathbf{p}_i$  le centre des primitives du voisinage, le bruit cellulaire peut s'écrire

$$\begin{aligned} \mathbb{R}^n &\rightarrow [0, 1] \\ \mathbf{p} &\mapsto \max_i d(\mathbf{p}, \mathbf{p}_i) \end{aligned} \quad (2.35)$$

En tant que champ de distance, le bruit cellulaire a toujours une valeur positive. Si la fonction de distance utilisée pour définir le champ est la distance euclidienne, le bruit cellulaire laisse apparaître des cellules de Voronoï et il est possible d'utiliser différentes fonctions de distance pour varier les motifs, ou bien de calculer le complément à un du bruit cellulaire (figure 2.17).



**FIGURE 2.17.** – Un exemplaire de bruit cellulaire (haut) et son complémentaire (bas) avec distance euclidienne sur une grille 2D de fréquence égale à 16 avec .

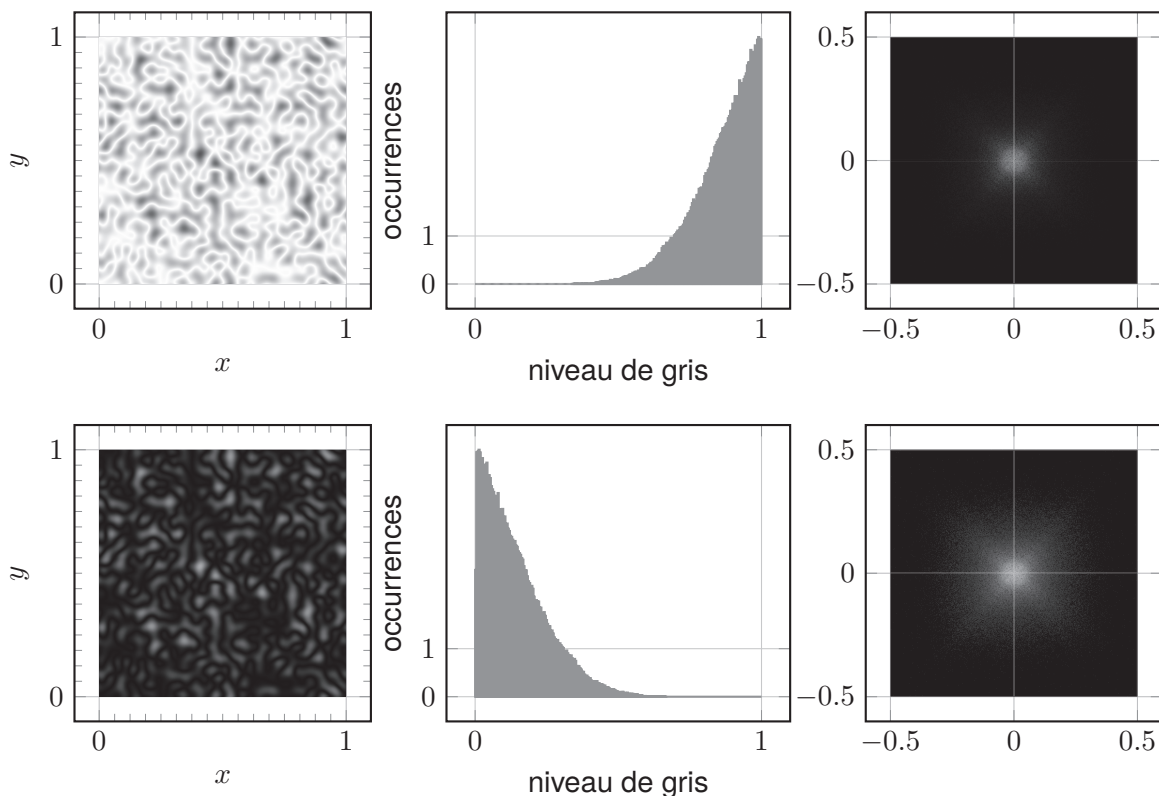
Le bruit cellulaire permet notamment de reproduire la texture nuageuse des *Cumulus* bourgeonnants, tandis que son complémentaire donne plutôt une apparence de nuage qui s'effiloche, tel que l'illustre la figure 2.22 page 86.

### Bruit de crêtes

Les bruits de valeur et de gradient produisent des motifs lisses, mais des discontinuités sont parfois désirables pour créer des crêtes montagneuses par exemple. C'est l'idée derrière le bruit de crête (*ridge noise*) qui s'obtient classiquement en appliquant la fonction valeur absolue à un bruit  $f$  dont les valeurs appartiennent à l'intervalle  $[-1, 1]$ ,

ce qui produit des discontinuités aux points où le bruit  $f$  passe par zéro, puis en prenant le complément à un pour placer les discontinuités aux sommets plutôt qu'aux vallées (figure 2.18) :

$$\begin{aligned} \mathbb{R}^n &\rightarrow [0, 1] \\ \mathbf{p} &\mapsto 1 - |f(\mathbf{p})| \end{aligned} \quad (2.36)$$



**FIGURE 2.18.** – Un exemplaire de bruit de crête (haut) et son complémentaire (bas) à partir de bruit de gradient avec interpolation quintique sur une grille 2D de fréquence égale à 16.

## 2.2.2 Bruits à multiples bandes fréquentielles

Par construction, les bruits simples tels que le bruit de valeur ou le bruit de gradient présentent des fréquences limitées. En effet, la taille de la maille de la grille à partir de laquelle ils sont calculés contrôle directement leur spectre : une maille fine correspond à une bande de plus haute fréquence et inversement. Afin de créer des motifs plus complexes, il est possible de combiner un bruit et différentes copies de lui-même (ou d'autres types de bruits) à différentes fréquences. Une manière triviale de combiner plusieurs bruits tout en conservant l'intervalle de leurs valeurs est d'en faire la moyenne. Malheureusement, accorder autant d'importance à toutes les composantes tend à produire un bruit de type bruit blanc car les motifs se chevauchent avec la même amplitude et le chaos augmente rapidement. Il est cependant possible d'éviter cela en remplaçant la moyenne des composantes par une moyenne pondérée qui favorise certaines fréquences.

## Bruit fractal

Afin d'éviter de retomber sur un bruit de type bruit blanc, l'intensité de la suite de bruits est pondérée par un facteur d'amplitude qui réduit l'influence des bruits de haute fréquence. Le type de bruit ainsi obtenu est appelé bruit brownien fractionnaire (fbm) ou bruit fractal car composé de motifs qui se répètent à différentes échelles, pondérés par un facteur d'amplitude. De plus, bien qu'il soit possible de sommer n'importe quelle combinaison de bruits pour obtenir un bruit fractal, il est courant de construire une suite régulière. Dans ce cas, les paramètres de la suite de bruits sont décrits par des suites géométriques simples. La raison de la suite de fréquences est appelée lacunarité et la raison de la suite de facteurs d'amplitude est appelée persistance. Soit  $n^d$  une fonction de bruit sur la grille canonique à  $d$  dimensions,  $f_0$  la fréquence de départ,  $l$  la lacunarité,  $a_0$  l'amplitude de départ et  $\rho$  la persistance ; la suite de bruits associée est

$$n_n(\mathbf{p}) = a_n \cdot n^d(\mathbf{p} \cdot f_n) \quad (2.37)$$

$$= a_0 \cdot \rho^n \cdot n^d(\mathbf{p} \cdot f_0 \cdot l^n). \quad (2.38)$$

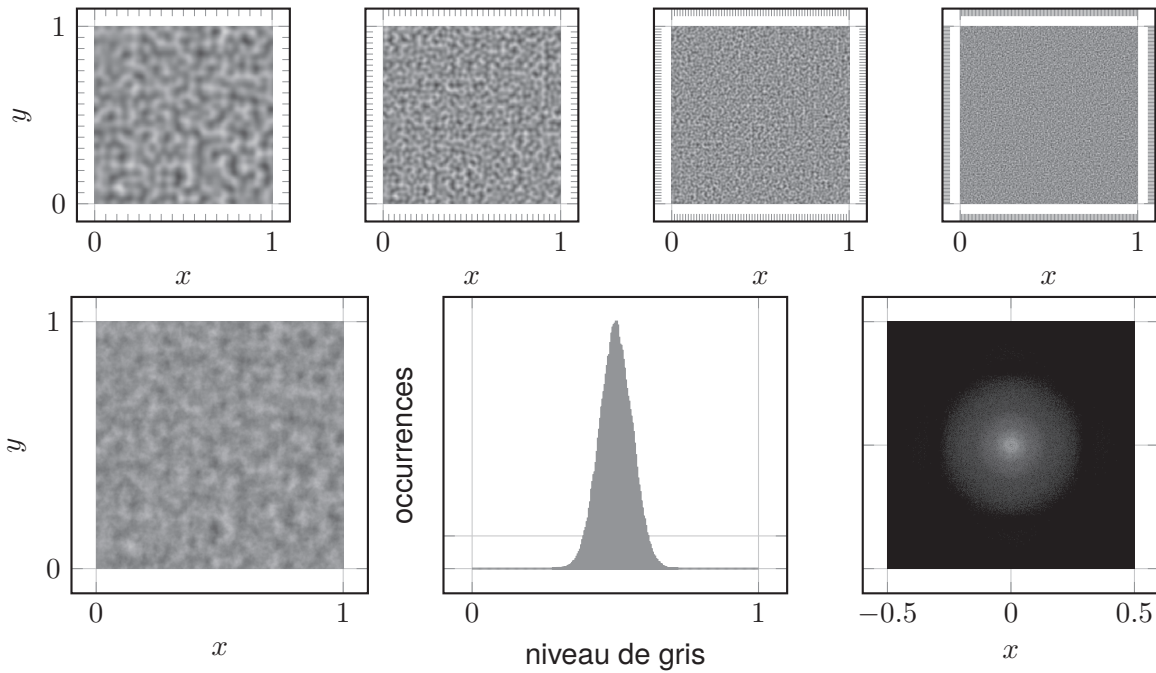
Le bruit fractal s'exprime comme la somme des termes de la suite de bruits. Il est impossible d'évaluer la somme infinie des bruits composant le bruit fractal. Heureusement, cela n'est pas nécessaire car le nombre de termes à sommer peut être calculé en fonction du niveau de détail nécessaire et de la lacunarité. Bien que cela corresponde plutôt aux signaux dont la fréquence est multiple de deux, il est commun de désigner un bruit de la somme par le terme d'octave, quelle que soit la lacunarité du bruit fractal. Soit  $o$  le nombre d'octaves à calculer ; le bruit fractal s'exprime par

$$n_{frac}(\mathbf{p}) = \sum_{n=0}^{o-1} n_n(\mathbf{p}). \quad (2.39)$$

## Bruit de turbulence

Plutôt que de sommer directement une suite de bruits comme dans le bruit fractal, une variante consiste à sommer leur valeur absolue. Le bruit ainsi obtenu est appelé bruit de turbulence par PERLIN [118]. Soit  $o$  le nombre d'octaves à calculer ; le bruit de turbulence s'exprime par

$$n_{turb}(\mathbf{p}) = \sum_{n=0}^{o-1} |n_n(\mathbf{p})|. \quad (2.40)$$



**FIGURE 2.19.** – Du bruit fractal (bas) composé de quatre octaves (haut) de bruit de gradient sur une grille 2D avec interpolation quintique  $\mathcal{I}_q$ .

### Bruit fractal de crête

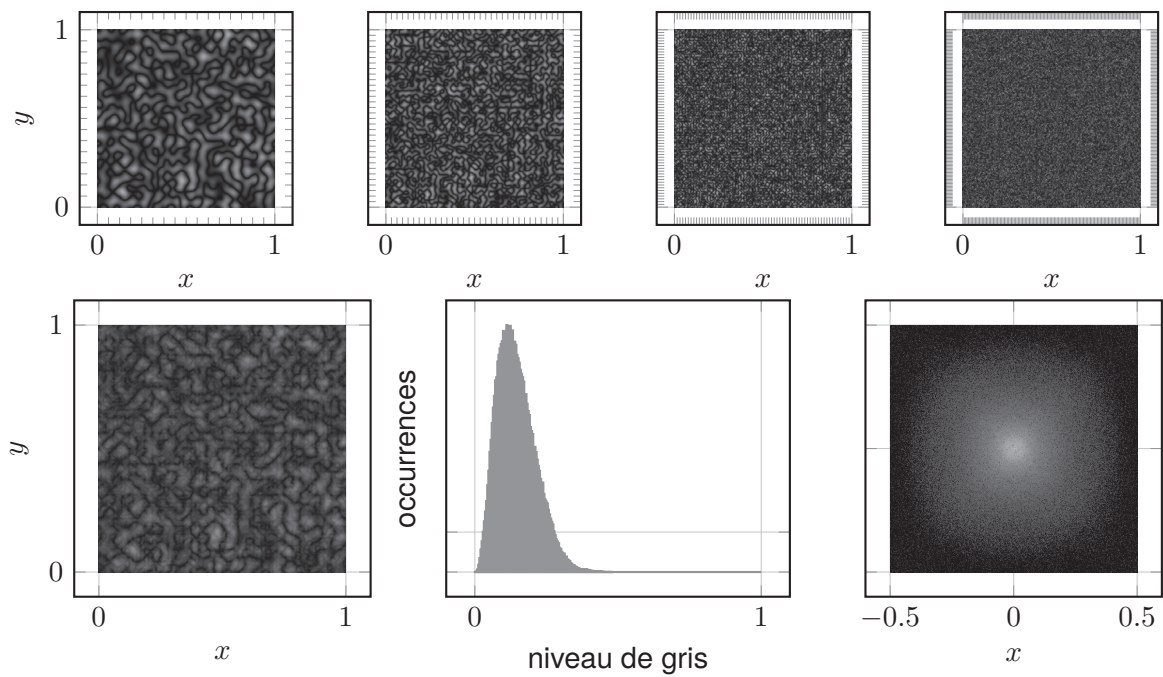
En appliquant à la fois le principe du bruit fractal et du bruit de crête, nous obtenons encore un autre motif que nous appelons bruit fractal de crête. Soit  $o$  le nombre d'octaves ; le bruit fractal de crête s'exprime par :

$$n_{rfn}(\mathbf{p}) = \sum_{n=0}^{o-1} 1 - |n_n(\mathbf{p})|. \quad (2.41)$$

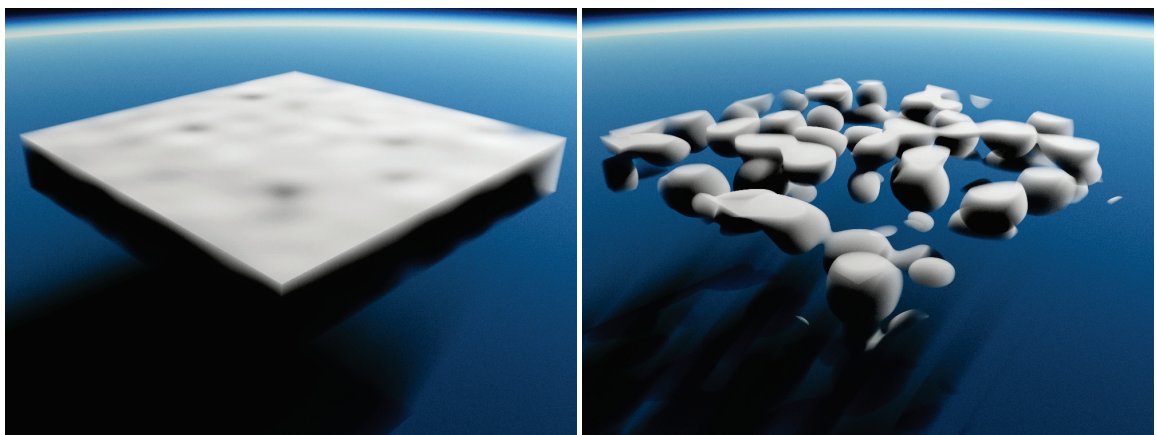
### 2.2.3 Conclusion

Les différents bruits procéduraux dont nous avons rappelé les définitions nous permettent de produire des motifs nuageux en les interprétant comme champ de densité. Afin d'éviter que la densité nuageuse ne couvre tout le domaine, les valeurs sous un certain seuil sont ramenées à zéro, ce qui produit des amas nuageux dont les motifs sont visuellement exploitables (figure 2.21 page ci-contre). La figure 2.22 page 86 illustre un exemple simple de motif nuageux obtenu par bruit fractal combinant des octaves de bruit cellulaire.

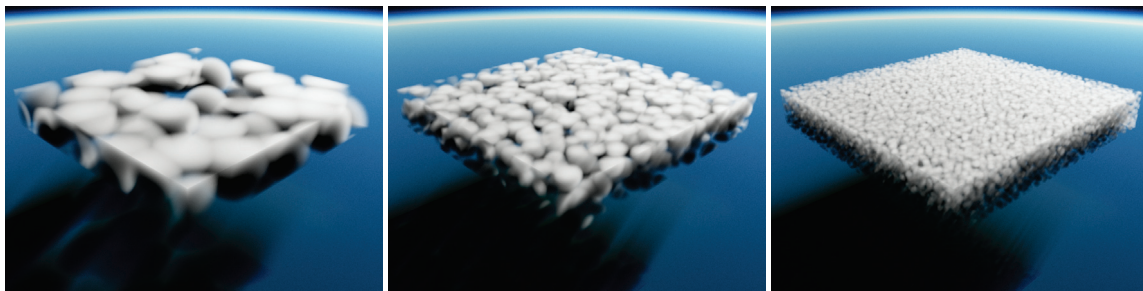




**FIGURE 2.20.** – Du bruit de turbulence (bas) composé de quatre octaves (haut) de bruit de crête de gradient sur une grille 2D avec interpolation quintique  $\mathcal{I}_q$ .



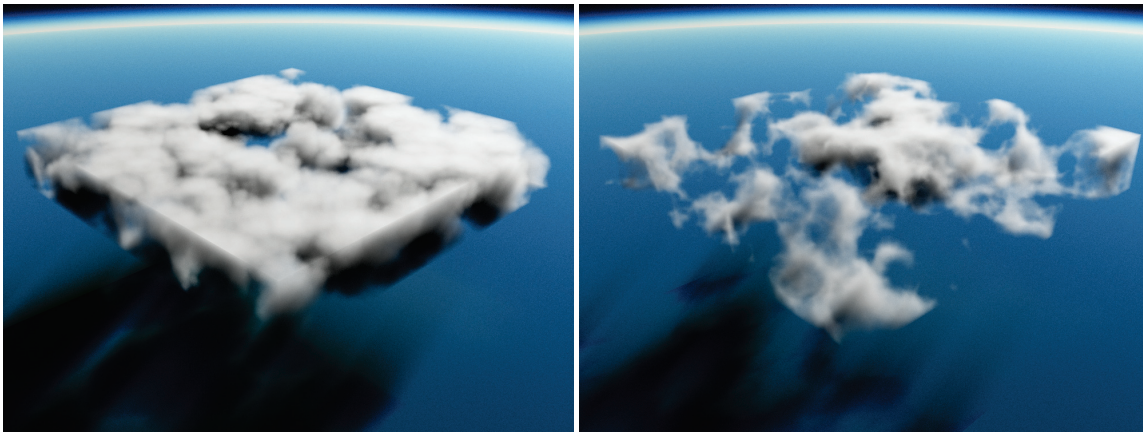
**FIGURE 2.21.** – Rendu volumique d'une octave de bruit simplexe tridimensionnel sans (gauche) et avec (droite) filtrage des valeurs supérieures à 0.6.



(a) Première octave

(b) Seconde octave

(c) Troisième octave



(d) Somme pondérée des trois octaves

(e) Complémentaire à un de la somme

**FIGURE 2.22.** – Rendu d'une couche de trois octaves de bruit cellulaire reproduisant l'aspect bourgeonnant des *Cumulus*, ou l'aspect effiloché.

## Modèle de paysage nuageux

Ce chapitre détaille le modèle fonctionnel de paysage nuageux élaboré tout au long de cette thèse. En tout point et à tout moment, la densité nuageuse atmosphérique est construite sous la forme d'une hiérarchie de fonctions à trois niveaux (figure 3.1 page suivante). Au plus haut niveau, la fonction d'atmosphère regroupe les contributions des différentes couches nuageuses (section 3.1). Au niveau de chaque couche nuageuse, la présence nuageuse à grande échelle est combinée à des détails procéduraux d'aspect nuageux. Pour finir, la fonction de présence est issue d'un mélange de primitives implicites à squelette (section 3.2 page 92), tandis que l'aspect nuageux est obtenu à partir de multiples octaves de bruits procéduraux section 3.3 page 96.

La génération procédurale des éléments animés du modèle par métamorphose est quant à elle traitée au chapitre 4 page 103.

### 3.1 Fonction d'atmosphère

Au plus haut niveau, le modèle de paysage nuageux est caractérisé par la fonction d'atmosphère  $a : \mathbb{R}^3 \times \mathbb{R} \rightarrow [0, 1]$  qui définit la densité globale de l'atmosphère en particules nuageuses en tout point de l'espace et à tout moment. La fonction d'atmosphère regroupe les contributions indépendantes des couches des différents types de nuage  $\mathcal{T}_i$  en mélangeant leurs fonctions de densité : les fonctions de couche  $C_i$ .

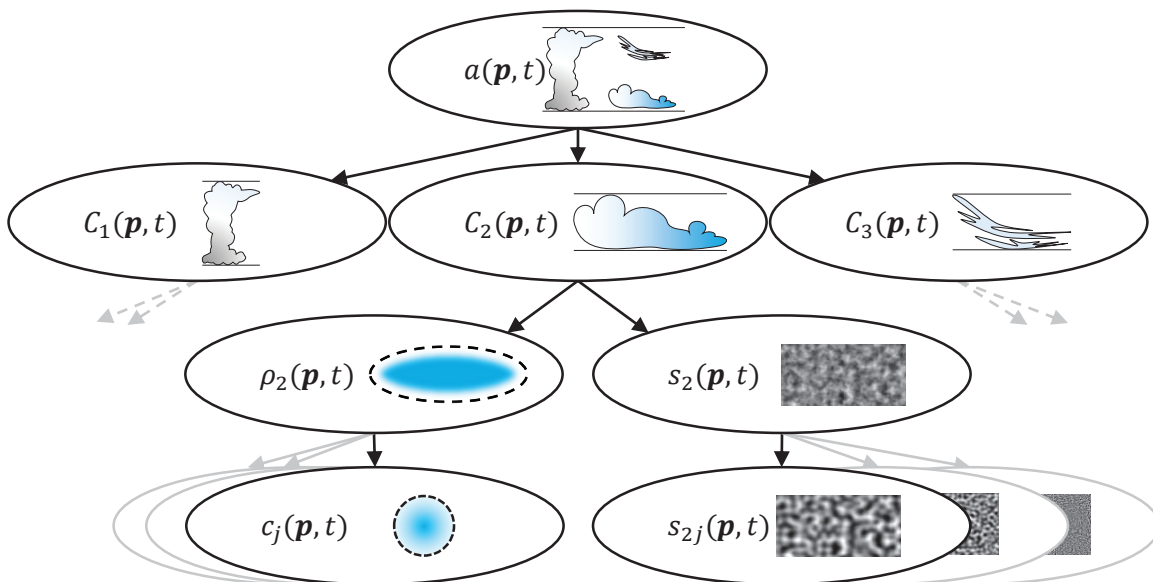
Soit  $(C_i)$  la famille des fonctions de couche de  $\mathbb{R}^3 \times \mathbb{R}$  dans  $[0, 1]$  ; la fonction d'atmosphère est exprimée comme la somme normalisée des fonctions de couche :

$$a(\mathbf{p}, t) = \sigma \left( \sum_{i \in \mathcal{T}} C_i(\mathbf{p}, t) \right) \quad (3.1)$$

où  $\sigma(x) = x / \sqrt{1 + x^2}$  est une fonction de normalisation proche de la sigmoïde. La normalisation est nécessaire car plusieurs couches peuvent se chevaucher, comme l'illustre la figure 3.3 page 89.

#### 3.1.1 Types de nuage

De nombreux types de nuage sont observables dans l'atmosphère terrestre. Ils sont nommés et classés d'après leurs formes caractéristiques ainsi que d'après les altitudes auxquelles ils ont été observés [146, 147, 148]. Ainsi, il y a trois genres principaux de nuages liés à leur aspect :



**FIGURE 3.1.** – Hiérarchie de fonctions du modèle de paysage nuageux.

- les nuages cumuliformes dont l'archétype est le *Cumulus* sont des nuages composés de bulbes convectifs qui s'élèvent sur toute la gamme des altitudes,
- les nuages stratiformes dont l'archétype est le *Stratus* sont des nuages formant une couche quasiment uniforme,
- et les nuages cirriformes dont l'archétype est le *Cirrus* sont des nuages filandreux de haute altitude et principalement composés de cristaux de glace.

La figure 3.2 reproduit des photographies des trois archétypes nuageux. Les genres principaux sont combinés pour désigner des genres intermédiaires tels que les *Stratocumulus* qui présentent à la fois les aspects des *Cumulus* et des *Stratus*, ou encore les *Cirrostratus* qui sont plutôt entre les *Cirrus* et les *Stratus*.



(a) *Cumulus* (GLG [48])

(b) *Stratus* (FAMARTIN [37])

(c) *Cirrus* (IDZKIEWICZ [64])

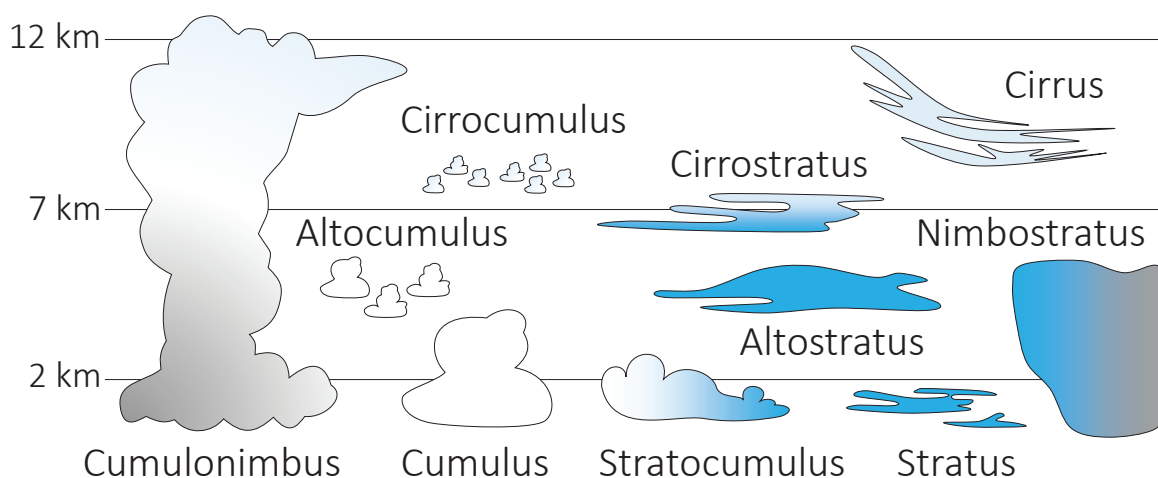
**FIGURE 3.2.** – Photographies des archétypes nuageux.

Les types de nuage prennent aussi en compte les précipitations. Anciennement un type de nuage à part entière appelé *Nimbus*, les nuages pluvieux sont aujourd'hui plutôt nommés par adjonction d'un préfixe comme pour les *Nimbostratus*, ou d'un suffixe comme pour les *Cumulonimbus*.

Enfin, le préfixe *Alto* est ajouté aux archétypes de *Cumulus* et de *Stratus* pour désigner les nuages à des altitudes moyennes, sauf dans le cas des nuages cirriformes qui n'existent

**TABLE 3.1.** – Intervalle d'altitude (km) de la base et extension verticale (km) des principaux types de nuages troposphériques à des latitudes tempérées.

	Type	Base	Extension
Haut	<i>Cirrus</i>	8.6 - 13.0	1.0 - 2.0
	<i>Cirrostratus</i>	6.0 - 8.0	0.1 - 3.0
	<i>Cirrocumulus</i>	6.0 - 8.0	0.2 - 0.4
Moyen	<i>Altostratus</i>	3.0 - 5.0	1.0
	<i>Alto cumulus</i>	2.0 - 6.0	0.2 - 0.7
Bas	<i>Stratocumulus</i>	0.5 - 1.5	0.2 - 1.2
	<i>Stratus</i>	0.1 - 0.7	0.2 - 0.8
	<i>Cumulus</i>	0.8 - 1.5	0.1 - 5.0
	<i>Nimbostratus</i>	0.1 - 1	2.0 - 5.0
	<i>Cumulonimbus</i>	0.8 - 1.0	7.0 - 12.0



**FIGURE 3.3.** – Schéma des différents types de nuages.

qu'à haute altitude. Le tableau 3.1 synthétise les altitudes auxquelles sont observables les principaux types de nuage tandis que la figure 3.3 schématise leur apparence et leur positionnement dans la troposphère.

### 3.1.2 Fonctions de couche

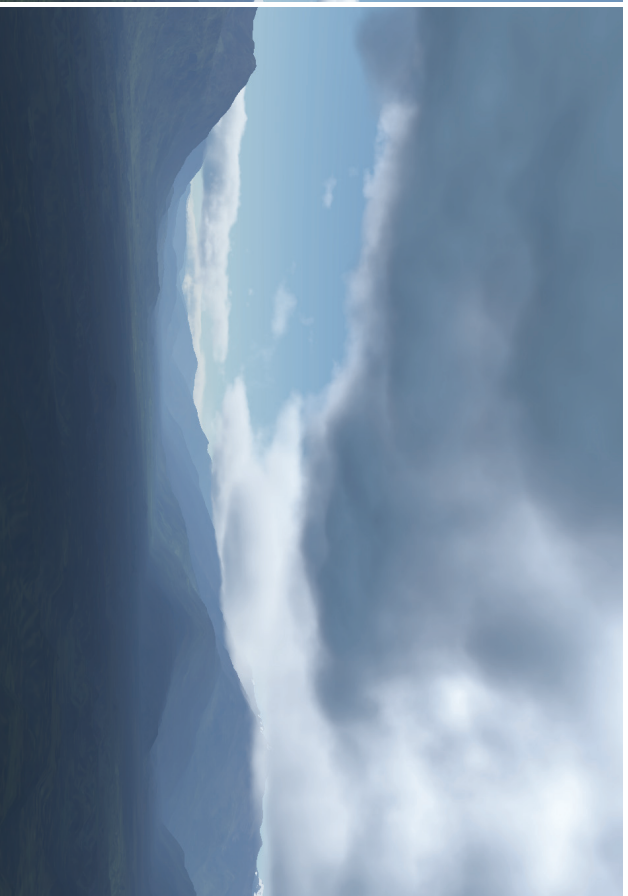
Au second niveau de la hiérarchie de fonctions du modèle de paysage nuageux, chaque fonction de couche représente chacune la densité nuageuse d'un type de nuage. Chaque fonction de couche est obtenue par combinaison d'une fonction de présence (section 3.2 page 92) et d'une fonction de texture nuageuse (section 3.3 page 96). La définition de chaque fonction de présence et de chaque fonction de texture nuageuse, ainsi que la manière de les combiner est différente pour chaque type de nuage du fait de leurs caractéristiques particulières.

**Caractéristiques nuageuses** Les nuages stratiformes comme les *Stratus* sont des couches nuageuses quasiment uniformes. Une fonction de couche nuageuse qui leur est adaptée peut simplement se résumer à une constante en tout point du support, ou avec une petite décroissance aux abords du support.

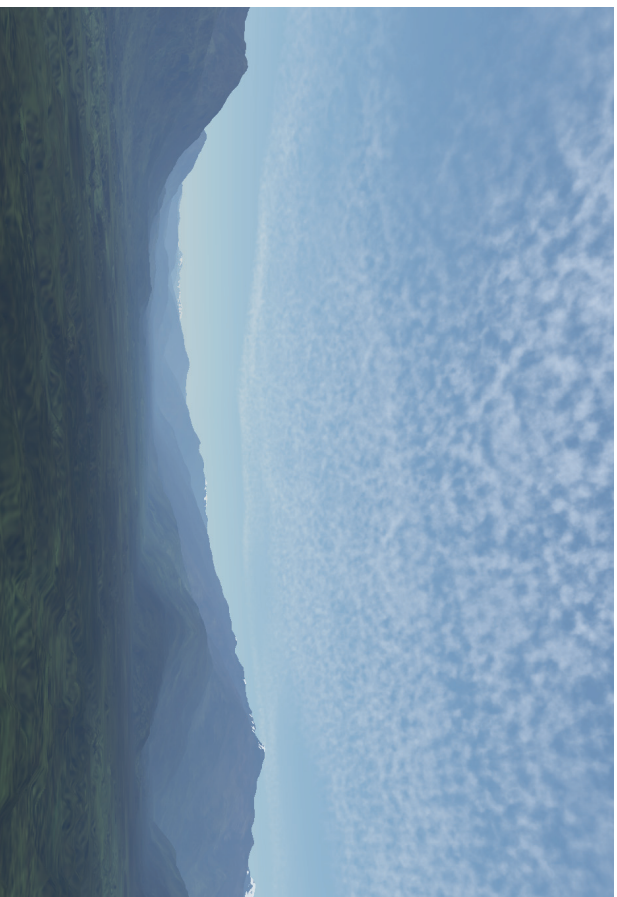




(a) *Cumulus*



(b) *Stratocumulus*



(c) *Cirrocumulus*



(d) *Nimbostratus*

**FIGURE 3.4.** – Rendus de différents types nuageux modélisés.

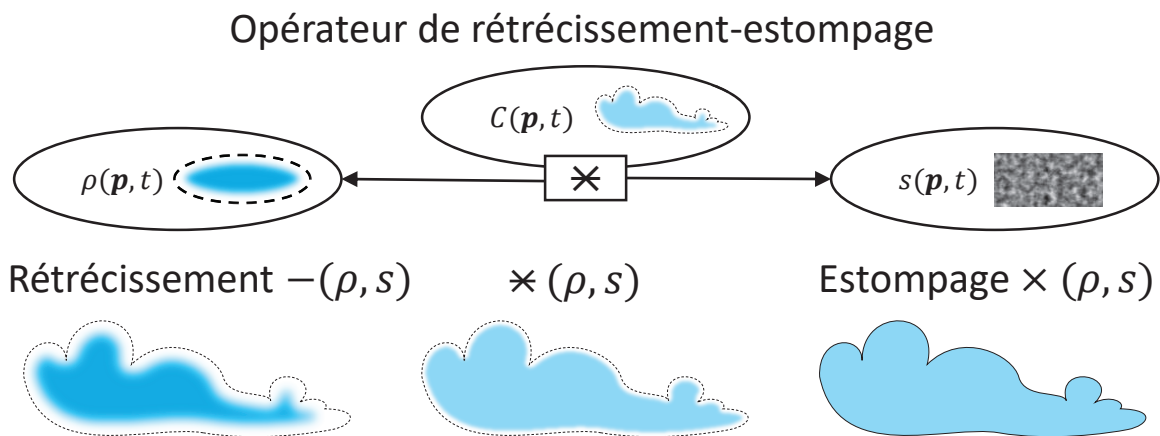
Mais il est rare que les nuages couvrent tout le ciel, et les autres types nuageux sont plus localisés et présentent des variations de densité importantes. Les nuages cumuliformes comme les *Cumulus* sont composés de bulbes très denses qui se mélangent dans leur ascension convective. Une fonction de couche nuageuse qui leur est plus adaptée est définie comme le potentiel d'un mélange de primitives implicites (section 3.2 page suivante). Les centres, densités et rayons des primitives peuvent même être animés pour reproduire l'effet de convection.

Cependant, les nuages cirriformes qui s'effilochent au vent présentent des détails très fins. C'est d'ailleurs le cas de la plupart des nuages, s'ils sont observés d'assez près. Dans le cas des *Cumulus*, ce sont ces détails qui leur donnent un aspect cotonneux. Bien qu'un mélange de primitive implicites puisse théoriquement représenter de tels détails, le nombre de primitives nécessaires devient rapidement trop grand pour passer à l'échelle d'une couverture nuageuse. En revanche, une combinaison de bruits procéduraux (section 3.3 page 96) peut être générée à la volée pour limiter l'empreinte mémoire d'un trop grand nombre de primitives, et permettre d'obtenir des détails à très haute fréquence.

**Support** Puisque les fonctions de couche sont sommées dans l'expression de la fonction d'atmosphère (équation 3.1 page 87), restreindre au moins grossièrement leurs supports permet d'accélérer l'évaluation de la densité atmosphérique globale. Deux propriétés restreignent verticalement le support de chaque fonction de couche  $\mathcal{T}_i$  :

- $e_i$ , l'intervalle d'altitudes entre lesquelles la base d'un nuage peut exister,
- et  $r_i$ , l'extension verticale maximale d'un nuage.

Les valeurs utilisées par le modèle de paysage nuageux sont synthétisées dans le tableau 3.1 page 89.



**FIGURE 3.5.** – Influence du paramètre de rétrécissement–estompage sur la combinaison entre présence texture nuageuses.

**Opérateur de rétrécissement–estompage** Combiner les fonctions de présence et de texture nuageuse revient à définir comment un type de nuage apparaît et disparaît. Or, visuellement, tous les types de nuage n'apparaissent et ne disparaissent pas de la même manière (figure 3.5). En effet, les nuages stratiformes ont tendance à apparaître et disparaître en douceur, en s'estompant graduellement et uniformément. Pour ces nuages, la fonction de présence est multipliée par la fonction de texture pour obtenir cet effet de fondu. En revanche,

le processus de grossissement et de dissipation des nuages cumuliformes est plus complexe. Par conséquent, la fonction de couche des nuages stratiformes est plutôt exprimée par une somme entre présence et texture nuageuses. Dans ce cas, une réduction de la présence entraîne un rétrécissement par érosion du support nuageux.

Plutôt que de se limiter soit à un produit soit à une somme entre les fonctions de présence et de texture nuageuses, la fonction de couche est exprimée à l'aide d'une interpolation linéaire entre les deux formes.

Soit  $s_i$  la fonction de texture nuageuse,  $\rho_i$  la fonction de présence et  $\alpha_i \in [0, 1]$  le coefficient d'interpolation ; la fonction de couche est alors définie par

$$C_i = *_{i}(\rho_i, s_i) \quad (3.2)$$

$$= (1 - \alpha_i) \cdot (\rho_i + s_i - 1) + \alpha_i \cdot \rho_i s_i. \quad (3.3)$$

Le comportement de chaque type nuageux différent  $\mathcal{T}_i$  est ainsi contrôlé par un coefficient unique de rétrécissement–estompagement, noté  $\alpha_i \in [0, 1]$ , qui module l'effet de disparition entre fondu progressif et érosion. Le rétrécissement est adapté aux nuages cumuliformes tandis que l'estompagement est plus approprié au brouillard ou aux nuages stratiformes de haute altitude. Choisir un coefficient  $\alpha_i = 0$  produit un effet de rétrécissement tandis que  $\alpha_i = 1$  génère un effet d'estompagement ; des valeurs intermédiaires permettent de moduler les effets (figure 3.6 page ci-contre).

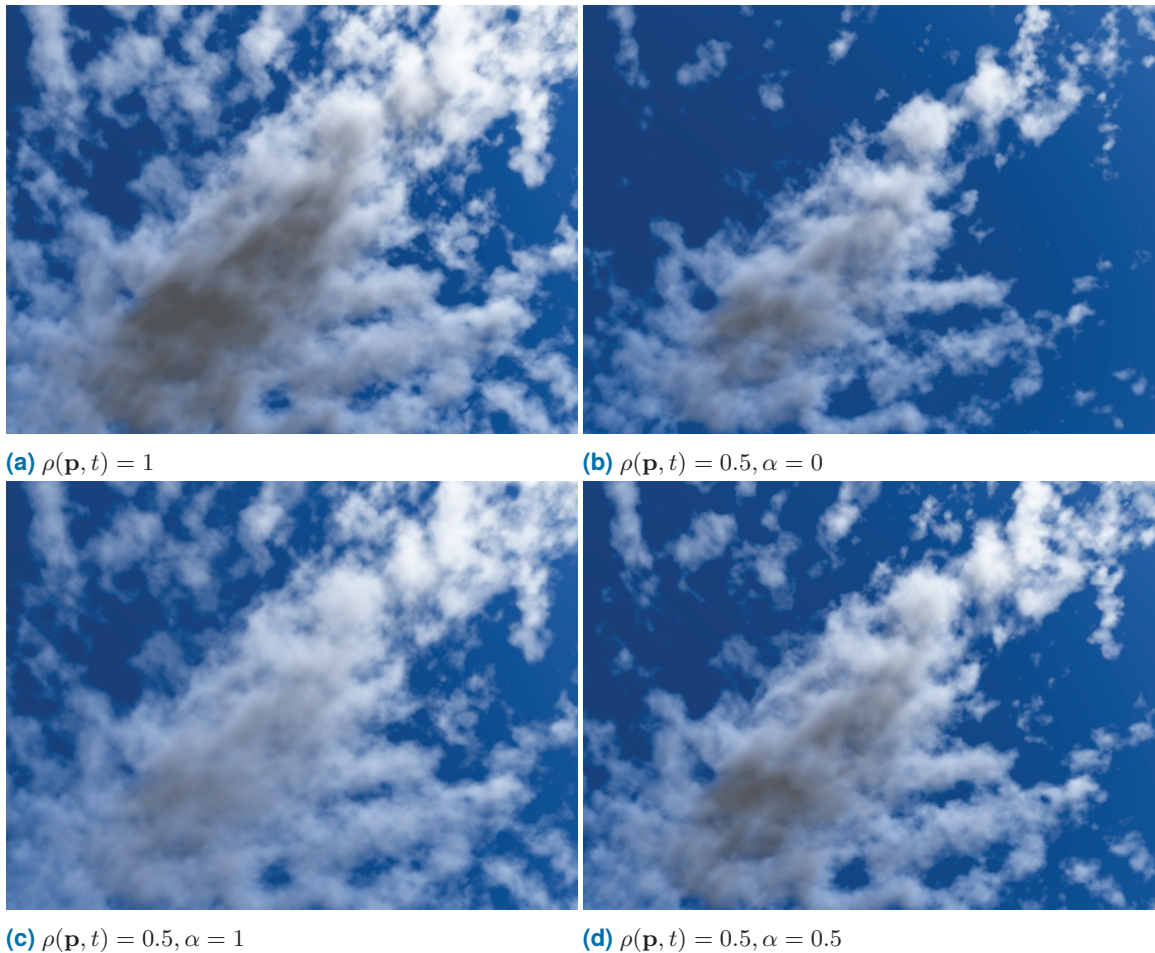
Le coefficient de rétrécissement–estompagement  $\alpha_i$  est une constante définie pour chaque type nuageux, et n'est pas interpolé entre les images clefs. Dans l'implémentation réalisée au cours de cette thèse, la valeur  $\alpha_i = 0.1$  est utilisée pour les *Cumulus* tandis que la valeur  $\alpha_i = 1$  est utilisée pour les *Cirrostratus*.

Il est intéressant de noter que la valeur du coefficient de réduction–estompagement influence la complexité d'évaluation des champs nuageux (figure 3.6 page ci-contre). En effet, dans le cas d'un effet d'estompagement, *id est*  $\alpha_i = 1$ , le support n'est pas modifié. En revanche, régler  $\alpha_i = 0$  entraîne une réduction du support au fur et à mesure que  $\rho_i$  diminue.

## 3.2 Fonctions de présence

Au troisième niveau de la hiérarchie de fonctions du modèle de paysage nuageux, une fonction de présence  $\rho_i(\mathbf{p}, t)$  est définie et décrit la répartition de la couverture nuageuse à grande échelle pour chaque type de nuage  $\mathcal{T}_i$ . Sans perte de généralité, un seul type de nuage est considéré à la fois, permettant de se passer de l'indigage pour la suite de cette section.

La fonction de présence peut être construite de nombreuses façons : par description manuelle, de manière procédurale, ou par une simulation physique par exemple. La figure 3.7 page 94 montre des exemples de fonctions de présence prescrites manuellement par l'utilisateur afin d'obtenir des effets spéciaux tels qu'une couverture nuageuse en forme de cœur, des



**FIGURE 3.6.** – Influence du paramètre de rétrécissement–estompage  $\alpha$  (équation 3.3 page précédente). La couverture nuageuse de référence (a) peut être modifiée afin d’obtenir un effet d’estompage (c), un effet de rétrécissement (b), ou bien un effet intermédiaire (d).

traînées de condensation, ou bien une couverture de nuages cumuliformes percée à des endroits choisis pour offrir une vue sur de hauts pics montagneux.

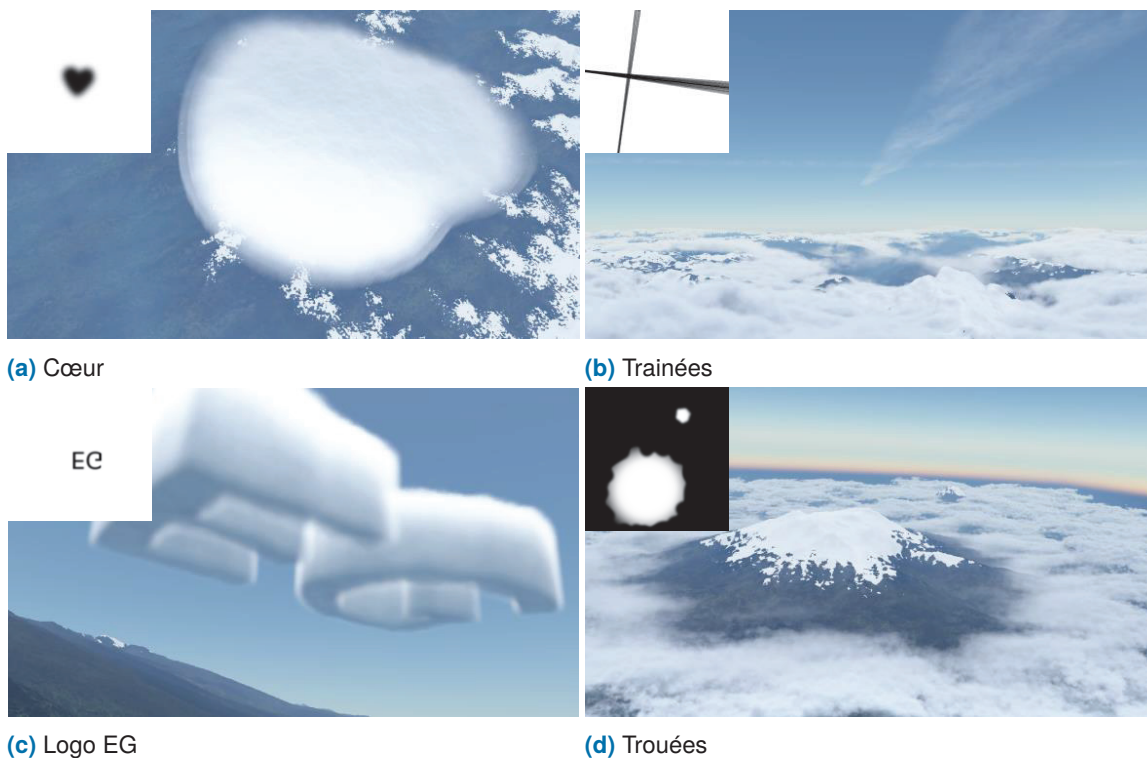
Afin de pouvoir être animée par métamorphose (chapitre 4 page 103), la fonction de présence est définie comme un mélange de primitive implicites à squelette. Pour le lecteur non familier de la modélisation implicite, les fondements sont rappelés dans la section 2.1 page 65.

Contrairement à la modélisation de surface implicite, l’objectif n’est pas d’obtenir une isosurface mais un volume qui représente la densité nuageuse. Aussi, le potentiel scalaire défini par un mélange de primitives implicites à squelette est utilisé directement en tant que champ de densité.

Soit  $(c_j)$  la famille de primitives implicites, la fonction de présence est définie comme le champ scalaire de potentiel normalisé du mélange des primitive (équation 2.6 page 67) :

$$\rho(\mathbf{p}, t) = \sigma \left( \sum_j c_j(\mathbf{p}, t) \right). \quad (3.4)$$





**FIGURE 3.7.** – Rendus de paysages nuageux selon différentes fonctions de présence. Les fonctions de présences sont visualisées en niveau de gris : le noir correspond à la présence maximale et *vice versa*. Un nuage en forme de cœur est sculpté dans une couche de *Stratocumulus* (a); des *Cirrus* forment des traînées de condensation produites par un avion volant à haute altitude (b); lettres sculptées en *Stratus* (c); des percées sont imposées à une couche de *Cumulus* (d).

où, de même que pour le mélange des différentes couches nuageuses (équation 3.1 page 87), la somme est normalisée en utilisant la fonction  $\sigma$  similaire à la sigmoïde.

### 3.2.1 Primitives de présence

Le squelette des primitives implicites utilisées est ponctuel et la fonction de distance utilisée est plus ou moins déformée selon le type de nuage concerné. Utiliser une fonction de distance peu ou pas modifiée entraîne des zones de densité sphériques utiles pour modéliser les bulbes convectifs de *Cumulonimbus* par exemple. Dans le cas de nuages plus aplatis comme des *Stratus*, augmenter la fonction de distance grâce à un facteur vertical permet de décrire des primitives plus ellipsoïdales, un peu comme des capsules ou des lentilles, limitant ainsi le nombre de primitives nécessaires pour couvrir le support d'une grande zone nuageuse à faible extension verticale.

Chaque primitive implicite à squelette ponctuel  $\mathcal{C}_j$  est caractérisée par ses paramètres (section 2.1 page 65) :

- son centre  $\mathbf{C}_j$ ,
- son rayon d'influence horizontal  $R_{Hj}$ ,
- son rayon d'influence vertical  $R_{Vj}$ ,



— et sa densité maximale  $D_j$ .

Le champ scalaire à support compact de la primitive est défini en combinant une fonction d'atténuation  $g$  décroissante et lisse, et une fonction de distance au centre :

$$c_j(\mathbf{p}, t) = D_j \cdot g \circ d_j(\mathbf{p}, t) \quad (3.5)$$

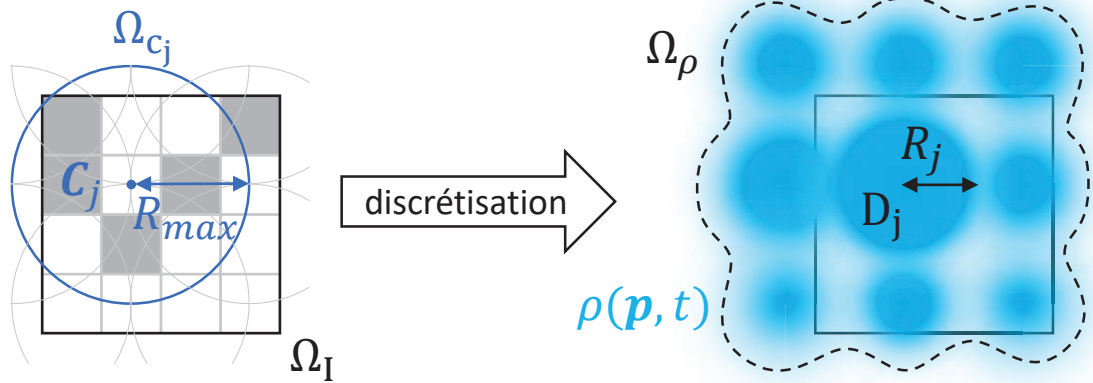
où  $d_j$  est la fonction de distance euclidienne au centre de l'ellipsoïde, normalisée par le rayon d'influence de la primitive :

$$d_j(\mathbf{p}, t) = \left\| (\mathbf{C}_j(t) - \mathbf{p}) \cdot \left( \frac{\mathbf{e}_0}{R_{H_j}(t)} + \frac{\mathbf{e}_1}{R_{H_j}(t)} + \frac{\mathbf{e}_2}{R_{V_j}(t)} \right) \right\| \quad (3.6)$$

et où la fonction d'atténuation cubique (équation 2.14 page 72) est utilisée :

$$g_j(r) = \begin{cases} (1 - r^2)^3 & \text{si } 0 \leq r \leq 1 \\ 0 & \text{sinon} \end{cases} \quad (3.7)$$

### 3.2.2 Génération des primitives de présence



**FIGURE 3.8.** – Schéma de discrétisation de la carte de présence fournie par l'utilisateur en un ensemble de primitives de présence.

Spécifier manuellement la fonction de présence d'un type nuageux est un processus fastidieux et répétitif. Les dimensions du paysage nuageux à définir et le nombre de types nuageux différents qui le composent augmentent la complexité de la modélisation. Et c'est encore pire dans le cas de nuages animés en présence d'un relief montagneux avec lequel ils pourraient interagir car une animation cohérente requiert notamment d'éviter les intersections des nuages avec le terrain et avoir des trajectoires cohérentes qui sont influencées par le relief. Aussi, une méthode procédurale et contrôlable qui permet de modéliser et d'animer facilement les fonctions de présence des nuages est proposée.

Afin que le processus de création de l'utilisateur reste simple et efficace, les primitives sont générées automatiquement d'après des images rastérisées fournies par l'utilisateur (figure 3.8). Le niveau de gris de chaque image représente la densité de la couverture nuageuse pour un type de nuage à un instant donné. Elle n'a pas besoin d'être détaillée car c'est une description grossière qui est convertie en un ensemble de primitives. Pour ce

faire, le domaine de la carte de présence est couvert par un support de primitives et chaque primitive se voit associée un facteur de densité maximale proportionnel à la densité des pixels dans sa sphère d'influence. Une manière de couvrir ainsi le domaine de la carte de présence est de placer des primitives sur une grille de manière à ce que leurs sphères d'influence se chevauchent. Le champ de potentiel obtenu présente ainsi les mêmes composantes connexes que la carte, à la précision d'échantillonnage près. Les paramètres d'altitude et de rayon d'influence vertical des primitives sont déterminés de manière à concorder avec le type de nuage de la couche (section 3.1.2 page 89).

### 3.2.3 Animation et métamorphose

Il est important de noter que les paramètres des primitives sont exprimés en fonction du temps car les primitives sont animées. Pour que l'animation des primitives soit elle-aussi simple, intuitive et contrôlable, une méthode d'animation automatique par métamorphose entre images clefs est proposée au chapitre 4 page 103.

## 3.3 Fonctions de texture nuageuse

Au troisième niveau de la hiérarchie de fonctions du modèle de paysage nuageux, une fonction de texture nuageuse  $s_i(\mathbf{p}, t)$  ajoute des détails à la fonction de présence  $\rho_i$  pour modéliser l'aspect nuageux spécifique de chaque type de nuage  $\mathcal{T}_i$ .

Les fonctions de texture nuageuse sont elles-même décomposées en combinaisons de bruits procéduraux de différents types et fréquences afin de reproduire les motifs caractéristiques de chaque type de nuage à modéliser. Pour le lecteur non familier des bruits procéduraux, les fondements sont rappelés dans la section 2.2 page 75.

Le domaine des fonctions de texture nuageuse n'est pas seulement spatial mais aussi temporel. Chaque fonction de texture nuageuse possède une animation propre et une animation qui dépend des conditions de vents de son type de nuage.

### 3.3.1 Motifs nuageux

Certaines combinaisons de bruits procéduraux permettent de reproduire des motifs, des aspects intéressants car ils se répètent dans différents types de nuage.

#### **Motif stratiforme**

Les nuages stratiformes ont un aspect proche d'une couche nuageuse homogène. Comme les nuages stratiformes sont peu détaillés, quelques octaves de bruit simplexe font très bien l'affaire. Il n'est pas nécessaire de filtrer les valeurs, à moins de vouloir créer des percées dans la couche nuageuse (figure 3.9 page 98).

## Motif cumuliforme

Les nuages cumuliformes présentent des bulbes convectifs de tailles variées qui s'élèvent à partir d'une base commune. Les différentes échelles de bulbes s'amalgament en une forme boursouflée. Le motif cumuliforme s'obtient en suivant le schéma du bruit fractal avec des octaves de bruit cellulaire tridimensionnel et en filtrant une part importante des valeurs les plus basses pour extraire des ilots convectifs (figure 3.10 page 99).

## Motif cirriforme

Les nuages cirriformes sont les plus compliqués à reproduire car ils présentent des filaments qui s'étirent et se déforment au gré des vents. De plus, les nuages cirriformes sont si fins que leur rendu volumique est difficile, que ce soit par *raymarching* ou *path tracing* (annexe A page 135). Nos meilleurs résultats de *Cirrus* ont été obtenus en appliquant une déformation au domaine du bruit afin d'allonger les ilots nuageux et de reproduire les ondulations caractéristiques des filaments (figure 3.11 page 100).

### 3.3.2 Animation de texture nuageuse et dynamique de la turbulence

Les nuages sont évidemment mus par les vents à grande échelle, mais à plus petite échelle, ils présentent aussi une animation propre turbulente et chaotique (figure 3.12 page 101).

#### Animation par le vent

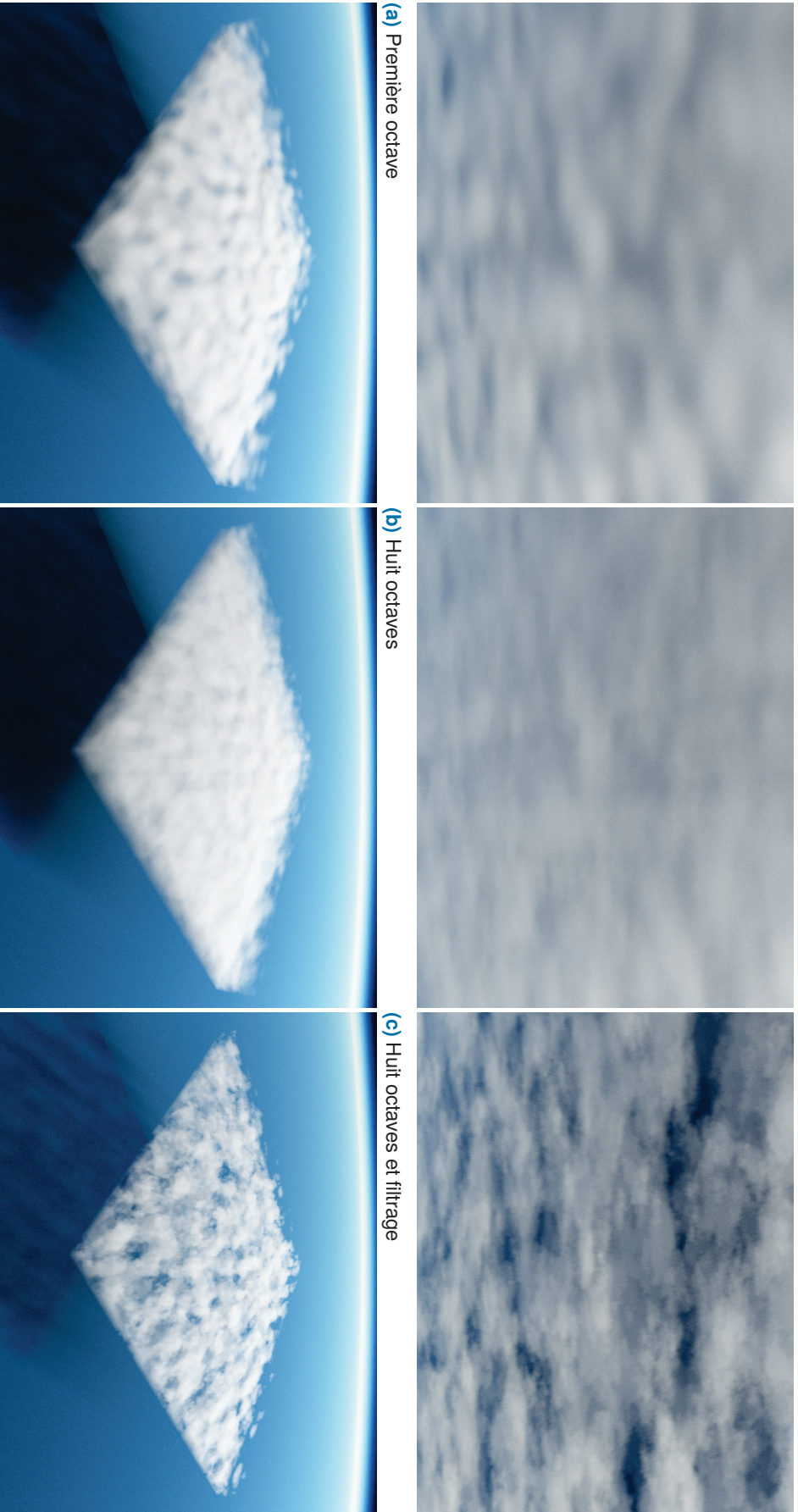
L'influence du vent sur chaque texture nuageuse est obtenue par translation de son domaine selon son vecteur  $\mathbf{v}_i$  représentant la direction principale et l'intensité moyenne du vent du type de nuage  $\mathcal{T}_i$

$$s_i(\mathbf{p}, t) = n_i(\mathbf{p} - t \cdot \mathbf{v}_i, t). \quad (3.8)$$

#### Animation propre turbulente

L'animation propre turbulente de chaque fonction de texture nuageuse correspond à l'animation intrinsèque des bruits quadridimensionnels qui la composent, *id est* l'agitation qui est indépendante des conditions de vent.

Comme le fait remarquer SAKAS *et al.* [128] d'après l'étude des turbulences gazeuses, l'agitation des parcelles gazeuses dépend de l'échelle. L'agitation est maximale pour une taille de parcelle gazeuse donnée mais décroît en deçà et au-delà. Le schéma classique de combinaison des octaves de bruit fractal (section 2.2.2 page 83) reproduit bien la cascade de décroissance énergétique observée dans la turbulence gazeuse avec l'augmentation de la



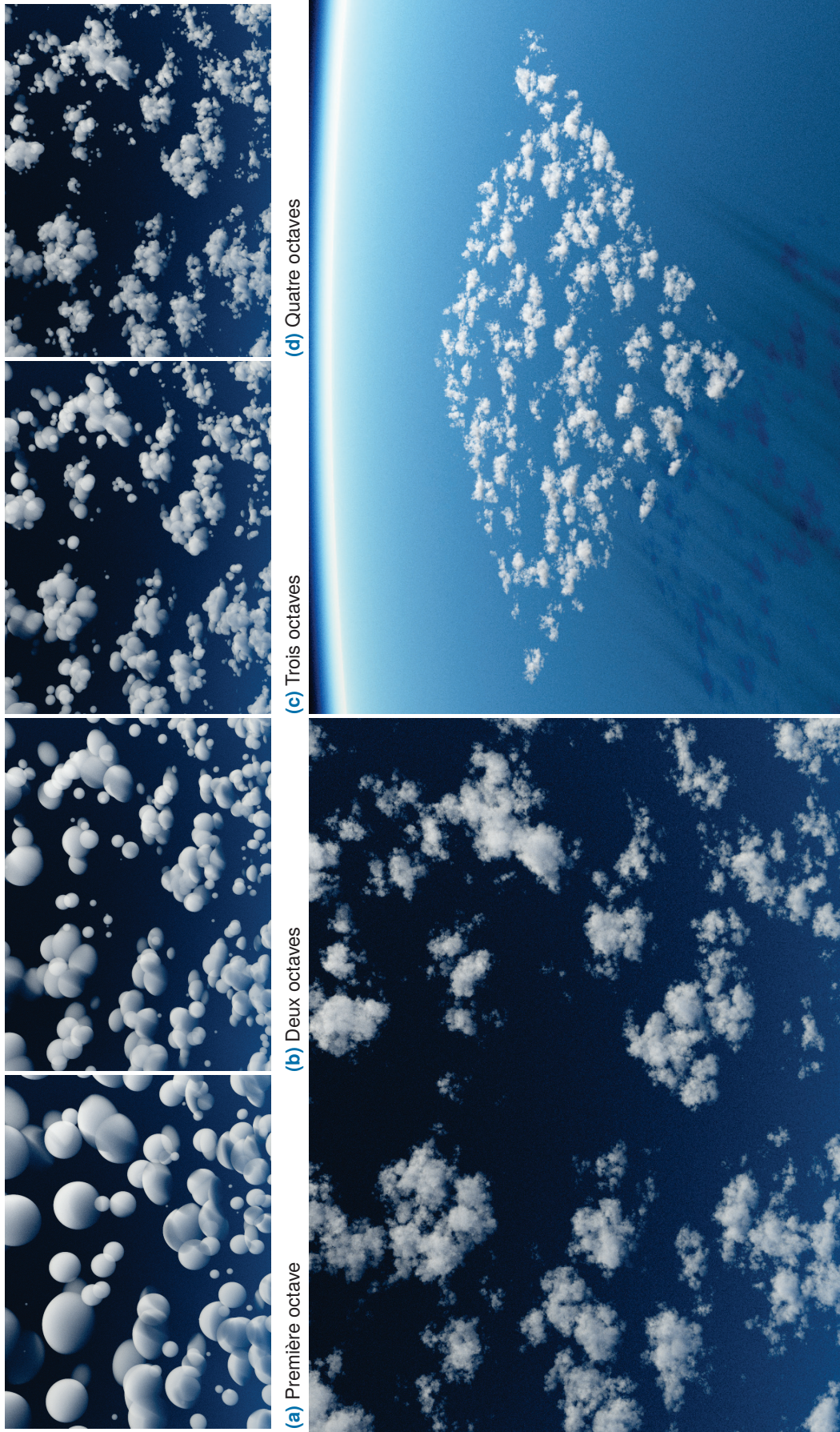
(d) Première octave

(e) Huit octaves

(f) Huit octaves et filtrage

**FIGURE 3.9.** – Rendus volumiques d'une couche de motif stratiforme obtenu par combinaison fractale d'octaves de bruit simplexe tridimensionnel. La colonne de droite montre comment obtenir des percées dans la couche nuageuse en seillant les valeurs les plus faibles. Les dimensions de la maille pour la première octave sont d'environ  $4.25 \times 8.5 \times 25.5 \text{ km}^3$  pour environ  $6.5 \times 13 \times 4$  oscillations le long des côtés du domaine nuageux tandis que le bruit fractal est paramétré par une lacunarité de 2,  $02$  et une persistance de  $0,52$ , et les valeurs entre  $0,2$  et  $0,5$  ont été normalisées avant d'être interprétées comme densité. Les percées de la dernière colonne sont obtenues en ramenant à zéro les valeurs inférieures à  $0,7$ .





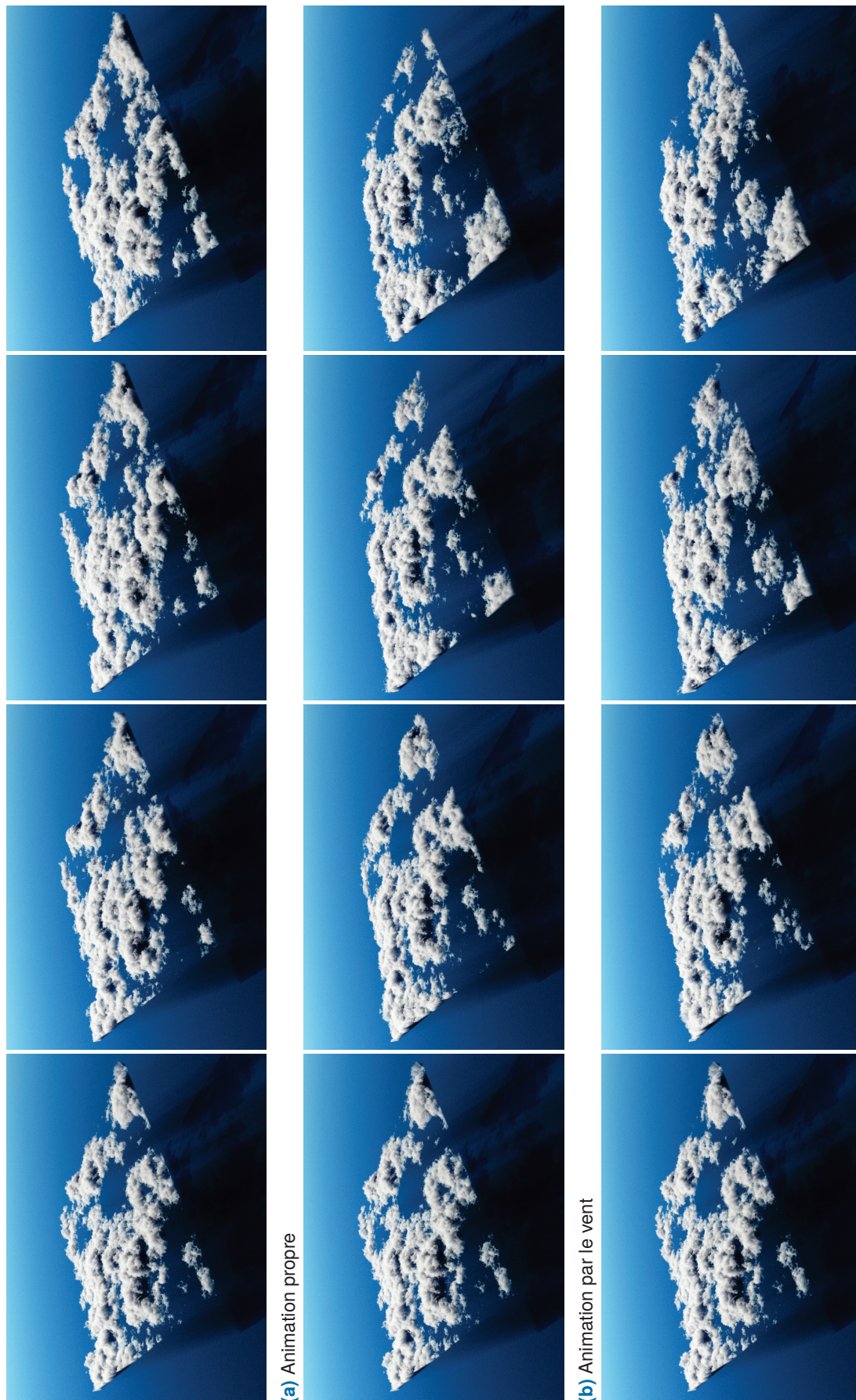
**FIGURE 3.10.** – Rendus volumiques de motif cirriforme obtenu par combinaison fractale d'octaves de bruit cellulaire tridimensionnel. La ligne du haut montre le raffinement du motif cirriforme avec l'ajout des quatre premières octaves de, tandis que les deux rendus du bas montrent le résultat après l'ajout de huit octaves. La taille de la maille pour la première octave est d'environ 8.5 km pour environ 13 oscillations sur la largeur du domaine nuageux tandis que le bruit fractal est paramétré par une lacunarité de 2.02 et une persistance de 0.72, et les valeurs inférieures à 0.5 ont été ramenées à zéro avant d'être interprétées comme densité.





**FIGURE 3.11.** – Rendus volumiques de motif cirriforme obtenu par combinaison fractale de bruit complexe tridimensionnel et déformation du domaine. Selon l'axe des  $x$ , le domaine est étiré par un facteur 10 tandis que sur l'axe des  $y$ , une octave de bruit simplexe paramètre la translation du domaine avec une amplitude de 5 km. Avant déformation, la taille de la maille pour la première octave est d'environ 8,5 km pour environ 13 oscillations sur la largeur du domaine nuageux tandis que le bruit fractal est paramétré par une lacunarité de 2.02 et une persistance de 0.72, et les valeurs appartenant à l'intervalle [0.55, 0.61] sont ramenées à la valeur maximale avant d'être interprétées comme densité.





(a) Animation propre

(b) Animation par le vent

(c) Animation combinée

**FIGURE 3.12.** – Rendu d'une séquence d'animation de texture nuageuse cumuloforme.

fréquence. Concernant les basses fréquences, cette partie du spectre relève de l'animation à grande échelle, c'est-à-dire de la fonction de présence et nous n'appliquons pas de traitement particulier pour adhérer au comportement physique de la turbulence gazeuse. Une étude récente [87] suggère d'ailleurs une autre méthode pour produire une animation plus physique.

## 3.4 Conclusion

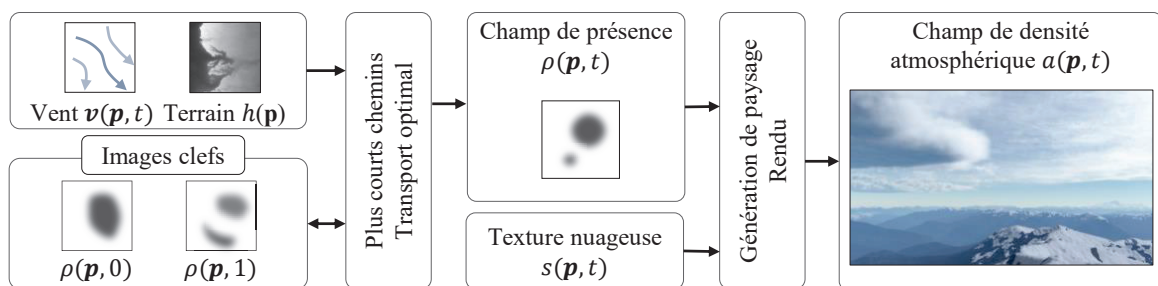
L'ensemble du modèle d'atmosphère est animé, mais chaque couche nuageuse est animée indépendamment. L'évolution globale de la couverture nuageuse ainsi que le déplacement visuel des zones nuageuses est obtenu à travers l'animation du champ de présence  $\rho_i(\mathbf{p}, t)$  et des paramètres des primitives implicites à squelettes qui le composent. Le chapitre 4 page ci-contre détaille comment métamorphoser des primitives nuageuses de manière cohérente et contrôlable, tout en prenant en compte des conditions de vents et le relief du paysage nuageux.

De manière totalement indépendante, la fonction  $s_i(\mathbf{p}, t)$  définit l'animation propre de la texture nuageuse de chaque type de nuage par l'utilisation de fonctions de bruit quadrimensionnelles, *id est* spatiotemporelles, de type simplexe et cellulaire (section 2.2 page 75).

Bien que cette séparation entre l'animation des primitives et celle des textures nuageuses procure une grande liberté créative, elle peut avoir des effets étranges lorsqu'elle est mal menée. Pour prendre un exemple simple, translater les détails de l'aspect nuageux d'après une direction de vent ne prend pas nécessairement en compte la trajectoire que peut suivre une primitive du champ de présence. Dans ce cas, la primitive se détache clairement dans la dynamique de l'animation et apparaît comme une fenêtre sur le bruit animé qui glisse de manière incohérente avec les détails.

## Métamorphose de paysages nuageux

Dans ce chapitre, nous cherchons à générer les primitives animées qui contrôlent à grande échelle la répartition spatiotemporelle des nuages dans le modèle de paysage nuageux (chapitre 3 page 87). Plutôt que de positionner et animer manuellement des primitives, nous proposons une méthode procédurale par métamorphose qui génère automatiquement les primitives interpolantes les plus cohérentes d'après des images clefs, un terrain et des vents (figure 4.1).



**FIGURE 4.1.** – Aperçu : étant donné un ensemble d'images clef de nuages en entrée, nous générons une fonction de contrôle animée générique  $c_i(\mathbf{p}, t)$  qui définit la couverture nuageuse animée. Cette fonction est combinée avec le modèle de couche nuageuse pour calculer la fonction de densité  $a_i(\mathbf{p}, t)$  qui est finalement rendue.

Pour chaque type de nuage  $\mathcal{T}_i$ , une suite d'images clefs, chacune constituée d'un ensemble de primitives statiques, est spécifiée à des instants clefs donnés. Sur chaque intervalle de temps formé par un couple d'images clefs subséquentes, la méthode procède en quatre grandes étapes (figure 4.2 page suivante) :

1. le calcul du plus court chemin entre chaque couple de primitives statiques (section 4.1 page 105),
2. la mise en correspondance optimale des primitives (section 4.2 page 114),
3. l'ajout de primitives fantômes (section 4.3 page 119),
4. et l'instanciation des primitives interpolantes (section 4.4 page 122).

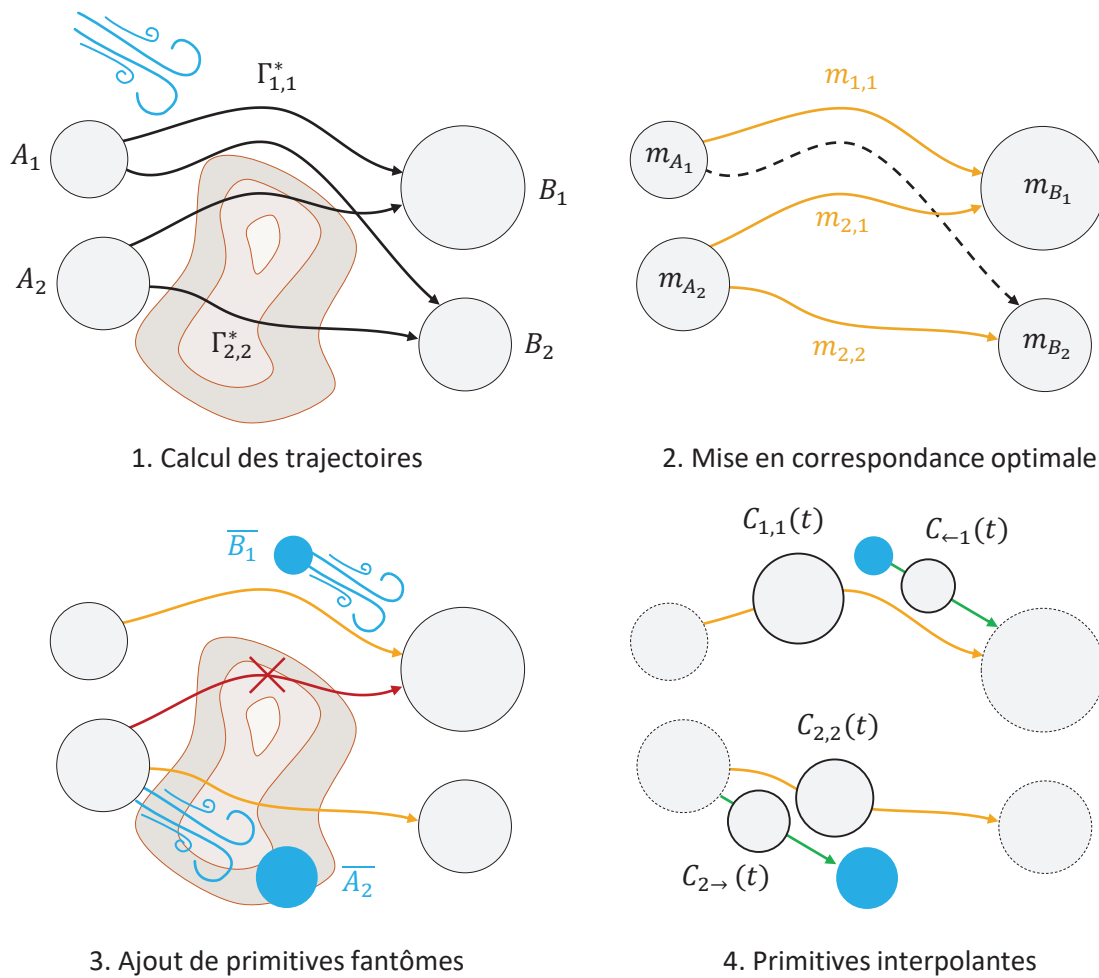


FIGURE 4.2. – Aperçu du processus de métamorphose.

Cette méthode a fait l'objet d'une publication à une conférence internationale [142].

**Conventions et notations** Par souci de lisibilité et sans perte de généralité, nous considérons dans la suite de ce chapitre une seule métamorphose à la fois : la métamorphose d'une couche nuageuse d'un type de nuage  $\mathcal{T}$  donné, entre une paire donnée d'images clés subséquentes  $(A, B)$  définies aux instants clés  $t_A$  et  $t_B$ .

La première image clé et ses  $n_A$  primitives statiques sont qualifiées d'initiales et notées  $A = \{A_i \mid i \in \llbracket 1, n_A \rrbracket\}$ .

La seconde image clé et ses  $n_B$  primitives statiques sont qualifiées de finales et notées  $B = \{B_j \mid j \in \llbracket 1, n_B \rrbracket\}$ .

Une primitive animée interpolant un couple de primitives statiques  $(A_i, B_j)$  est qualifiée d'interpolante et est notée  $C_{i,j}(t)$  avec  $t_A \leq t \leq t_B$ .

Une table répertoriant l'ensemble des symboles utilisés au travers de ce manuscrit est fournie 147.



## 4.1 Calcul des trajectoires

Afin d'être cohérente avec l'environnement des nuages, la trajectoire de chaque primitive interpolante doit satisfaire un certain nombre de contraintes comme de ne pas traverser le terrain par exemple. Pour juger de la qualité d'une trajectoire par rapport aux contraintes, une fonction de coût est définie en tout point du domaine spatio-temporel qui sépare les primitives initiales et finales. Bien qu'il soit ainsi possible de choisir les meilleures trajectoires *a posteriori* parmi un ensemble de trajectoires pré-construites, nous proposons de calculer l'ensemble des plus courts chemins directement d'après la fonction de coût.

### 4.1.1 Environnement

L'environnement nuageux comprend deux éléments dont l'influence est spécifique à chaque couche nuageuse : des conditions de vents et un terrain.

**Vents** Les conditions de vents qui affectent un type nuageux sont représentées sous la forme d'un champ vectoriel de vent. Le vent est considéré comme variable selon le temps entre les instants clefs et la position géographique. En revanche, l'altitude est ignorée dans l'expression du champ vectoriel de vent au sein d'une couche nuageuse. Différentes couches nuageuses peuvent néanmoins présenter différents champs vectoriels de vent, même si les couches se chevauchent. Soit  $v_{\max}$  la vitesse maximale des vents considérés, le champ vectoriel de vent est noté

$$\mathbf{v}(x, y, t) = \mathbf{v} \in \mathbb{R}^3 \mid \|\mathbf{v}\| \leq v_{\max}. \quad (4.1)$$

**Terrain** Le terrain au-dessus duquel les nuages se déplacent est quant à lui représenté par une carte de hauteur, qui ne varie pas dans le temps, et donne l'altitude de la surface du terrain par rapport au plan  $z = 0$ . Soit  $\Omega_h \subseteq \mathbb{R}^2$  le domaine spatial du terrain, et  $h_{\min}$  et  $h_{\max}$  les altitudes minimales et maximales du terrain ; la hauteur de la surface est notée

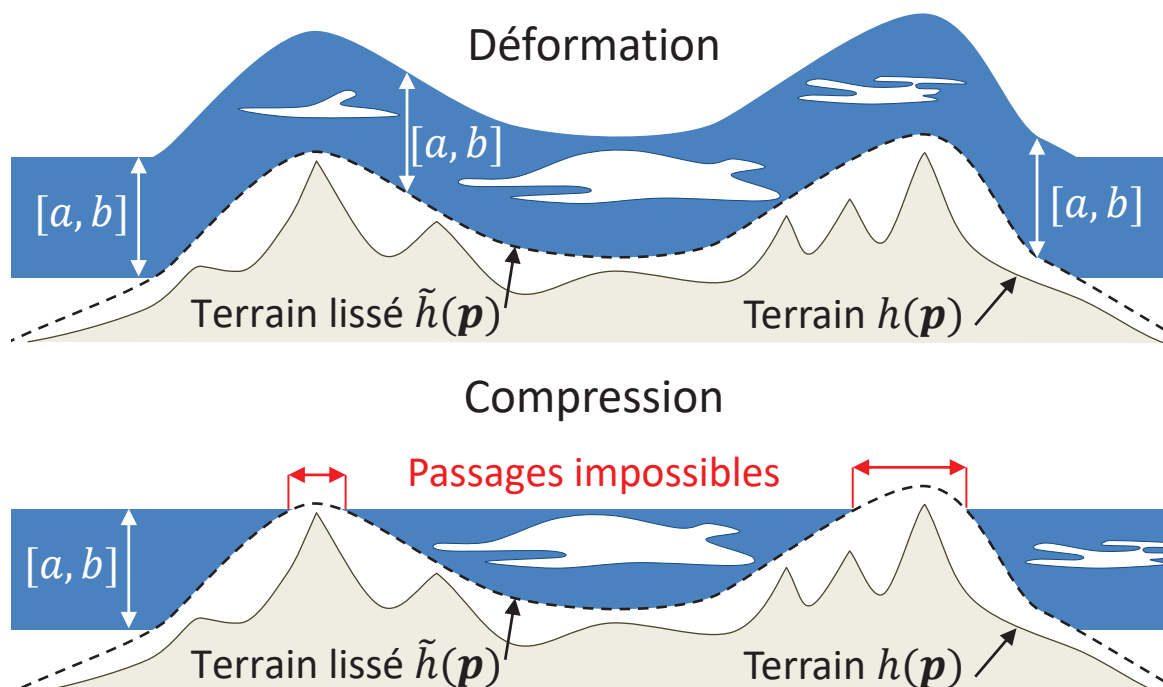
$$h : \Omega_h \rightarrow [h_{\min}, h_{\max}]. \quad (4.2)$$

La courbure de la planète est négligée à l'échelle des paysages nuageux modélisés, sauf pour la densité atmosphérique en air qui est décrite en fonction de l'altitude haut-dessus du globe terrestre. Les plus grands paysages nuageux modélisés durant les travaux de cette thèse couvrent un terrain dont le domaine spatial  $\Omega_h$  couvre de l'ordre de  $100 \times 100 \text{ km}^2$  (section 4.5.1 page 127).

### 4.1.2 Contraintes des trajectoires

L'intervalle d'altitudes du support des couches nuageuses ainsi que le terrain et les conditions de vent régissent la construction de trajectoires valides pour les primitives interpolantes. Ainsi, les trajectoires qui passent à travers le terrain ou en dehors de la couche nuageuse sont des exemples de trajectoires qui sont absurdes et à éviter.

**Contraintes imposées par le terrain** Une des contraintes imposées aux trajectoires est de ne pas causer d'intersection entre les primitives et le terrain. L'intervalle d'altitudes de la base qui est donné pour chaque type de nuage et la fonction de hauteur du terrain sont utilisés pour guider la composante verticale des trajectoires, voire même en réduire le domaine horizontal lorsque le terrain traverse la couche nuageuse.



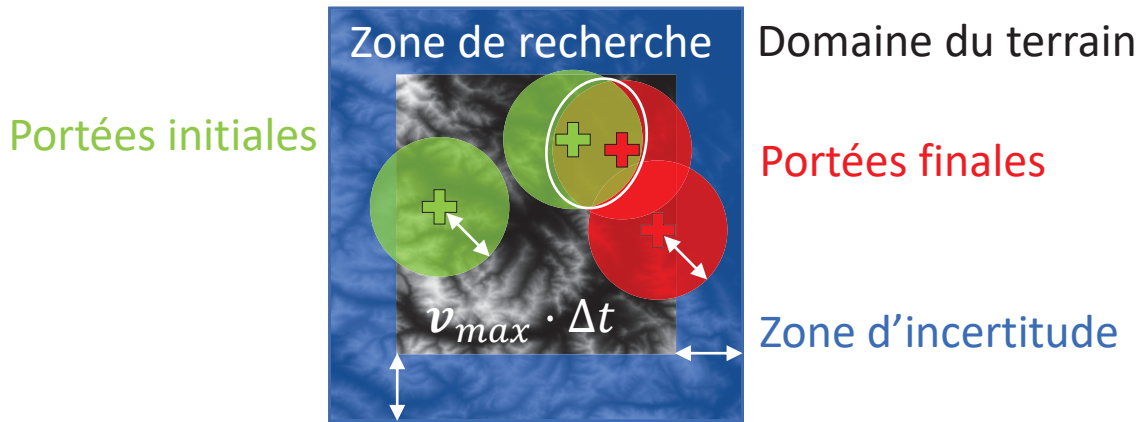
**FIGURE 4.3.** – Schématisation des deux manières d'imposer les contraintes de hauteur du terrain sur la couche nuageuse : la déformation uniforme dans le cas d'altitudes relatives (haut), et la compression de l'intervalle d'altitudes de la base nuageuse par le terrain (bas).

Il y a deux manières d'interpréter l'altitude de la base nuageuse : soit comme la hauteur relative à la surface du terrain, soit comme la hauteur par rapport au niveau de la mer (figure 4.3). Dans le premier cas, l'ensemble du domaine de la couche nuageuse est uniformément déformé, ce qui évite toute intersection avec le terrain par construction. L'altitude des trajectoires appartient alors à l'intervalle  $[a + h(x, y), b + h(x, y)]$ . Dans le second cas, la surface du terrain peut se révéler plus haute que la base nuageuse, voire même traverser tout l'intervalle d'altitudes possibles. L'altitude des trajectoires appartient à l'intervalle  $[\max(a + h(x, y)), b]$ , et tout point où la surface du terrain est plus élevée que l'altitude maximale de la base nuageuse est exclu du domaine de définition des trajectoires. Le domaine horizontal des trajectoires est ainsi réduit par les plus hauts reliefs que les primitives se retrouvent obligées de contourner.

**Contraintes imposées par les vents** Une autre contrainte est la contrainte sur la vitesse maximale des vents. Les vents terrestres les plus violents jamais observés sont de l'ordre de quelques centaines de  $\text{km h}^{-1}$ , et correspondent à des conditions extrêmes telles que des tornades ou des cyclones. Aussi, nous utilisons une vitesse maximale notée  $v_{\max}$  de  $100 \text{ km h}^{-1}$  voire moins dans nos expérimentations.

**Restriction du domaine d'exploration** Il est important de noter que le domaine spatio-temporel des trajectoires est limité par le domaine horizontal du terrain, la vitesse maximale de vent

et la durée de la métamorphose. En effet, les trajectoires sont construites en fonction du terrain et ne sont donc définies qu'au-dessus de ce dernier. De plus, la vitesse maximale et l'intervalle de temps entre deux images clés consécutives limitent la longueur totale qu'une trajectoire peut atteindre.



**FIGURE 4.4.** – Visualisation de la restriction du domaine de recherche des trajectoires par érosion du domaine du terrain et dilatation des centres des sites initiaux et finaux.

Soit  $\Delta t = t_B - t_A$  l'intervalle de temps entre les deux images clés ; comme l'illustre la figure 4.4, le domaine d'exploration des trajectoires peut être restreint en appliquant une érosion de rayon  $v_{\max} \cdot \Delta t$  au domaine de définition de la fonction de hauteur.

De plus, il est tout à fait possible de se trouver dans l'incapacité de construire une trajectoire permettant de relier deux primitives trop éloignées, *id est* si  $\|\mathbf{C}_{A_i} - \mathbf{C}_j\| > v_{\max} \cdot \Delta t$ . Le domaine de recherche peut alors encore être restreint en appliquant une dilatation de rayon  $v_{\max} \cdot \Delta t$  à l'ensemble des points de départ et points d'arrivée constitués par les centres des primitives initiales et finales.

### 4.1.3 Fonction de coût

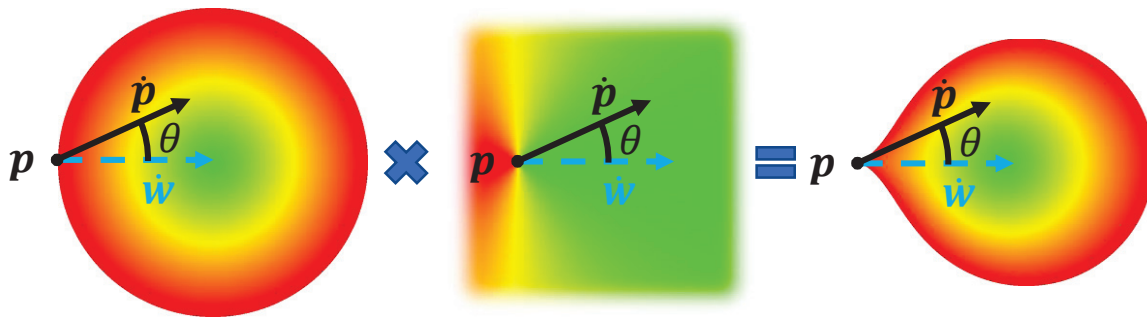
La fonction de coût quantifie à quel point un déplacement infinitésimal va à l'encontre des contraintes de l'environnement des nuages. Elle est utilisée lors du calcul des plus courts chemins entre primitives afin de guider l'exploration de l'espace des trajectoires, tandis que son intégrale sert de critère de comparaison qualitative entre trajectoires.

Les influences des vents et du relief sont exprimées séparément par des sous-fonctions de coût  $\kappa_v$ , et  $\kappa_h$  et  $\kappa_s$  respectivement. Soit  $\mathbf{p}$  une position,  $\dot{\mathbf{p}}$  le vecteur de vitesse instantanée,  $\ddot{\mathbf{p}}$  le vecteur d'accélération instantanée, et  $t$  l'instant associés à un déplacement infinitésimal entre les instants clés  $t_A$  et  $t_B$  ; la fonction de coût est une combinaison linéaire des sous-fonctions de coût

$$\kappa : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times [t_A, t_B] \rightarrow \mathbb{R}^+ \quad (4.3)$$

$$(\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}, t) \mapsto \lambda_v \kappa_v(\mathbf{p}, \dot{\mathbf{p}}, t) + \lambda_h \kappa_h(\mathbf{p}) + \lambda_s \kappa_s(\mathbf{p}, \dot{\mathbf{p}}). \quad (4.4)$$

**Fonction de coût associée au vent** La fonction de coût associée au vent mesure à quel point un déplacement infinitésimal dévie et s'écarte du flux induit par les vents. Cette mesure



**FIGURE 4.5.** – De gauche à droite : visualisation des termes d'accélération et de retournement par rapport au vent, et leur produit qui constitue la fonction de coût associée au vent.

d'écart est exprimée comme le produit de deux termes : un terme d'accélération et un terme angulaire (figure 4.5). Le terme d'accélération mesure la norme de l'accélération instantanée par rapport au flux du vent à la position et au moment du déplacement. Cela revient à mesurer la distance infinitésimale entre la position qu'aurait un mobile entraîné par le flux à partir de l'origine du déplacement, et la position après application du déplacement infinitésimal puis à diviser par la durée infinitésimale du déplacement. L'angle mesuré est l'angle de retournement contre le vent formé par la direction du flux à l'origine du déplacement considéré et la direction de déplacement considérée. Soit  $\mathbf{p}$  une position,  $\dot{\mathbf{p}}$  le vecteur vitesse instantanée, et  $t$  l'instant associés à un déplacement entre les instants clefs  $t_A$  et  $t_B$ ; la fonction de coût prenant en compte le champ vectoriel de vent  $\mathbf{v}$  s'exprime par

$$\kappa_v : \mathbb{R}^3 \times \mathbb{R}^3 \times [t_A, t_B] \rightarrow \mathbb{R}^+ \quad (4.5)$$

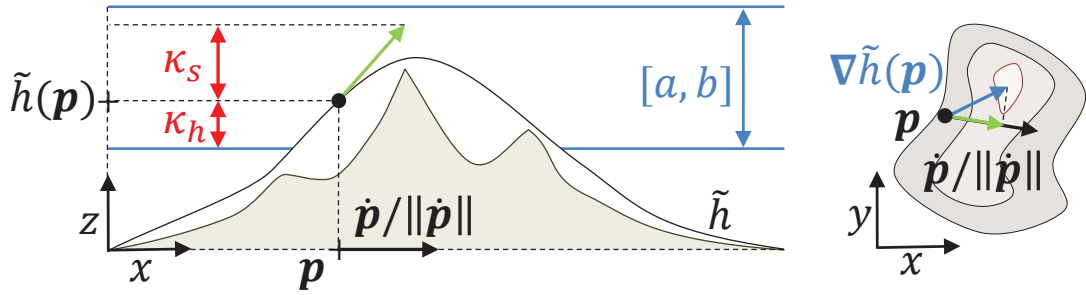
$$(\mathbf{p}, \dot{\mathbf{p}}, t) \mapsto \begin{cases} \underbrace{\|\dot{\mathbf{p}} - \mathbf{v}(\mathbf{p}, t)\|}_{\text{accélération}} \cdot \underbrace{(2 - \cos \theta)}_{\text{retournement}} & \text{si } \dot{\mathbf{p}} \neq \mathbf{0} \wedge \mathbf{v}(\mathbf{p}, t) \neq \mathbf{0} \\ \|\dot{\mathbf{p}} - \mathbf{v}(\mathbf{p}, t)\| & \text{sinon} \end{cases} \quad (4.6)$$

où

$$\cos \theta = \frac{\dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|} \cdot \frac{\mathbf{v}(\mathbf{p}, t)}{\|\mathbf{v}(\mathbf{p}, t)\|}. \quad (4.7)$$

Le terme angulaire a pour objectif de défavoriser encore plus fortement les mouvements qui se retournent contre les vents mais ce terme est indéfini en l'absence de vent ou de vitesse.

**Fonctions de coût associées au terrain** Les fonctions  $\kappa_h$  et  $\kappa_s$  concernent toutes les deux la prise en compte du terrain dans le calcul du coût d'un déplacement (figure 4.6 page suivante). Les primitives nuageuses interpolantes qui constituent la fonction de présence en métamorphose sont définies comme de larges ellipsoïdes dont les centres suivent des trajectoires à déterminer. Aussi, afin d'éviter que les moindres détails du relief n'influencent les trajectoires des primitives interpolantes, la version détaillée et haute résolution de la fonction de hauteur du terrain notée  $h$  n'est pas utilisée directement. À la place, une version lissée et sous-échantillonnée de la carte de hauteur est calculée. Le terrain lissé est ensuite étiré de manière à conserver le même point le plus haut que le terrain d'origine.



**FIGURE 4.6.** – Visualisation des termes intervenant dans les fonctions de coût associées au terrain  $\kappa_h$  et  $\kappa_s$ .

Soit  $\mathbf{p}$  une position,  $\tilde{h}$  la version lissée du terrain et  $[a, b]$  l'intervalle d'altitudes de la base nuageuse ;  $\kappa_h$  quantifie la compression de l'intervalle d'altitudes par le terrain :

$$\kappa_h : \mathbb{R}^2 \rightarrow \mathbb{R}^+ \quad (4.8)$$

$$\mathbf{p} \mapsto \max(\tilde{h}(\mathbf{p}) - a, 0). \quad (4.9)$$

Ce coût n'est positif et n'a donc d'intérêt que si l'altitude de la base nuageuse est absolue car une altitude relative à la surface du terrain empêche le terrain de traverser la couche nuageuse (section 4.1.2 page 105, figure 4.3 page 106).

La seconde fonction de coût prenant en compte le terrain est  $\kappa_s$ , et quantifie la variation d'altitude due au terrain lors d'un déplacement horizontal infinitésimal. Soit  $\nabla \tilde{h}(\mathbf{p})$  le gradient du terrain lissé, le terme de pente peut être écrit comme :

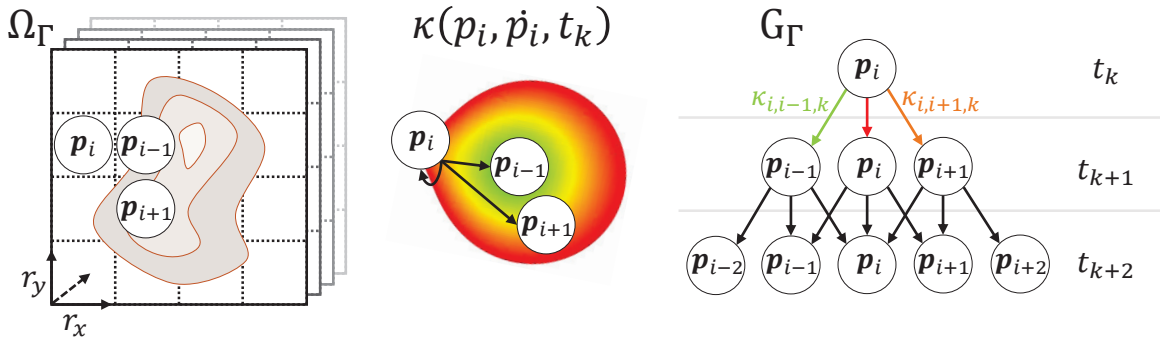
$$\kappa_s : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^+ \quad (4.10)$$

$$(\mathbf{p}, \dot{\mathbf{p}}) \mapsto \left| \nabla \tilde{h}(\mathbf{p}) \cdot \frac{\dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|} \right|. \quad (4.11)$$

Il est important de noter l'utilisation de la valeur absolue dans cette expression, ce qui assure le même coût positif, que le mouvement soit dans le sens ascendant ou descendant de la pente. D'autres expressions sont envisageables, mais il est nécessaire que le coût soit positif afin d'éviter les cycles dans la recherche de plus court chemin (section 4.1.4 page suivante). Il est par exemple possible d'appliquer une pondération différente suivant l'angle entre le déplacement et la pente pour favoriser les pentes descendantes en remplaçant les valeurs négatives de  $\kappa_s$  par zéro, ou encore d'appliquer un coefficient différent suivant le sens de la pente afin que les nuages nécessitent plus d'énergie pour se déplacer en montée qu'en descente.

**Coût d'une trajectoire** La fonction de coût donne le coût d'un déplacement infinitésimal en tout point d'après les conditions de vents et le terrain. La quantification du coût total d'une trajectoire vis-à-vis de l'environnement est alors l'intégrale de la fonction de coût le long de





**FIGURE 4.7.** – Aperçu de la structure du graphe de recherche de plus court chemin avec, de gauche à droite : la grille du domaine spatio-temporel  $\Omega_\Gamma$  dont les cellules forment les nœuds ; la fonction de coût  $\kappa$  qui pondère les arcs ; une partie du graphe  $\mathcal{G}_\Gamma$  orienté, acyclique et pondéré de recherche.

l'abscisse curviligne de la trajectoire à évaluer. Soit  $\Gamma$  une trajectoire de classe  $C^2$  entre les instants  $t_A$  et  $t_B$  ; la fonctionnelle de coût de la trajectoire s'écrit

$$K : ([t_A, t_B] \rightarrow \mathbb{R}^3) \rightarrow \mathbb{R}^+ \quad (4.12)$$

$$\Gamma \mapsto \int_{t_A}^{t_B} \kappa(\Gamma(t), \dot{\Gamma}(t), \ddot{\Gamma}(t), t) dt. \quad (4.13)$$

#### 4.1.4 Calcul de plus court chemin spatio-temporel anisotrope

Pour chaque couple composé d'une primitive initiale et d'une primitive finale, le plus court chemin qui les relie est calculé en utilisant la fonction de coût comme distance. Soit  $\mathbf{a} = (\mathbf{C}_{A_i}, t_A)$  et  $\mathbf{b} = (\mathbf{C}_{B_j}, t_B)$  les positions spatio-temporelles initiales et finales, et  $\Omega_\Gamma$  le domaine des chemins de classe  $C^2$  qui relie  $\mathbf{a}$  et  $\mathbf{b}$  ; le problème de plus court chemin anisotrope continu consiste à trouver un chemin qui minimise la fonctionnelle de coût :

$$\Gamma_{i,j}^* = \operatorname{argmin}_{\Gamma \in \Omega_\Gamma} K(\Gamma). \quad (4.14)$$

Afin de déterminer une solution approchée au problème de plus court chemin anisotrope continu, le domaine des chemins est discrétisé en un graphe fini et, suivant la finesse de la discrétisation employée, des tests supplémentaires sont appliqués pour éviter des trajectoires absurdes. Ensuite, un plus court chemin est déterminé grâce à l'algorithme classique de Dijkstra. L'algorithme A\* a aussi été implémenté, mais s'il peut se révéler plus efficace que l'algorithme de Dijkstra dans le cas d'une unique recherche de plus court chemin, il ne permet pas d'optimiser facilement les recherches incrémentales.

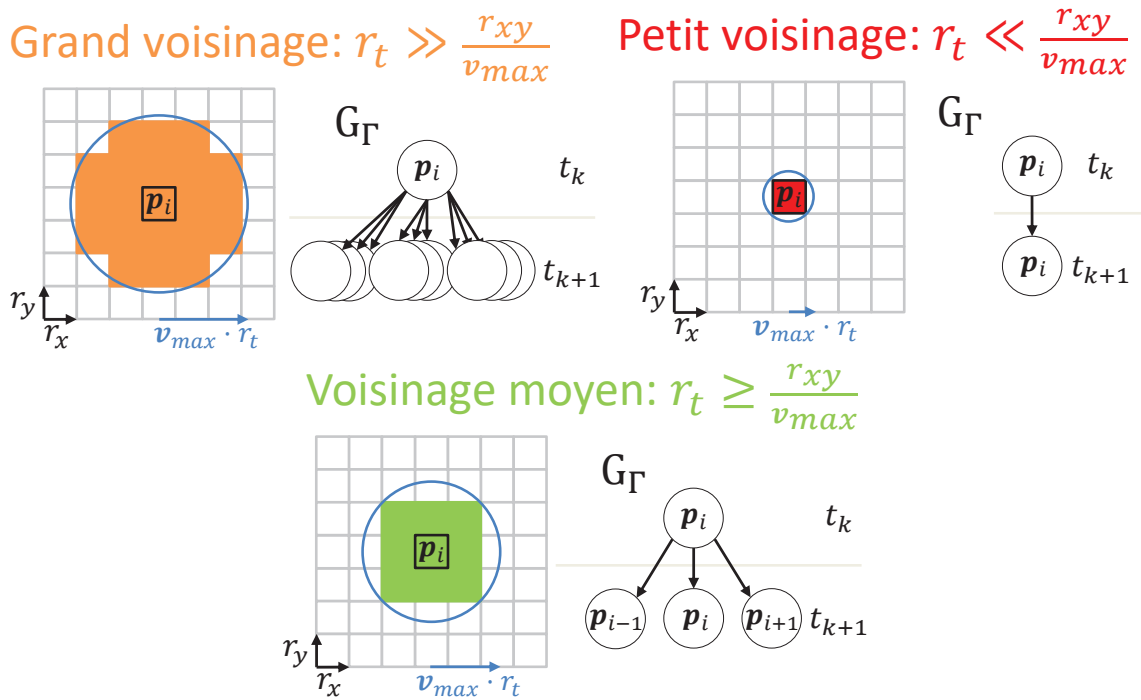
**Construction du graphe** La boîte englobante spatio-temporelle  $\Omega \times T \supseteq \Omega_\Gamma$  du domaine continu des chemins  $\Omega_\Gamma$  est discrétisée sous la forme d'une grille régulière (figure 4.7). L'intervalle de temps  $T$  est subdivisé en pas de temps de durée  $r_t$ . Quant au rectangle spatial  $\Omega$ , il est subdivisé en cellules de dimensions  $r_x \times r_y$ .

Chaque cellule de la grille obtenue constitue un nœud  $(\mathbf{p}_i, t_k)$  du graphe de recherche. Les arcs du graphe représentent les déplacements discrets. Ils sont implicitement définis entre chaque couple de cellules–nœuds chronologiquement voisines  $(\mathbf{p}_i, t_k)$  et  $(\mathbf{p}_j, t_{k+1})$  et pondérés d’après la fonction de coût par  $\kappa(\mathbf{p}_0, \frac{\mathbf{p}_1 - \mathbf{p}_0}{r_t}, t_0) \times r_t$ .

Enfin, en prenant en compte la vitesse limite  $v_{\max}$ , le voisinage d’une cellule de coordonnées spatio-temporelles  $(\mathbf{p}, t) \in \Omega_\Gamma$  est constitué de l’ensemble des cellules qui :

- ont pour coordonnée temporelle  $t + r_t$ ,
- et ont des coordonnées spatiales qui appartiennent à la boule de rayon  $r_t \times v_{\max}$ , *id est*  $\{\mathbf{p} + \mathbf{v} \in \Omega \mid \|\mathbf{v}\| \leq r_t \times v_{\max}\}$ .

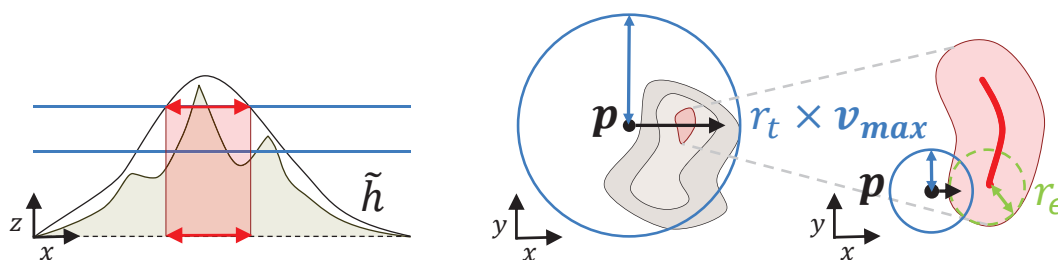
Pour rappel, le plus court chemin est calculé en seulement trois dimensions : deux dimensions spatiales plus la dimension temporelle. En effet, l’altitude n’est pas échantillonnée, ce qui amènerait une quatrième dimension. À la place, l’altitude et la pente du terrain sont prises en compte lors du calcul du coût associé aux arcs (Figure 4.6 page 109) puis les trajectoires sont déformées pour suivre la surface du terrain comme dans [39].



**FIGURE 4.8.** – Influence du pas de temps  $r_t$  sur la taille du voisinage spatial de chaque nœud du graphe de recherche, relativement à la vitesse maximale de vent  $v_{\max}$  et à la taille de maille spatiale  $r_{xy}$ . De gauche à droite : un grand pas de temps ; un pas de temps moyen ; un pas de temps si petit que le voisinage est réduit au nœud initial.

**Finesse de la discrétisation du domaine** La discrétisation du domaine de recherche spatio-temporel nécessite une attention particulière afin d’y trouver des chemins valides. En effet, la discrétisation du domaine temporel  $T$  conditionne directement le rayon spatial  $r_v$  du voisinage proportionnellement à la vitesse maximale de déplacement :  $r_v = r_t \times v_{\max}$ . Ainsi, plus le pas de temps est court plus l’arité de chaque cellule nœud diminue (figure 4.8). Un pas de temps plus court que  $\min(r_x, r_y)/v_{\max}$  réduit le voisinage à la cellule du pas de temps suivant qui partage exactement les mêmes coordonnées spatiales, *id est* le seul déplacement possible est de rester immobile. Au contraire, plus le pas de temps est long,

plus le voisinage est grand et plus il y a de choix dans la fraction de vitesse maximale à adopter.



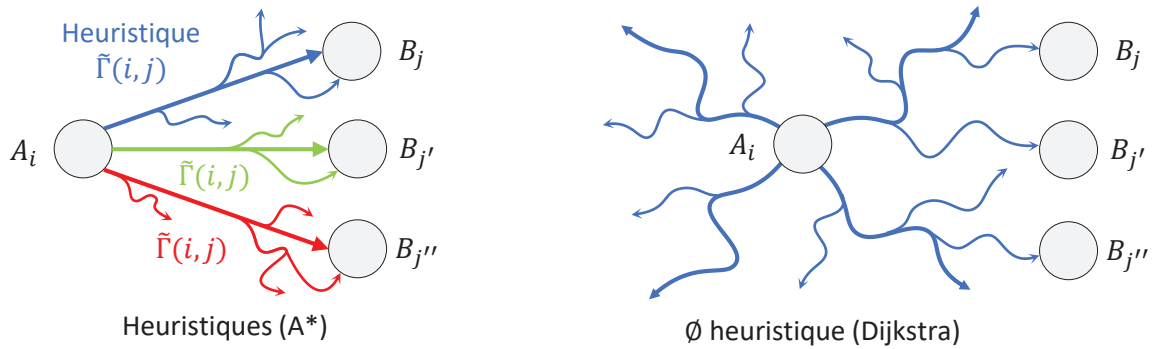
**FIGURE 4.9.** – Influence du pas de  $r_t$  relativement à la vitesse maximale de vent  $v_{\max}$  et à la taille minimale  $r_\epsilon$  des obstacles constitués par le terrain (à gauche). Au milieu : un pas de temps trop grand qui permet de passer à travers l'obstacle. À droite : un pas de temps assez petit pour empêcher de traverser les obstacles.

Malheureusement, un pas de temps trop long peut causer des erreurs dans la recherche de chemin en permettant de traverser un obstacle comme un pic rocheux de terrain (figure 4.9). Afin d'éviter ces erreurs, il y a deux solutions. La première consiste à choisir un pas de temps assez court pour que la vitesse maximale parcourue soit inférieure à l'épaisseur du plus petit obstacle, ce qui empêche de traverser l'obstacle en un seul déplacement, et permet donc de le détecter en tombant dessus. Le problème est alors que la taille du graphe modélisé grossit rapidement et avec elle la complexité de la recherche de chemin.

La seconde solution consiste à discrétiser normalement le domaine spatio-temporel pour la définition du graphe de recherche, en vérifiant néanmoins finement pour chaque déplacement envisagé s'il n'y a pas intersection avec un obstacle le long du vecteur déplacement.

**Algorithme de Dijkstra** L'algorithme de Dijkstra applique le principe de l'inondation, *id* est un front de progression inonde le graphe depuis le nœud de départ appelé source jusqu'à atteindre le nœud d'arrivée appelé puits ou jusqu'à avoir exploré l'ensemble de la composante connexe de la source sans trouver de chemin menant au puits. Le front est constitué de l'ensemble des nœuds mouillés mais non immergés (ou entourés), *id* est les nœuds qui ont déjà été atteints par l'exploration mais dont certains voisins n'ont pas encore été parcourus. Il est initialisé avec la source qui est considérée comme à distance nulle d'elle-même. Le voisin le plus proche de l'ensemble du front lui est ajouté et le front est mis à jour ainsi que la plus courte distance à chaque nœud exploré. La recherche se termine lorsque le puits totalement immergé et quitte le front et sa distance constitue la distance totale du plus court chemin. Il ne reste plus qu'à reconstruire le dit plus court chemin en remontant la chaîne des voisins à plus courte distance de l'origine jusqu'à atteindre le nœud de départ.

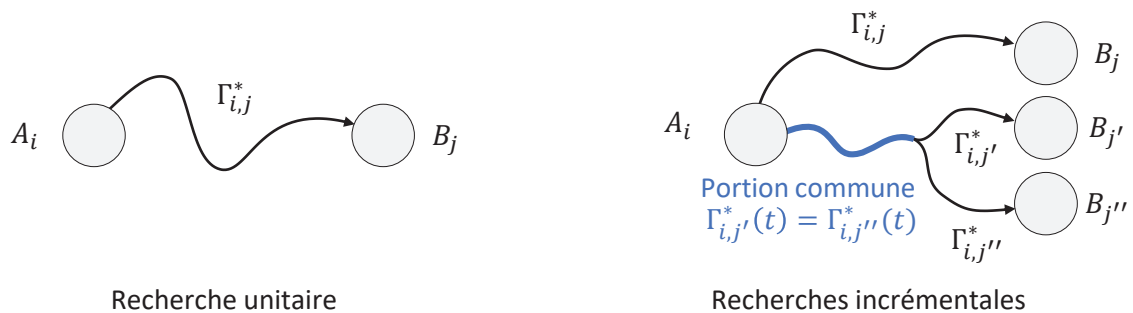
**Algorithme A\*** L'algorithme A\* est une amélioration de l'algorithme de Dijkstra qui exploite une heuristique évaluant la distance à l'arrivée (figure 4.10 page suivante). La valeur de l'heuristique est ajoutée à la distance déjà parcourue, ce qui favorise les nœuds proches du puits. Si l'heuristique est bonne, l'algorithme A\* nécessite d'explorer moins de nœuds que l'algorithme de Dijkstra, ce qui accélère la recherche si le surcoût de l'heuristique n'est pas trop important. Il faut cependant prendre garde à l'admissibilité de l'heuristique employée ou la



**FIGURE 4.10.** – Visualisation des algorithmes de recherche de plus court chemin : Dijkstra sans heuristique (droite), et A\* avec les heuristiques (gauche).

recherche peut terminer avec un chemin qui n'est pas le plus court. Pour être admissible, une heuristique ne doit jamais surestimer la distance restante à parcourir. Dans le cas classique d'un graphe plongé dans l'espace, la distance euclidienne est une heuristique admissible, peu coûteuse et qui donne un bon avantage à l'algorithme A\* comparé à l'algorithme de Dijkstra. En revanche, une bonne heuristique est plus complexe à définir pour le graphe associé à une couche nuageuse et pondéré par la fonction de coût, notamment à cause des vents.

**Recherches incrémentales** Dans le cas d'une unique recherche de plus court chemin, une bonne heuristique rend l'algorithme A\* bien plus efficace que l'algorithme de Dijkstra. En revanche, lorsque la recherche vise à trouver les plus courts chemins de tous les couples d'un ensemble de nœuds, l'algorithme de Dijkstra peut être réutilisé incrémentalement, ce qui est plus difficile pour l'algorithme A\*. En effet, la distance restante à parcourir qui est prédite par l'heuristique de l'algorithme A\* dépend du puits. Après une première recherche d'une source vers un puits, changer de puits nécessite d'invalider les distances en chaque nœud et de repartir de zéro pour appliquer l'algorithme A\*.



**FIGURE 4.11.** – Le passage d'une unique recherche de plus court chemin entre deux sites (gauche) à de multiples plus courts chemins d'un site vers plusieurs permet de factoriser les portions communes entre les plus courts chemins.

Au contraire, l'algorithme de Dijkstra peut reprendre l'inondation où elle en était et la poursuivre tant que le nouveau puits n'est pas immergé. Pour cette raison, c'est l'algorithme de Dijkstra qui est utilisé dans le contexte de la métamorphose de paysage nuageux. Les trajectoires depuis une primitive initiale vers l'ensemble des primitives finales sont calculées itérativement en changeant successivement le puits (figure 4.11). Pour chaque primitive initiale, le lot de recherches itératives est indépendant et peut être traité en parallèle ou

itérativement. L'ensemble des trajectoires entre tous les couples de primitives initiales et finales est ainsi calculé plus efficacement.

## 4.2 Mise en correspondance optimale

Afin de définir la métamorphose entre deux images clefs  $(A, B)$ , leurs primitives sont mises en correspondance par un graphe de correspondance. Ce graphe est un graphe biparti orienté et pondéré dont :

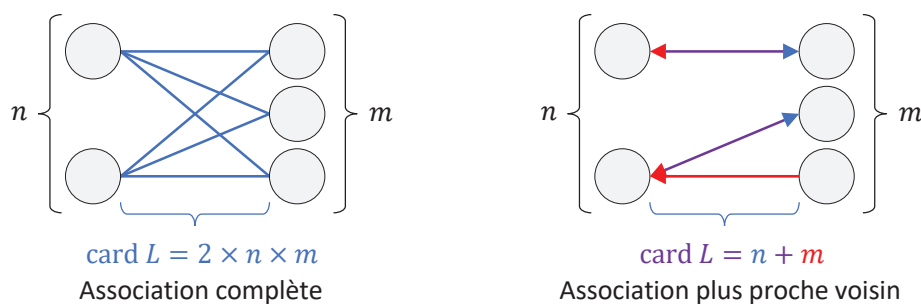
- les deux types de nœuds représentent les primitives initiales et finales,
- les arcs représentent les relations de correspondance,
- et la pondération de chaque arc représente la portion de masse nuageuse répartie sur ce lien.

Par convention, l'ensemble des arcs orientés dans l'ordre chronologique, *id est* depuis une primitive initiale vers une primitive finale est noté  $L_{\rightarrow} \subseteq A \times B$ . Dans l'ordre chronologique inverse, l'ensemble des arcs est noté  $L_{\leftarrow} \subseteq B \times A$ . Chaque paire de primitives reliées par un arc donne lieu à la création d'une seule primitive interpolante, que la relation de correspondance soit symétrique (un arc dans chaque sens) ou non (un seul arc). Pour limiter le nombre de primitives interpolantes, il convient donc notamment de favoriser les relations de correspondance symétriques.

La construction automatique d'un graphe de correspondance cohérent est un problème fondamental de la métamorphose pour lequel des heuristiques ont déjà été proposées [38], et nous en rappelons les plus simples.

Nous proposons une nouvelle heuristique de mise en correspondance, inspirée des travaux de BONNEEL *et al.* [11] sur l'interpolation de distributions par transport optimal. L'idée consiste à voir la mise en correspondance sous la forme du plan de transport optimal discret de la masse nuageuse le long des plus courts chemins les plus adaptés à la situation.

### 4.2.1 Heuristiques de mise en correspondance



**FIGURE 4.12.** – Schématisation des heuristiques d'association du processus de mise en correspondance : heuristique complète (gauche), heuristique des plus proches voisins (droite).



La construction du graphe de correspondance peut être séparée en deux sous-problèmes :

- l'association qui détermine l'ensemble des arcs  $L = L_{\rightarrow} \cup L_{\leftarrow}$ ,
- et la pondération qui attribue les poids  $w_{i \rightarrow j}$  et  $w_{i \leftarrow j}$  aux arcs.

L'association répond qualitativement au problème de mise en correspondance, tandis que la pondération y répond quantitativement.

**Graphe complet** Une première heuristique de mise en correspondance consiste à associer indifféremment toutes les primitives initiales à toutes les primitives finales (figure 4.12 page précédente). Le graphe complet a ainsi pour arcs :

$$L_* = A \times B \cup B \times A \quad (4.15)$$

et représente  $n_A n_B$  relations de correspondance symétriques pour  $n_A$  primitives initiales et  $n_B$  primitives finales.

Cette heuristique est simple à mettre en œuvre car elle ne différencie pas les primitives à associer, *id est* aucune information complémentaire sur les primitives n'est nécessaire. Cela constitue aussi sa plus grande limitation car il n'y a du coup aucune cohérence apportée.

**Graphe des plus proches voisins** Une seconde heuristique de mise en correspondance consiste à associer les primitives initiales et finales les plus proches selon une fonction de distance donnée (figure 4.12 page ci-contre). La fonction de distance peut par exemple être la distance euclidienne ou bien le coût du plus court chemin entre les centres des primitives (section 4.1.4 page 110). Soit  $d$  la fonction de distance ; le graphe des plus proches voisins a ainsi pour arcs

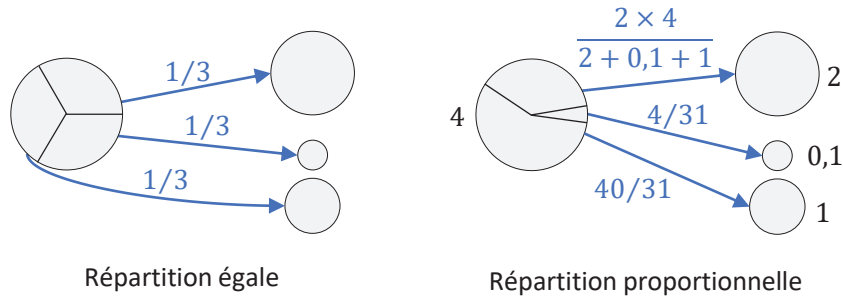
$$L_{\rightarrow} = \left\{ (A_i, B_j) \in A \times B \mid d(\mathbf{C}_{A_i}, \mathbf{C}_j) = \min_{B_k \in B} d(\mathbf{C}_{A_i}, \mathbf{C}_k) \right\}, \text{ et} \quad (4.16)$$

$$L_{\leftarrow} = \left\{ (B_j, A_i) \in B \times A \mid d(\mathbf{C}_j, \mathbf{C}_{A_i}) = \min_{A_k \in A} d(\mathbf{C}_j, \mathbf{C}_k) \right\}. \quad (4.17)$$

La relation de plus proche voisin n'est pas nécessairement symétrique, et il est même possible qu'une primitive possède plusieurs voisins à distance minimale. En ne considérant qu'un seul plus proche voisin par primitive, le graphe contient alors  $n_A + n_B$  arcs, soit entre  $\max(n_A, n_B)$  et  $n_A + n_B$  primitives interpolantes.

Comparée à l'heuristique complète, cette heuristique demande un peu plus d'informations sous la forme d'une métrique entre primitives initiales et finales. Mais n'associer que les plus proches voisins a l'avantage d'apporter un peu de cohérence à la métamorphose en évitant les associations les plus aberrantes. Cela permet aussi de limiter le nombre de primitives interpolantes nécessaires.

**Pondération équilibrée** Une première heuristique de pondération consiste à répartir équitablement la masse nuageuse de chaque primitive sur l'ensemble de ses liens de correspon-



**FIGURE 4.13.** – Schématisation des heuristiques de pondération du processus de mise en correspondance : pondération équilibrée (gauche), pondération proportionnelle (droite).

dance (figure 4.13). Soit  $d^+$  le degré sortant d'un nœud ; la pondération équilibrée associe aux arcs les poids :

$$w_{i \rightarrow j} = \frac{1}{d^+(A_i)} \quad \forall (A_i, B_j) \in L_{\rightarrow}, \text{ et} \quad (4.18)$$

$$w_{i \leftarrow j} = \frac{1}{d^+(B_j)} \quad \forall (B_j, A_i) \in L_{\leftarrow}. \quad (4.19)$$

Cette heuristique de pondération est simple et ne nécessite aucune autre information que la valence de chaque nœud du graphe de mise en correspondance. De manière similaire à l'heuristique d'association complète, cette heuristique de pondération équilibrée est totalement adaptée en l'absence d'information complémentaire sur les différentes primitives. Dans le cas contraire, elle se révèle incapable de valoriser la métamorphose.

**Pondération proportionnelle** Une seconde heuristique de pondération consiste à répartir la masse nuageuse de chaque primitive sur l'ensemble de ses liens de correspondance proportionnellement à un ratio d'importance tel que le ratio de masse nuageuse (figure 4.13). Soit  $m_{A_i}$  et  $m_{B_j}$  les masses d'une primitive initiale et d'une primitive finale ; la pondération proportionnelle associe aux arcs les poids

$$w_{i \rightarrow j} = \frac{m_{B_j}}{\sum_{B_k \in B} m_{B_k}} \quad \forall (A_i, B_j) \in L_{\rightarrow}, \text{ et} \quad (4.20)$$

$$w_{i \leftarrow j} = \frac{m_{A_i}}{\sum_{A_k \in A} m_{A_k}} \quad \forall (B_j, A_i) \in L_{\leftarrow}. \quad (4.21)$$

Afin d'appliquer cette heuristique de pondération, il est nécessaire de calculer la masse de chaque primitive et le ratio de sa masse sur la masse totale de son image clef. L'intérêt de favoriser les primitives les plus massives est notamment de minimiser les variations de masse des primitives interpolantes. L'animation est alors plus cohérente du point de vue de la répartition de la masse.

## 4.2.2 Heuristique optimale

Le problème original du transport optimal considère une répartition de sable initiale et cherche à déterminer comment déplacer ce sable pour obtenir une autre répartition objectif

en minimisant l'effort nécessaire. Un ensemble valide de déplacements de sable conserve la quantité initiale et est appelé plan de transport, et un plan de transport qui minimise l'effort nécessaire est qualifié d'optimal. Dans le cas général, la quantité de sable est remplacée par une quantité de masse répartie dans un domaine continu par une fonction de densité. Le cas du transport optimal discret considère un domaine discret, composé d'un nombre de sites fini.

Nous appliquons les méthodes issues du transport optimal discret au problème de mise en correspondance afin construire la métamorphose de la présence nuageuse (figure 4.14). La masse considérée est alors la masse nuageuse normalisée, et les sites où elle se concentre sont les centres des primitives statiques. Le coût de transport d'une unité de masse est quant à lui donné par le coût du plus court chemin calculé d'après les contraintes de l'environnement nuageux (section 4.1.4 page 110).

Le plan de transport est représenté par une matrice dont chaque élément donne la quantité de masse déplacée entre une primitive initiale et une primitive finale :

$$\mathbf{M} = (m_{i,j})_{(i,j) \in \llbracket 1, n_A \rrbracket \times \llbracket 1, n_B \rrbracket}, \quad (4.22)$$

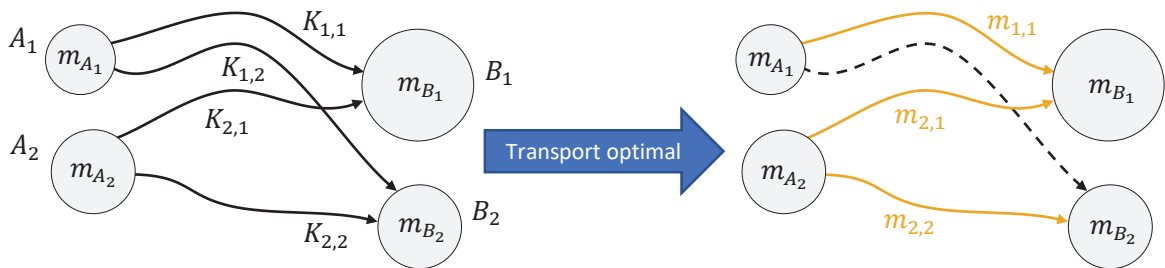
tout en conservant la masse :

$$\sum_{i=1}^{n_A} \sum_{j=1}^{n_B} m_{i,j} = \sum_{i=1}^{n_A} m_{A_i} = \sum_{i=1}^{n_B} m_{B_j}. \quad (4.23)$$

Le coût d'un plan de transport est donné par la fonctionnelle :

$$\kappa_{OT}(\mathbf{M}) = \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} m_{i,j} \cdot \kappa \left( \Gamma^* \left( \mathbf{C}_{A_i}, \mathbf{C}_{B_j} \right) \right), \quad (4.24)$$

et le plan de transport optimal est alors la matrice  $\mathbf{M}^* = \operatorname{argmin}(\kappa_{OT})$  qui minimise le coût global de transport.



**FIGURE 4.14.** – Schématisation du processus de mise en correspondance optimale : d'après les coûts  $K_{i,j}$  des plus courts chemins  $\Gamma^*_{i,j}$  de chaque relation de correspondance possible  $(A_i, B_j)$ , le sous-ensemble des relations choisies et leur pondération garantit le transport optimal des masses nuageuses initiales  $m_{A_i}$  et finales  $m_{B_j}$ .

**Masse nuageuse d'une primitive** Soit  $\mathcal{C}$  une primitive de contrôle de densité maximale  $D$  ; sa masse  $m$  est définie comme l'intégrale de son champ de densité  $c$  sur son support compacte  $\Omega$ ,  $id$  est la fonctionnelle

$$m = M(\mathcal{C}) = D \cdot \iiint_{\Omega} c(\mathbf{p}) \, dV. \quad (4.25)$$

Le support compact  $\Omega$  des primitives de présence est une ellipsoïde orientée selon les axes canoniques et son rayon horizontal est noté  $R_H$  tandis que son rayon vertical est noté  $R_V$  (section 3.2 page 92). L'expression de la masse est alors :

$$\begin{aligned}
M(\mathcal{C}) &= D \cdot \iiint_{\Omega} c(\mathbf{p}) \, dx \, dy \, dz & (4.26) \\
&= D \cdot \int_0^{\pi} \int_0^{2\pi} \int_0^1 c(\mathbf{p}) \left| \frac{\partial(x, y, z)}{\partial(r, \theta, \bar{\varphi})} \right| \, dr \, d\theta \, d\bar{\varphi} \\
&= D \cdot \int_0^{\pi} \int_0^{2\pi} \int_0^1 c(\mathbf{p}) \begin{vmatrix} a \cos \theta \sin \bar{\varphi} & -ar \sin \theta \sin \bar{\varphi} & ar \cos \theta \cos \bar{\varphi} \\ b \sin \theta \sin \bar{\varphi} & br \cos \theta \sin \bar{\varphi} & ar \sin \theta \cos \bar{\varphi} \\ c \cos \bar{\varphi} & 0 & -cr \sin \bar{\varphi} \end{vmatrix} \, dr \, d\theta \, d\bar{\varphi} \\
&= D \cdot abc \int_0^{\pi} \int_0^{2\pi} \int_0^1 c(\mathbf{p}) r^2 \sin \bar{\varphi} \, dr \, d\theta \, d\bar{\varphi} \\
&= D \cdot 4\pi abc \int_0^1 c(\mathbf{p}) r^2 \, dr \\
&= D \cdot 4\pi R_V R_H^2 \int_0^1 g \circ d(\mathbf{p}) r^2 \, dr \\
&= D \cdot 4\pi R_V R_H^2 \int_0^1 g(r) r^2 \, dr. & (4.27)
\end{aligned}$$

Suivant la fonction d'atténuation utilisée, l'intégration donne différentes expressions :

$$M(\mathcal{C}) = \begin{cases} D \cdot \frac{4}{3 \times 5} \pi R_V R_H^2 & \text{pour } g(r) = -2(1-r)^3 + 3(1-r)^2 \\ D \cdot \frac{4}{3 \times 20} \pi R_V R_H^2 & \text{pour } g(r) = (1-r)^3 \end{cases}. \quad (4.28)$$

**Normalisation de masse totale** Le calcul du plan de transport optimal s'applique directement lorsqu'il y a conservation de la masse entre les sites initiaux et finaux. Or, la masse nuageuse totale de chaque image clef n'est pas nécessairement constante. C'est pourquoi nous normalisons la masse nuageuse avons d'appliquer l'algorithme de transport optimal qui travaille alors plutôt sur les ratios de masse nuageuse

$$\tilde{m}_{A_i} = \frac{m_{A_i}}{\sum_{k \in \llbracket 1, n_A \rrbracket} m_{A_k}} \quad \forall i \in \llbracket 1, n_A \rrbracket, \text{ et} \quad (4.29)$$

$$\tilde{m}_{B_j} = \frac{m_{B_j}}{\sum_{k \in \llbracket 1, n_B \rrbracket} m_{B_k}} \quad \forall j \in \llbracket 1, n_B \rrbracket. \quad (4.30)$$

**Grphe optimal** L'ensemble des relations de correspondance obtenues par transport optimal sont symétriques et correspondent aux coefficients non nuls du plan de transport

$$L_{\rightarrow} = \{(A_i, B_j) \in A \times B \mid m_{i,j} > 0\}, \text{ et} \quad (4.31)$$

$$L_{\leftarrow} = \{(B_j, A_i) \in B \times A \mid m_{i,j} > 0\}. \quad (4.32)$$

Une propriété notable du transport optimal discret est que le nombre de déplacements de masse augmente linéairement avec le nombre total de sites de manière similaire à l'heuristique de mise en correspondance des plus proches voisins (section 4.2.1 page 114)

$$\text{card } L_{\rightarrow} = \text{card } L_{\leftarrow} \leq n_A + n_B - 1. \quad (4.33)$$

En plus de réduire la complexité du model interpolant, le nombre limité de relations de correspondance permet à l'optimisation globale du transport optimal de sélectionner les meilleures trajectoires d'après la métrique de coût. Les trajectoires les moins coûteuses sont choisies en priorité tandis que les trajectoires les plus coûteuses ne sont choisies que lorsqu'elles sont nécessaires au transport, *id est* lorsqu'il n'y a pas d'alternative pour déplacer la masse nuageuse.

**Pondération optimale** Les coefficients de pondération des liens du graphe optimal sont proportionnels à la masse des primitives et aux ratios de masse du plan de transport optimal

$$w_{i \rightarrow j} = m_{i,j} \cdot \tilde{m}_{A_i} \quad \forall (A_i, B_j) \in L, \text{ et} \quad (4.34)$$

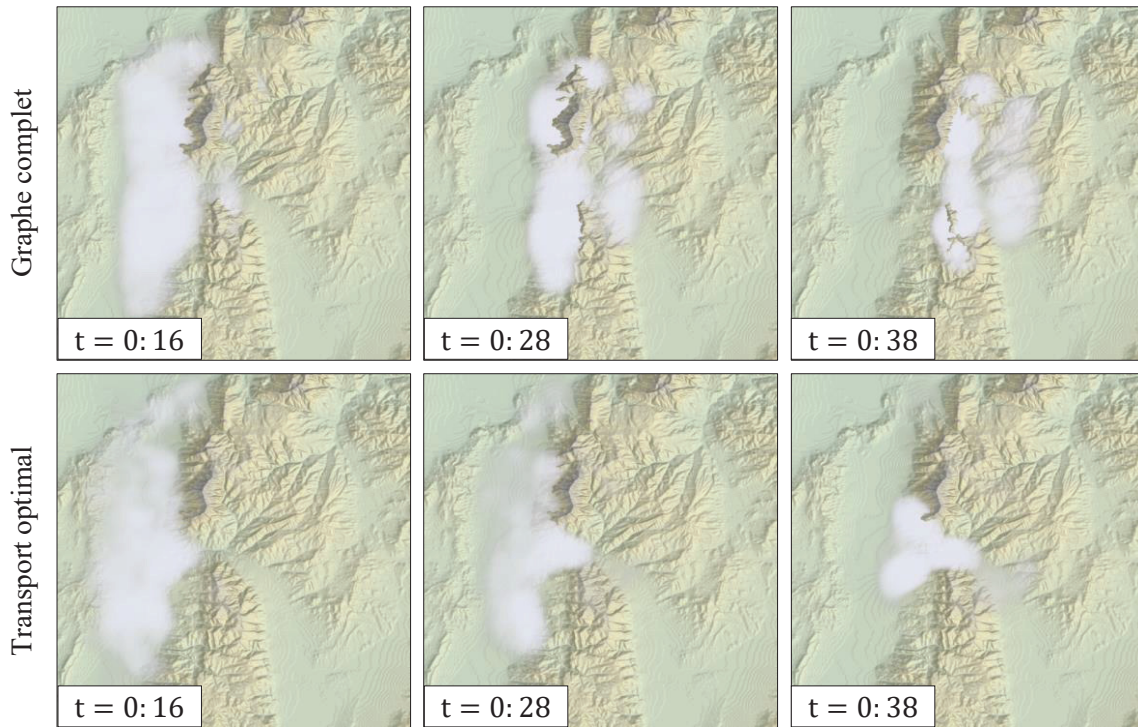
$$w_{i \leftarrow j} = m_{i,j} \cdot \tilde{m}_{B_j} \quad \forall (A_i, B_j) \in L. \quad (4.35)$$

Plutôt que de distribuer plus de masse d'une primitive d'une image clef aux primitives associées les plus denses comme pour l'heuristique de distribution proportionnelle (section 4.2.1 page 114), l'intérêt supplémentaire du transport optimal est de prendre en compte le coût de transport pour distribuer plus de masse le long des meilleures trajectoires. L'optimisation globale du transport optimal permet ainsi non seulement de limiter le nombre de primitives interpolantes, mais aussi de limiter la distribution de masse nuageuse le long des trajectoires les plus coûteuses.

### 4.3 Ajout de primitives fantômes

Lors de la recherche de plus court chemin entre une primitive initiale et une primitive finale (section 4.1 page 105), il est parfois impossible de trouver un seul chemin qui satisfasse les contraintes imposées à la métamorphose. De plus, certains des plus courts chemins calculés, bien que valides, peuvent se révéler coûteux et de piètre qualité et tout de même être utilisés par le plan de transport optimal (section 4.2 page 114). En effet, le transport optimal minimise globalement le transport de la masse nuageuse le long des chemins problématiques mais lorsque ce sont les seuls valides, il n'a pas d'autre choix (figure 4.16 page 121).





**FIGURE 4.15.** – La mise en correspondance complète génère de nombreuses primitives de contrôle et leurs trajectoires ne prennent pas en compte le champ de vent ni le relief du terrain. Au contraire, l'approche proposée génère bien moins de primitives et avec des trajectoires cohérentes.

De manière générale, les correspondances problématiques sont dues à l'isolement d'un ensemble de primitives d'une image clef par rapport à l'autre image clef (Figure 4.17 page ci-contre). En ne considérant que les conditions de vents, une telle situation se produit par exemple lorsque des primitives initiales se situent en aval du flux du vent par rapport aux primitives finales, ou à l'inverse, lorsque des primitives finales se trouvent en amont par rapport aux primitives initiales.

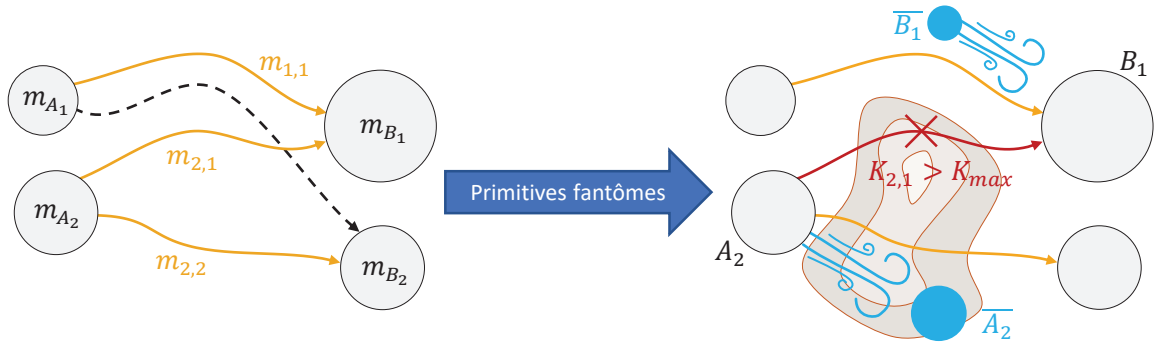
### 4.3.1 Seuillage du coût des trajectoires

Afin de corriger les trajectoires obtenues par le transport optimal qui s'avèrent irréalistes dans les cas les plus problématiques, une valeur seuil  $\kappa_{\max}$  est choisie, et les trajectoires dont le coût  $y$  est supérieur font partie des liens invalidés

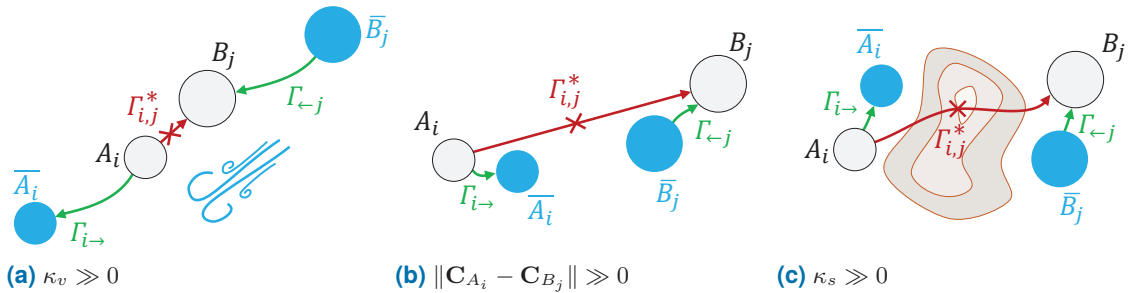
$$\overline{L_{\rightarrow}} = \left\{ (A_i, B_j) \in L_{\rightarrow} \mid K(\Gamma^*(C_{A_i}, C_{B_j})) > \kappa_{\max} \right\}, \text{ et} \quad (4.36)$$

$$\overline{L_{\leftarrow}} = \left\{ (B_j, A_i) \in L_{\leftarrow} \mid K(\Gamma^*(C_{A_i}, C_{B_j})) > \kappa_{\max} \right\}. \quad (4.37)$$

Il ne suffit pas de supprimer les primitives interpolantes définies sur les trajectoires invalidées, ou la cohérence de l'animation et la conservation de la masse seraient violées. Les primitives interpolantes concernées subissent plutôt un traitement différent : elles sont retirées de la métamorphose et les masses associées aux primitives initiales et finales qu'elles auraient interpolées sont inventoriées. Pour chaque primitive dont une portion de la masse



**FIGURE 4.16.** – Après l’application du transport optimal, les trajectoires ayant un coût supérieur à un seuil sont éliminées (gauche). Des fantômes sont insérés selon la trajectoire la moins coûteuse trouvée durant la recherche de chemin (droite).



**FIGURE 4.17.** – Exemples de cas problématiques nécessitant la création de primitives fantômes : des primitives séparées par un vent contraire trop puissant (a) ; des primitives trop éloignées l’une de l’autre (b) ; des primitives séparées par un obstacle trop grand (c).

est inventoriée comme orpheline, cette portion est interpolée de manière indépendante de l’image clef opposée.

Pour chaque primitive dont une partie de la masse est orpheline, une nouvelle primitive interpolante est créée qui ne transporte pas la masse orpheline vers une primitive de l’autre image clef, mais vers une primitive particulière appelée primitive fantôme. La particularité de ces primitives est qu’elles créent un effet de disparition ou de disparition des masses orphelines par fondu progressif.

### 4.3.2 Extrapolation et primitives fantômes

Les primitives fantômes sont appelées ainsi car elles sont de densité et donc de masse nulle et qu’elles ne sont pas issues des images clefs mais permettent de faire apparaître ou disparaître les portions orphelines des primitives. Conceptuellement, les primitives fantômes sont les projections des portions orphelines des primitives dans l’image clef opposée, permettant de réintroduire les masses orphelines dans le modèle interpolant qui ne considère que des couples de primitives interpolées le long d’une trajectoire donnée.

Pour chaque primitive initiale  $A_i$  ou finale  $B_j$  dont une portion de masse  $m_{i \rightarrow}$  ou  $m_{\leftarrow j}$  est orpheline, la primitive fantôme associée est notée  $\bar{A}_i$  ou  $\bar{B}_j$ . Contrairement aux plus courts chemins calculés entre tous les couples de primitives initiales et finales, les trajectoires suivies par les masses orphelines n’ont pas à relier deux positions imposées par les images

clefs. Elles sont extrapolées en suivant au mieux les conditions de l'environnement nuageux sur la même grille que celle utilisée pour la recherche de plus court chemin mais en réalisant simplement à chaque pas de temps le déplacement le moins coûteux. C'est une simulation très simplifiée d'un système de particules indépendantes qui suivent le principe de moindre effort.

Une autre différence avec la recherche de plus court chemin est qu'il y a deux cas de figure : le cas des masses orphelines qui sont issues de primitives initiales où l'extrapolation se fait en avant dans le temps, et le cas où elles sont issues de primitives finales où l'extrapolation se fait alors en remontant le temps. Les trajectoires sont notées  $\Gamma_{i \rightarrow}$  pour l'extrapolation en avant et  $\Gamma_{\leftarrow j}$  pour l'extrapolation en arrière.

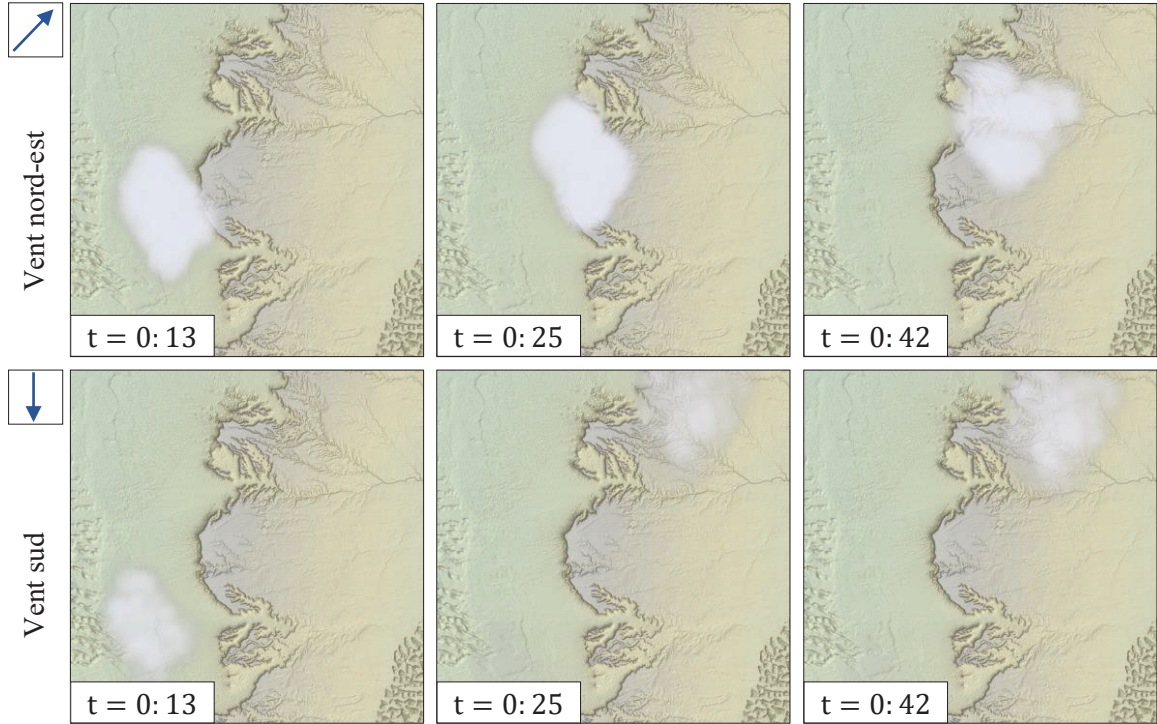
### 4.3.3 Cohérence et choix de la valeur de seuil

La création des primitives fantômes permet de relaxer le processus de métamorphose par transport optimal en retirant du transport les masses des primitives interpolantes dont les trajectoires sont trop irréalistes. L'effet de fondu progressif qui en résulte renforce la cohérence de l'animation vis-à-vis de l'environnement nuageux tout en maintenant le respect des images clefs par l'animation. De plus, la disparition et la réapparition soudaines de nuages n'est pas qu'un outil d'animation, c'est un phénomène qui se produit souvent dans la nature en fonction des variations des conditions du point de rosée le long de la trajectoire que les vents impriment aux nuages.

En fin de compte, le choix de la valeur de seuil  $\kappa_{\max}$  revient alors à l'utilisateur. Une valeur de seuil nulle consiste à donner la priorité à la cohérence de l'animation par rapport à l'environnement nuageux. Cela correspond à une totale intolérance de la métamorphose vis-à-vis des coûts des déplacements. La plupart pour ne pas dire toutes les masses se révèlent alors orphelines et suivent la simulation simplifiée, sauf dans le cas improbable où deux primitives sont parfaitement en cohérence et associées par un plus court chemin de coût nul. Au contraire, une valeur de seuil infinie donne la priorité aux images clefs et à la métamorphose. Il n'y a aucune création de primitives fantômes et toutes les primitives interpolantes transportent les masses entre les primitives des images clefs, même si toutes les trajectoires sont aberrantes.

## 4.4 Primitives interpolantes

Les images clefs  $A$  et  $B$  sont composées de primitives de contrôle statiques ayant chacune pour paramètres la position de son centre  $C$ , son rayon horizontal  $R_H$ , son rayon vertical  $R_V$  et sa densité maximale  $D$ . Les mélanges des primitives initiales et finales constituent les états de la fonction de présence d'un type de nuage donné (section 3.2 page 92) aux deux instants associés  $t_A$  et  $t_B$ . La métamorphose entre les deux images clefs est exprimée par un ensemble de primitives interpolantes  $C$  dont le mélange est équivalent aux images clefs aux instants associés.



**FIGURE 4.18.** – Sous des conditions favorables de vent nord-est (haut), aucun effet fantôme n’arrive puisque les nuages peuvent être transportés par des trajectoires admissibles. Sous des conditions de vent opposé (bas), des trajectoires sont éliminées et des fantômes sont créés pour chaque primitive concernée.

Pour chaque relation de correspondance entre une primitive initiale  $A_i$  et une primitive finale  $B_j$ , une primitive interpolante  $C_{i,j}$  transporte une portion de masse nuageuse le long du plus court chemin associé. De manière similaire, chaque portion de primitive rendue orpheline donne lieu à l’instanciation d’une primitive interpolante qui réalise un effet de fondu progressif. Pour une primitive fantôme  $\bar{A}_i$  associée à une primitive initiale  $A_i$ , la primitive interpolante  $C_{i \rightarrow}$  fait disparaître la masse orpheline jusqu’à la primitive fantôme. Pour une primitive fantôme  $\bar{B}_j$  associée à une primitive finale  $B_j$ , la primitive interpolante  $C_{\leftarrow j}$  fait apparaître la masse orpheline depuis la primitive fantôme.

#### 4.4.1 Interpolation entre primitives statiques

Soit un couple  $(A_i, B_j)$  composé d’une primitive initiale et d’une primitive finale dont la relation de mise en correspondance est pondérée par  $w_{i \rightarrow j}$  et  $w_{i \leftarrow j}$ , et  $\tilde{t} = \frac{t-t_A}{t_B-t_A}$  le paramètre temporel normalisé ; la primitive interpolante est définie par interpolation des paramètres initiaux et finaux.

Le centre de la primitive interpolante suit le plus court chemin entre les primitives statistiques

$$C_{i,j}(\tilde{t}) = \Gamma^*_{i,j}(\tilde{t}). \quad (4.38)$$

Les rayons peuvent tous être directement interpolés

$$\mathbf{R}_{i,j}(\tilde{t}) = \mathcal{I}(\mathbf{R}_{A_i}, \mathbf{R}_{B_j}, \tilde{t}), \quad (4.39)$$

mais il est aussi envisageable d'interpoler les volumes en premier lieu puis d'en déduire les rayons. Quant à la densité maximale, elle est déduite de l'interpolation de la masse dont les pondérations de correspondance dépendent

$$D_{i,j}(\tilde{t}) = M^{-1}(\mathcal{I}(m_{A_i} \cdot w_{i \rightarrow j}, m_{B_j} \cdot w_{i \leftarrow j}, \tilde{t}), \mathbf{R}_{i,j}(\tilde{t})). \quad (4.40)$$

#### 4.4.2 Interpolation entre primitive statique et primitive fantôme

En appliquant le seuillage du coût des trajectoires, un sous-ensemble  $\bar{L}$  de liens trop coûteux et les primitives interpolantes correspondantes sont invalidés. Les masses orphelines sont réintroduites dans la métamorphose en ajoutant des primitives interpolantes liant les portions orphelines des primitives à leurs projections extrapolées, les primitives fantômes. Pour ces primitives interpolantes, les rayons sont constants

$$\mathbf{R}_{i \rightarrow}(\tilde{t}) = \mathbf{R}_{A_i} \text{ et,} \quad (4.41)$$

$$\mathbf{R}_{\leftarrow j}(\tilde{t}) = \mathbf{R}_{B_j}. \quad (4.42)$$

Il y a ensuite deux cas de figure suivant que la portion orpheline est initiale ou finale. Dans le premier cas, le centre suit la trajectoire extrapolée en avant et la portion orpheline disparaît progressivement alors que dans le second cas, le centre suit la trajectoire extrapolée en arrière et la portion orpheline apparaît progressivement :

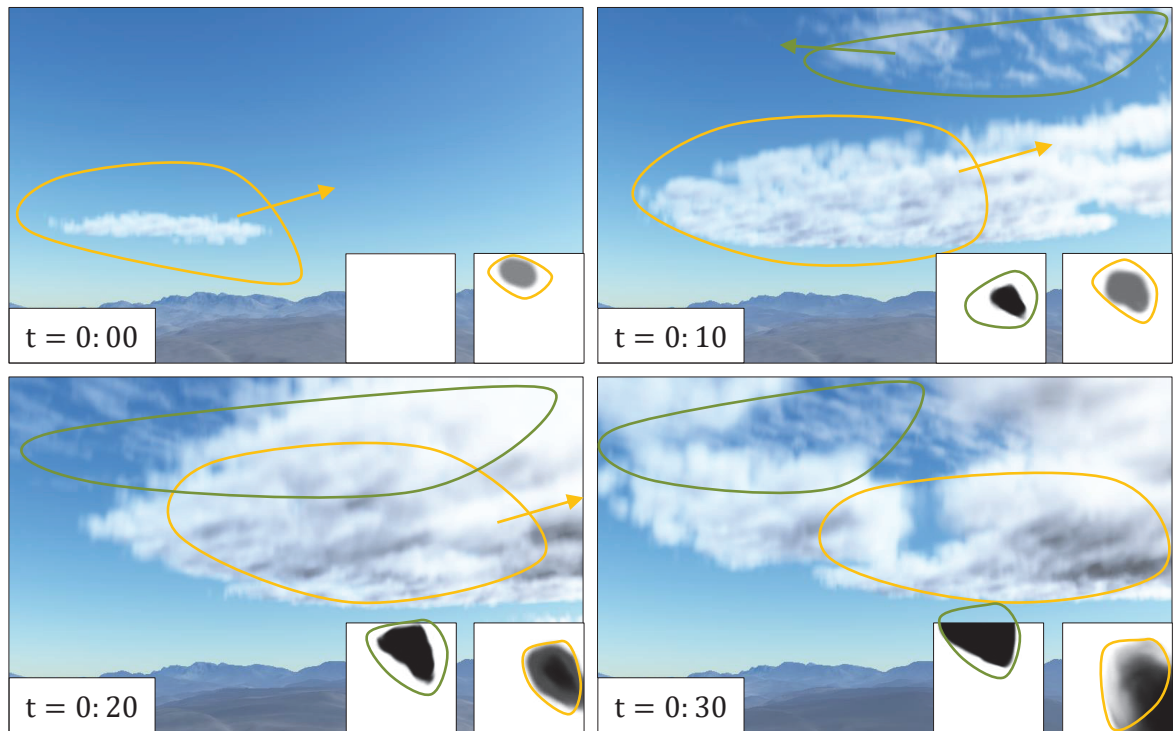
$$\mathbf{C}_{i \rightarrow}(\tilde{t}) = \mathbf{\Gamma}_{i \rightarrow}(\tilde{t}), \quad (4.43)$$

$$\mathbf{C}_{\leftarrow j}(\tilde{t}) = \mathbf{\Gamma}_{\leftarrow j}(\tilde{t}), \quad (4.44)$$

$$D_{i \rightarrow}(\tilde{t}) = \mathcal{I}\left(D_{A_i} \cdot \frac{m_{i \rightarrow}}{m_{A_i}}, 0, \tilde{t}\right) \text{ et} \quad (4.45)$$

$$D_{\leftarrow j}(\tilde{t}) = \mathcal{I}\left(0, D_{B_j} \cdot \frac{m_{\leftarrow j}}{m_{B_j}}, \tilde{t}\right). \quad (4.46)$$





**FIGURE 4.19.** – Exemple de vents croisés avec deux types de nuages (111 primitives de contrôle). Les *Stratus* se forment à basse altitude et suivent le vent qui souffle d’ouest en est. Des *Cirrus* apparaissent à haute altitude et dérivent du nord-est au sud-ouest selon une direction de vent indépendante. Les nuages commencent à couvrir le ciel par le truchement de leur champ de vent propre et d’après les images clefs définies par l’utilisateur.

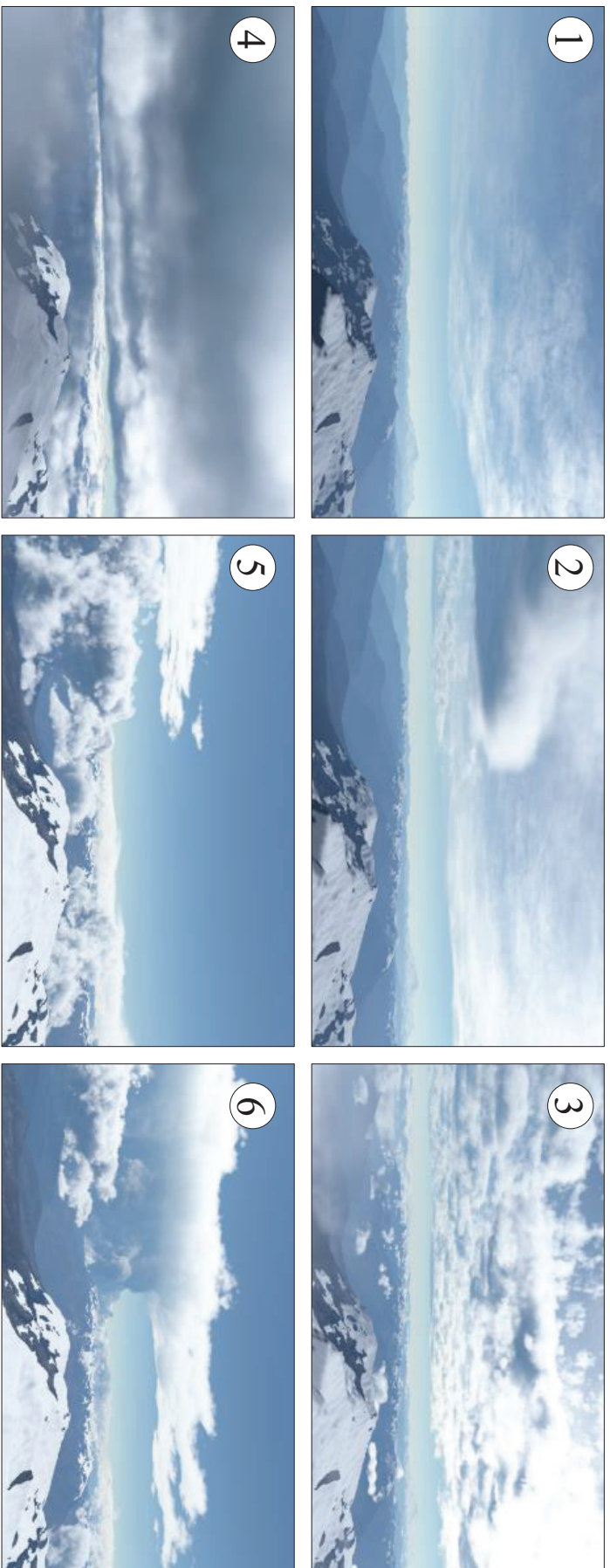
## 4.5 Résultats

Notre méthode a été implémentée et testée sur un processeur Intel Core i7-6700K avec 16 GB de *RAM* et une carte graphique Nvidia GTX 970. Les paysages nuageux montrés à travers ce manuscrit ont été générés avec notre algorithme. Le modèle de paysage nuageux et les fonctions de contrôle ont été implémentés sous forme de *compute-shaders GLSL*, et les différents paysages nuageux ont été rendus sur *GPU* par l’une de nos deux implémentations (annexe A page 135) : soit par *ray-marching* avec l’approximation de *single-scattering* soit par échantillonnage de MONTE CARLO.

### 4.5.1 Contrôle

Notre méthode permet de créer différentes animations de paysages nuageux. Le section 4.5.1 page 127 répertorie les statistiques de différents scénarios. Notez que l’algorithme de transport optimal garantit la stabilité des animations par rapport à l’opération d’insertion d’image clef.

Notre méthode permet d’appliquer différentes contraintes de vent pour chaque couche de nuages. La figure 4.19 montre des *Stratus* à altitude moyenne et des *Cirrus* à haute



**Air froid** Direction principale du vent **Air chaud**

**FIGURE 4.20.** – Ce scénario météorologique de front froid a été créé en métamorphosant une grande variété de types de nuages se déplaçant d’après la même direction de vent d’ouest en est. Le champ de contrôle de chaque couche a été défini par une primitive suivant la même direction de vent. 1. Le front froid commence avec des *Cirrus* suivis de *Cirrostratus*. 2. Arrivée de *Altostratus*. 3. Arrivée de *Altostratus* et *Cumulus humilis*. 4. Les nuages de moyenne altitude descendent et les *Stratocumulus* recouvrent le paysage. 5. Au cœur du front, les nuages stratiformes sont remplacés par des nuages convectifs ; *Cumulus congestus*. 6. Le *Cumulonimbus capillatus* finit par apparaître, mettant l’accent sur l’instabilité du front.

**TABLE 4.1.** – Statistiques de plusieurs animations : nombre de couches nuageuses, nombre de primitives de présence, dimensions du paysage, et durée du scénario.

Figure	Couches	Primitives	Dimensions	Durée
<i>Cross-winds</i> (4.19)	2	111	56 × 56 km <sup>2</sup>	0.5 h
Front froid (4.20)	8	8	588 × 120 km <sup>2</sup>	10 h
<i>Fog-rise</i> (4.22)	2	465	126 × 126 km <sup>2</sup>	2 h

altitude transportés par des vents soufflant dans des directions et à des vitesses différentes pour chaque intervalle d'altitudes.

La figure 4.20 page précédente montre une animation de front froid au-dessus des Alpes françaises. Un front froid est le bord d'attaque d'une masse d'air froide qui passe par-dessus une masse d'air plus chaude au niveau du sol. Le scénario de 10 h a été facilement créé en utilisant notre système et en métamorphosant des images clef de couvertures nuageuses pour chaque type de nuage, subissant le même vent orienté d'ouest en est. Le scénario de front froid implique huit types de nuages différents tels que des *Cumulus*, *Cumulus humilis* ou *Stratocumulus*.

La Figure 4.22 page 129 montre une chaîne de montagnes avec de la brume se dissipant depuis les vallées et des *Cumulus* se formant à plus haute altitude. Le vent associé à la couche de brume a été défini comme soufflant faiblement en direction ascendante, tandis que le vent de la couche de *Cumulus* souffle horizontalement et plus fort. Deux images clef ont été utilisées pour définir la métamorphose du brouillard. Dans la première image clef, des primitives ont été placées au fond des vallées, tandis que la seconde image clef a été définie comme vide, entraînant une disparition automatique grâce à la création de primitives fantômes.

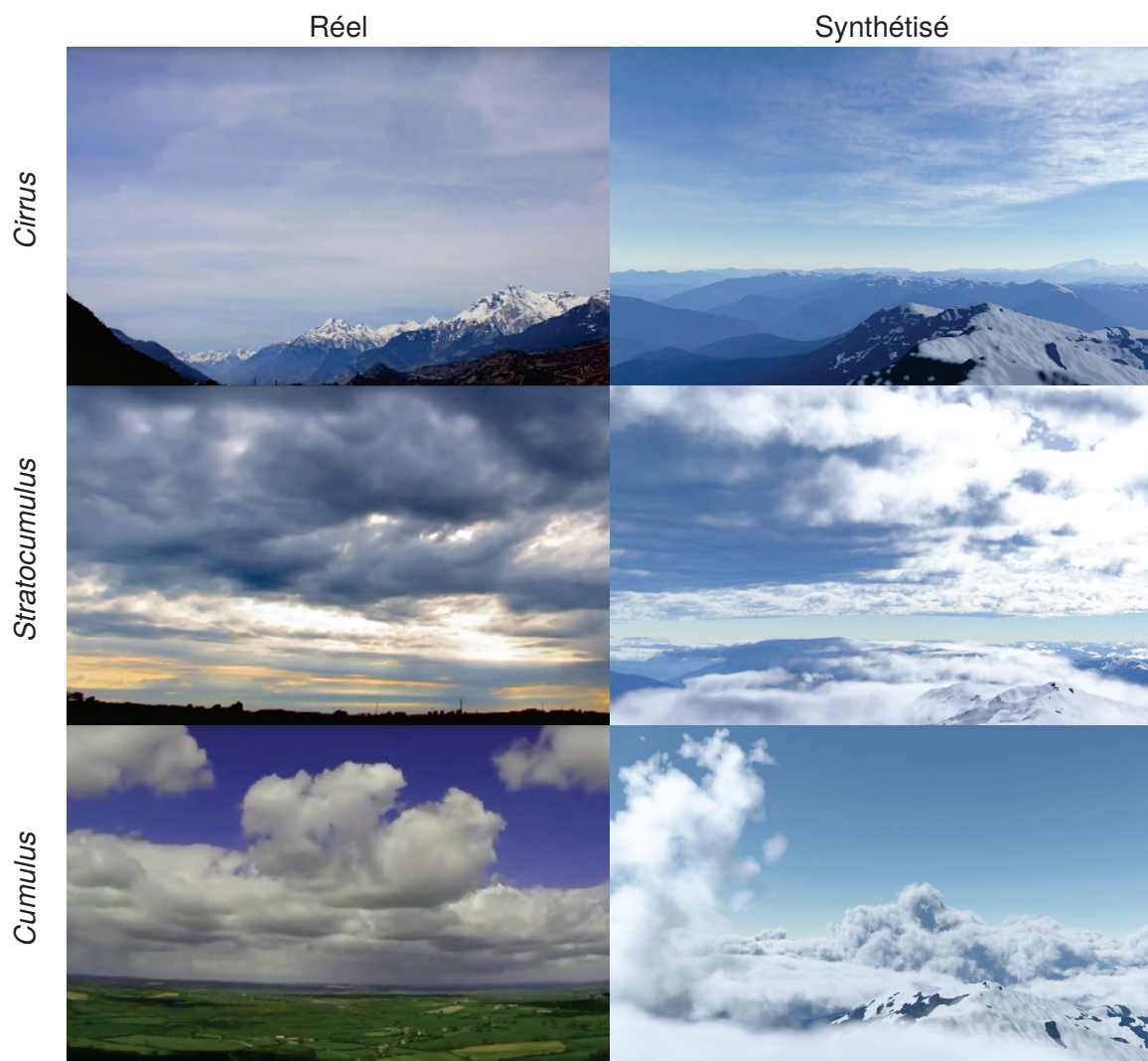
## 4.5.2 Performances

Le processus complet de métamorphose qui englobe le calcul des plus courts chemins et du transport optimal s'exécute en quelques secondes. Cette étape ne nécessite d'être calculée qu'une fois pour toutes pour chaque paire d'images clef. Le nombre de primitives pour les paysages nuageux *cross-winds* (Figure 4.19 page 125) et *fog-rise* (Figure 4.22 page 129) est 110 et 460 respectivement.

Nous avons implémenté le modèle de paysage nuageux et les fonctions de contrôle sous forme de *shaders* sur le *GPU*. Les différents paysages nuageux ont été rendus par un algorithme basé *GPU* de *ray-marching* avec l'approximation de *single-scattering*. Nos fonctions de nuages procéduraux avec expression analytique sont coûteuses en calculs car elles nécessitent de nombreuses évaluations de fonctions de bruit pour chaque type de nuage. Le rendu d'une seule image à la résolution 1280 × 720 a pris environ 10s.

Bien que la visualisation soit trop lente pour être utilisée dans des applications en temps réel, à notre connaissance, notre méthode est la première qui permet de générer des modèles analytiques pour des types de nuages variés et de contrôler les paysages nuageux de manière cohérente. Simplifier et accélérer le rendu pour des applications en temps réel en optimisant





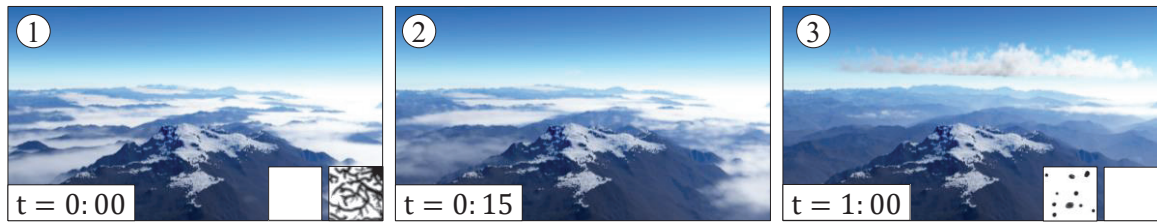
**FIGURE 4.21.** – Images extraites des vidéos utilisées dans l’expérience utilisateur (vidéos réelles : <https://youtu.be/vk-s9-qZ0bU>, <https://youtu.be/MtcxfjvwV1s>, et <https://youtu.be/Uhvlk2ZyIkg>).

la représentation hiérarchique des différentes primitives de nuage au sein de leur couche associée est un important projet de recherche en cours, en dehors du champ d’application de ce document.

### 4.5.3 Validation

Nous avons présenté les résultats à des experts météorologues (deux chercheurs et un professeur de prévisions météo), un pilote et un expert en logiciels d’environnement virtuel. Tous ont apprécié la qualité et le réalisme de l’animation et du rendu.

Nous avons mené une petite expérience utilisateur afin de comparer la plausibilité de nos animations de paysages nuageux à la réalité. Il était demandé aux participants de regarder une animation réelle de nuages d’un type donné, puis de visionner une animation de synthèse produite par notre méthode. La figure 4.21 montre quelques images extraites des vidéos de *Cirrus* et *Altostratus*. Nous avons demandé aux participants de noter la



**FIGURE 4.22.** – Exemple d'effets fantômes : le brouillard composé de *Stratus* qui couvrent les vallées (gauche) s'élève progressivement et disparaît au fur et à mesure que la température augmente, et qu'il est balayé par de petites brises (centre); lorsque le brouillard a presque entièrement disparu, des *Cumulus* commencent à se former à plus haute altitude. (465 primitives)

plausibilité de notre méthode par rapport à l'exemple réel sur une échelle de Likert à cinq niveaux. Nous avons 15 participants âgés de 21 à 54 ans. Certains d'entre eux avaient une formation significative en informatique graphique, ou étaient habitués à observer les nuages. Nous avons réalisé le test sur trois types de nuages : des *Cirrus*, des *Altostratus* et des *Cumulus*.

D'après l'hypothèse nulle des utilisateurs donnant autant de notes positives que négatives, la *p-value* obtenue par un test *chi-squared* était respectivement de moins de 0.16, 0.011 et 0.034 pour chaque type de nuage (valeurs  $\chi^2$  de 2, 6.5 et 4.5 respectivement).

L'hypothèse nulle peut ainsi être éliminée avec une relative certitude pour les *Altostratus* et *Cumulus*, démontrant l'efficacité de notre modèle procédural. Au contraire, les *Cirrus* n'ont pas été perçus comme suffisamment convaincants. Une raison possible pour cela est que les directions de vent étaient opposées dans les animations réelles et de synthèse, ce qui a en effet été constaté par certains participants à la fin de l'expérience.

## 4.6 Conclusion

Notre méthode se compare avantageusement aux systèmes d'animation de nuages basés primitives d'après plusieurs critères. Les systèmes traditionnels à images clef et à particules [130] nécessitent une édition fastidieuse des primitives ellipsoïdales qui constituent les nuages, et passent mal à l'échelle pour la création de grandes scènes. À l'inverse, notre approche permet de manipuler de grands paysages nuageux facilement.

De plus, notre stratégie fondée sur le transport optimal, utilisée pour associer les primitives composant de grandes couvertures nuageuses permet de synthétiser des animations de paysages nuageux cohérentes avec le champ de vent et le terrain. Les algorithmes de métamorphose [20, 72, 89, 90] qui ont été proposés antérieurement pour animer des nuages basés primitives génèrent des trajectoires qui interpolent linéairement les positions des primitives initiales et finales, ce qui produit des trajectoires irréalistes. De plus, les processus d'association fondés sur des heuristiques génèrent souvent de nombreuses primitives, en général  $O(n_A \times n_B)$ , ce qui les rend inadaptés à la métamorphose de grands paysages nuageux.



À l'inverse, notre approche de métamorphose, fondée sur la combinaison du transport optimal et du plus court chemin anisotrope, résout ces problèmes en générant des trajectoires cohérentes et en limitant le nombre de primitives générées à  $O(n_A + n_B)$ . En outre, la fonction de coût permet à l'utilisateur d'influencer les trajectoires des nuages. Une contribution importante de notre méthode est le calcul de trajectoire qui prend en compte les paramètres environnementaux tels que le champ vectoriel de vent et la forme du terrain. Ajouté à cela, nous proposons pour la première fois un modèle général qui incorpore tous les différents types de nuages d'une manière procédurale et contrôlable.

**Limitations** Notre approche ne se prête pas aux phénomènes météorologiques extrêmes tels que les cyclones ou les tornades. Dans ces cas particuliers, le champ vectoriel devrait être défini à de hautes résolutions, ce qui n'est présentement pas possible dans notre pipeline de modélisation.

Une autre limitation de notre méthode est que la métamorphose est limitée à un seul type de nuage, *id est* nous ne pouvons pas transformer un nuage d'un type en un nuage d'un autre type avec notre processus de métamorphose. Parce que nous contrôlons la couverture nuageuse avec de grandes primitives, certains des nuages générés n'interagissent pas correctement avec les montagnes au-dessus desquelles ils se déplacent. Une solution consisterait à utiliser un plus grand nombre de primitives de plus petite taille, au prix de calculs plus coûteux.

Pour finir, parce que nous mélangeons les densités des couches nuageuses au sein d'un seul champ de densité, nous ne pouvons pas modéliser différentes propriétés de diffusion de la lumière pour des nuages de types différents.

# Conclusion et perspectives

Nous avons proposé une méthode de génération procédurale de paysages nuageux animés. Notre modèle représente le champ de densité de ces paysages animés par une hiérarchie de fonctions. À la racine, les différents types nuageux sont tout d'abord séparés en couches indépendantes. Au second niveau, chacune de ces couches est séparée à son tour en deux sous-fonctions : une fonction contrôlant la présence nuageuse à grande échelle, et une fonction représentant les détails nuageux caractéristiques de la couche. La jonction entre fonction de présence et fonction de détails est obtenue par application d'un opérateur de rétrécissement–estompage qui permet de configurer une dynamique d'apparition et de disparition particulière à chaque type de nuage. Au dernier niveau, les fonctions de couche sont définies comme les mélanges lisses de primitives implicites à squelette obtenues à partir de cartes grossières fournies par l'utilisateur et animées par application de notre méthode de métamorphose. Les fonctions de détails sont quant à elles exprimées sous la forme de combinaisons de bruits procéduraux, pré-sélectionnées afin de reproduire l'apparence spécifique de différents types de nuage. Les motifs procéduraux présentent une animation turbulente propre qui dépend de leurs dimensions, mais ils sont aussi mus par les vents de leur couche nuageuse. Afin de visualiser notre modèle procédural, nous avons tout d'abord implémenté une méthode de rendu atmosphérique par marche de rayons diffusés seulement une fois entre le soleil et le terrain, les nuages ou le reste de l'atmosphère. Nous avons ensuite implémenté un lancer de rayon plus générique par échantillonnage de MONTE CARLO de l'espace des chemins de longueur quelconque.

Nous avons proposé d'animer nos paysages nuageux à grande échelle en interpolant leur composition à différents instants clefs, tout en prenant en compte l'influence du relief surplombé par les nuages et les vents qui les déplacent. L'utilisateur fournit des cartes des différentes couvertures nuageuses qui sont discrétisées en primitives implicites statiques. La transition ou métamorphose entre les instants clefs des cartes est obtenue en construisant un ensemble de primitives interpolant spatiotemporellement les primitives statiques subséquentes d'une couche. La première étape consiste à calculer le chemin qui va le moins à l'encontre des vents et du relief entre chaque couple de primitives statiques. La sélection d'un bon sous-ensemble suffisant parmi ces chemins constitue un problème de mise en correspondance des primitives subséquentes que nous proposons de résoudre sous la forme du transport optimal de la masse nuageuse d'après le coût des différents chemins. Puisqu'il arrive que certains chemins optimaux s'opposent encore trop au relief et aux vents ou au but de l'utilisateur, nous avons proposé de remplacer le transport de la masse nuageuse le long des chemins fautifs par des effets d'apparition–disparitions progressifs de la masse nuageuse en suivant les trajectoires qui adhèrent au mieux au relief et aux vents. La métamorphose des états du paysage nuageux ainsi obtenue est contrôlable par l'utilisateur et cohérente avec les conditions de l'environnement nuageux.

Au travers de cette thèse, nous affirmons apporter un certain nombre de contributions. Bien que de nombreuses méthodes emploient du bruit procédural pour produire des détails nuageux fins, notre sélection et la manière de les combiner pour reproduire les motifs parti-

culiers de différents types de nuage est inédite et a nécessité de nombreux essais et erreurs. Nous n'avons pas non plus connaissance d'autres méthodes ayant défini une dynamique d'apparition–disparition spécifique à chaque type de nuage tel que nous le permet notre opérateur d'estompage–rétrécissement. Concernant l'édition d'animation, notre méthode est la première à appliquer les outils pourtant classiques et intuitifs des images clefs et de la métamorphose au cas des nuages. De plus, aucune autre méthode d'animation procédurale de nuages n'apporte de cohérence à l'animation obtenue comme nous le faisons par l'intermédiaire de notre recherche de plus court chemin spatiotemporel et la prise en compte du relief et des vents. Dans le domaine de la métamorphose, nous avons proposé une nouvelle heuristique de mise en correspondance automatique par transport optimal qui limite efficacement la taille du graphe de correspondance tout en assurant que les relations soient symétriques.

En comparaison des méthodes cherchant à simuler les lois de la thermodynamique qui régissent la formation de nuages, notre méthode partage l'avantage des autres méthodes procédurales : elle est bien moins coûteuse et beaucoup plus contrôlable. En effet, dans le cas des simulations, l'évaluation de la densité nuageuse en un point et à un moment donné nécessite de calculer l'évolution de toute l'étendue nuageuse depuis l'état initial et ne peut généralement pas être guidée. Au contraire, notre modèle procédural en limite spatialement et temporellement la complexité et permet de contrôler l'état des nuages à des instants donnés et de modéliser des étendues nuageuses bien plus grandes et détaillées. En comparaison des autres méthodes procédurales, notre processus de métamorphose permet de produire une animation contrôlable de nuages et néanmoins cohérente. En effet, peu de méthodes procédurales s'intéressent à l'édition de nuages animés et celles qui le font proposent principalement de définir l'animation manuellement, sans même s'intéresser à sa cohérence.

**Tests d'utilisation par des artistes pour validation du potentiel créatif** L'un des objectifs qui a motivé l'élaboration de notre méthode d'édition de paysage nuageux animé était de fournir plus de contrôle que les méthodes existantes, pour faciliter l'expression du pouvoir créatif des artistes de l'industrie graphique. Il serait intéressant d'impliquer de tels artistes, de leur fournir un prototype de test et de recueillir leurs avis afin de vérifier si l'objectif est atteint, c'est-à-dire si notre méthode présente un réel intérêt à leurs yeux.

**Métamorphose entre différents types de nuages** La technique d'animation que nous avons proposée permet d'interpoler la présence nuageuse entre deux instants d'un même type de nuage, mais pas entre deux types de nuages différents. Il est bien entendu possible de fournir des images clefs qui correspondent implicitement à un transfert de présence entre différents types de nuage (c'est le cas de la figure 4.22), mais ce n'est pas automatique. Une extension de notre méthode consisterait ainsi de généraliser la métamorphose entre les différents types de nuages, c'est-à-dire mettre en correspondance et interpoler les primitives implicites de toutes les couches à la fois.

**Advection des détails et trajectoires des primitives interpolante** Notre méthode sépare la forme des nuages à grande échelle, qui est éditée par l'animateur, des détails caractéristiques des différents types de nuages, qui sont générés et animés procéduralement. L'intérêt est de fournir un contrôle efficace à l'utilisateur et de le libérer de l'édition pénible des détails nuageux. Cela entraîne cependant un problème d'incohérence entre les trajectoires des primitives interpolantes et l'animation des détails. En effet, une partie de l'animation des

détails est obtenue par translation du domaine selon le vecteur de vent principal, tandis que les primitives interpolantes suivent leurs propres trajectoires en s'écartant plus ou moins des conditions de vent, et donc du vecteur de vent principal. Une perspective serait de vérifier s'il est possible de prendre en compte les trajectoires des primitives interpolantes pour influencer l'advection des détails nuageux et assurer la cohérence entre advection des détails et métamorphose à grande échelle.

**Échelle planétaire et niveau de détail** Au début de cette thèse, nous souhaitons générer procéduralement l'ensemble de la couverture nuageuse planétaire depuis l'échelle macroscopique que nous connaissons tous à la surface du sol, voire en avion, jusqu'à l'échelle planétaire dont nous avons quelques aperçus à travers la conquête spatiale. Traiter l'échelle macroscopique s'est cependant révélé plus complexe que prévu, notamment à cause du coût du rendu atmosphérique. Afin de permettre le passage à l'échelle planétaire, tant du point de vue des performances que de la cohérence entre les échelles, nous suggérons d'étudier les propriétés statistiques et spectrales des nuages modélisés à différents niveaux de détail.

**Liens avec les modèles et données météorologiques** Nous avons proposé de générer les détails nuageux par génération procédurale *ex nihilo* en composant des octaves de bruit procédural. Nous avons choisi les types de bruits utilisés pour chaque type de nuage modélisé par essais et erreurs, en s'inspirant de références photographiques. L'analyse de modèles et relevés météorologiques permettrait de multiplier les points de vue de référence ainsi que de calculer les statistiques des nuages réels pour les comparer à ceux que nous avons modélisés. Il serait aussi possible d'utiliser les données météorologiques étiquetées pour les différents types de nuage comme atlas pour proposer une méthode de modélisation par l'exemple.





# Rendu atmosphérique

Nous rappelons dans cette section les fondements du rendu en milieu participatif, et en détaillons notre implémentation dans le cas du rendu atmosphérique. Pour plus de détails, le lecteur est invité à consulter l'état de l'art sur les méthodes de MONTE CARLO pour la simulation du transport volumique de la lumière présenté à Eurographics 2018 [113], ainsi que les articles de KUTZ *et al.* [82] et NOVÁK *et al.* [112] précédemment évoqués dans notre propre état de l'art (section 1.2 page 28), dont nous nous inspirons.

## A.1 Milieu participatif

Les nuages et le reste de l'atmosphère constituent un milieu participatif pour le transfert radiatif de la lumière. Les gouttelettes d'eau condensée ou de cristaux de glace des nuages, ainsi que les molécules des différents gaz de l'atmosphère influencent le trajet de la lumière. Il existe trois types d'interactions possibles dans un milieu participatif : l'émission de lumière, l'absorption de lumière, et sa déviation aussi appelée diffusion. La diffusion est généralement séparée en deux parties : la diffusion entrante (*inscattering*) qui collecte de la lumière depuis l'ensemble des directions et la redirige dans la direction incidente, et la diffusion sortante (*outscattering*) qui, au contraire, redistribue la lumière depuis la direction incidente vers l'ensemble des directions.

## A.2 Densité et probabilité d'interaction

De manière générale, lorsque la lumière traverse un volume participatif, la composition, la distribution, la forme et la taille des particules du milieu sont des facteurs qui déterminent l'influence des différents types d'interactions. Comme il n'est pas envisageable de représenter explicitement l'ensemble des particules et leurs formes, le problème est traité dans le domaine probabiliste en ignorant les interactions entre les particules elles-mêmes. Soit  $\rho$  la densité en particules par unité de volume ( $\text{m}^3$ ) et  $\theta$  la section transversale d'une particule ( $\text{m}^2$ ) ; la probabilité qu'une interaction se produise par une unité de distance ( $\text{m}$ ) s'exprime comme leur produit

$$\mu(\mathbf{p}) = \theta \cdot \rho(\mathbf{p}). \quad (\text{A.1})$$

Les probabilités d'absorption  $\mu_a$  et de diffusion  $\mu_s$  d'un photon par unité de distance parcourue sont ainsi exprimées en fonction des sections transversales associées  $\theta_a$  et  $\theta_s$ . La probabilité d'atténuation  $\mu_t = \mu_a + \mu_s$  combine les probabilité d'absorption et de diffusion.

**Définition 14 (Albédo)** *Le rapport entre l'énergie lumineuse diffusée et l'énergie lumineuse incidente est appelé l'albédo. Un albédo nul correspond au cas limite du corps noir qui absorbe toute la lumière. Au contraire, les nuages et l'air ont un albédo élevé car ils diffusent fortement la lumière mais n'en absorbent qu'une faible portion.*

## A.3 Diffusion et fonction de phase

**Définition 15 (Fonction de phase)** *Lorsque la lumière est diffusée, elle n'est pas nécessairement diffusée uniformément dans toutes les directions, id est de manière isotrope. La distribution directionnelle de l'énergie lumineuse après diffusion est appelée fonction de phase et notée  $f$ . Lorsque les particules sont orientées, la fonction de phase dépend de la direction incidente et de la direction de diffusion. Dans le cas contraire, la symétrie de la fonction de phase permet de l'exprimer en fonction de l'angle de déviation.*

Bien que ce ne soit pas toujours le cas dans la littérature, nous normalisons l'expression de la fonction de phase par son intégrale sur la sphère des directions. Ainsi, la fonction de phase isotrope est constante et vaut  $\frac{1}{4\pi}$  dans toutes les directions. Dans le cas général, la fonction de phase est plutôt anisotrope et dépend de la taille et/ou de la forme des particules présentes dans le milieu participatif.

Suivant la taille des particules relativement aux longueurs d'onde considérées pour la diffusion, les caractères ondulatoires et magnétiques de la lumière entrent en jeu et complexifient l'élaboration de la fonction de phase. Modéliser la diffusion atmosphérique requiert ainsi des fonctions de phase adaptées notamment aux molécules des gaz atmosphériques et aux gouttelettes d'eau ou cristaux de glace qui composent les nuages.

### A.3.1 Diffusion de RAYLEIGH

Lorsque les particules diffusant la lumière sont petites par rapport à la longueur d'onde, comme les molécules de dioxygène par rapport à la lumière visible, c'est plutôt le modèle de diffusion de RAYLEIGH qui est utilisé.

Pour la diffusion de la lumière par l'air, nous utilisons la fonction de phase donnée par SLOUP [132]

$$f_g(\theta) = \frac{3}{2} \frac{1}{2 + \delta} [(1 + \delta) + (1 - \delta) \cos^2 \theta] \quad (\text{A.2})$$

où  $\delta = 0.035$  est un facteur de dépolarisation.

### A.3.2 Diffusion de MIE

Lorsque la taille des particules est de l'ordre de la longueur d'onde, il est bien plus complexe d'élaborer une fonction de phase. Selon la forme et la répartition des particules considérées ; il n'y a la plupart du temps pas de solution analytique exempte d'approximation.

Dans le cas de la diffusion de la lumière par des gouttelettes d'eau composant un nuage, il est courant d'appliquer la solution de MIE, mais cela signifie cependant de faire l'approximation que toutes les gouttelettes d'eau sont de forme sphérique. Considérant les cristaux de glace qui composent les nuages de haute altitude tels que les *Cirrus*, il n'y a pas de modèle analytique de la fonction de phase, il faut par exemple simuler les interactions d'une onde lumineuse avec un cristal de forme donnée ou selon une collection de formes différentes.

## Fonction de phase de HENY-EY-GREENSTEIN

Les fonctions de phase peuvent être très complexes, comme le montre la fonction de phase dans le cas de la théorie de MIE appliquée aux gouttes d'eau d'un nuage. Une approximation efficace et possédant des propriétés pratiques est la fonction de phase de HENY-EY-GREENSTEIN [61] définie en fonction d'un paramètre d'anisotropie  $g \in [-1, 1]$  :

$$f_{HG}(\theta, g) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g \cos \theta)^{3/2}}. \quad (\text{A.3})$$

Une valeur proche de 0 pour le paramètre d'anisotropie  $g$  correspond au cas particulier d'une diffusion anisotrope, pour une valeur proche de  $-1$ , la lumière est principalement renvoyée en direction inverse de la direction d'incidence, tandis que pour une valeur proche de 1, la lumière est peu déviée et continue principalement son chemin.

La fonction de phase de HENY-EY-GREENSTEIN est pratique notamment car elle permet un échantillonnage par importance du cosinus de l'angle de diffusion  $\theta$  :

$$\cos(\theta) = \begin{cases} \frac{1}{2g} \left( 1 + g^2 - \left( \frac{1-g^2}{1-g+2gr} \right)^2 \right) & \text{si } g \neq 0, \\ 2r - 1 & \text{sinon} \end{cases}. \quad (\text{A.4})$$

## A.4 Équation de transfert

L'équation de transfert radiatif de la lumière [18] décrit la variation de luminance (ou radiance) le long d'un rayon orienté dans la direction  $\omega$  et traversant un volume infinitésimal en un point  $\mathbf{p}$  :

$$\nabla_{\omega} L(\mathbf{p}, \omega) = \overset{\text{émission}}{\mu_a L_e(\mathbf{p}, \omega)} - \overset{\text{absorption}}{\mu_a(\mathbf{p}) L(\mathbf{p}, \omega)} - \overset{\text{outscattering}}{\mu_s(\mathbf{p}) L(\mathbf{p}, \omega)} + \overset{\text{inscattering}}{\mu_s(\mathbf{p}) \int_{\Omega} f(\mathbf{p}, -\omega', \omega) \cdot L(\mathbf{p}, -\omega') d\omega'}. \quad (\text{A.5})$$

L'expression de la luminance est obtenue en intégrant l'équation précédente le long de la direction de propagation  $\omega$  :

$$L(\mathbf{p}, \omega) = \int_0^{\infty} T(\mathbf{p}, \mathbf{p} + t\omega) \cdot \left[ \mu_a L_e(\mathbf{p}, \omega) + \mu_s(\mathbf{p}) \int_{\Omega} f(\mathbf{p}, -\omega', \omega) \cdot L(\mathbf{p}, -\omega') d\omega' \right] dt \quad (\text{A.6})$$

où  $T(\mathbf{p}, \mathbf{p} + t\omega)$  est la transmittance.

**Définition 16 (Transmittance)** *La transmittance est le ratio de luminance qui traverse le milieu participatif entre deux points malgré l'absorption et la diffusion sortante. L'expression de la transmittance est*

$$T(\mathbf{p}, \mathbf{p} + t\omega) = e^{-\int_0^t \mu_t(\mathbf{p} + s\omega) ds}. \quad (\text{A.7})$$

#### A.4.1 Transfert atmosphérique multimatériaux et spectral

Dans le cas spécifique des nuages, les particules d'eau n'émettent pas de lumière et nous pouvons donc ignorer le phénomène d'émission. Il en va de même pour l'atmosphère si nous ignorons les phénomènes atmosphériques exotiques tels que les aurores (section 1.1.1 page 21). Dans ce contexte de diffusion atmosphérique, nous ne considérons donc que les phénomènes d'absorption et de diffusion de la lumière.

En revanche, les milieux associés aux gaz atmosphériques et aux particules d'eau des nuages diffèrent notamment par leurs fonctions de phase (annexe A.3 page 136), et leurs probabilités d'absorption et de diffusion respectives (annexe A.2 page 135). Le rendu atmosphérique que nous souhaitons calculer nécessite alors de généraliser l'équation du transfert radiatif (équation A.5 page précédente) au cas d'un milieu multimatériaux. Dans la suite, nous considérons le cas le plus simple où les deux milieux ne s'influencent pas mutuellement, *id est* où leurs propriétés sont indépendantes.

De plus, bien que le milieu nuageux puisse être considéré comme blanc sans trop d'approximation, la probabilité de diffusion par l'air dépend fortement de la longueur d'onde, ce qui lui confère ses couleurs. Pour généraliser l'équation de transfert au cas atmosphérique il faut donc prendre en compte les composantes spectrales de la luminance.

Soit  $\mu_{sg}$  la probabilité spectrale de diffusion des gaz atmosphériques et  $\mu_{sc}$  la probabilité de diffusion des nuages, et  $f_g$  et  $f_c$  leurs fonctions de phase ; l'expression de la luminance atmosphérique est alors

$$\mathbf{L}_{atm}(\mathbf{p}, \omega) = \int_0^\infty \mathbf{T}_{atm}(\mathbf{p}, \mathbf{p} + t\omega) \circ \left[ \mu_{sg}(\mathbf{p}) \circ \int_{\Omega} f_g(\mathbf{p}, -\omega', \omega) \cdot \mathbf{L}_{atm}(\mathbf{p}, -\omega') d\omega' \right. \\ \left. + \mu_{sc}(\mathbf{p}) \circ \int_{\Omega} f_c(\mathbf{p}, -\omega', \omega) \cdot \mathbf{L}_{atm}(\mathbf{p}, -\omega') d\omega' \right] dt \quad (\text{A.8})$$

où  $\circ$  est le produit terme à terme des composantes spectrales et  $\mathbf{T}_{atm}$  est la transmittance spectrale combinée de l'atmosphère qui s'exprime simplement comme le produit des transmittances individuelles des gaz atmosphériques et des nuages

$$\mathbf{T}_{atm}(\mathbf{p}, \mathbf{p}') = \mathbf{T}_g(\mathbf{p}, \mathbf{p}') \cdot T_c(\mathbf{p}, \mathbf{p}'). \quad (\text{A.9})$$

## A.5 Lancer de rayon

Le rendu d'une image de synthèse consiste à évaluer la quantité d'énergie lumineuse perçue par chaque pixel d'un écran virtuel placé dans la scène. Pour estimer l'illumination d'un pixel, un certain nombre de rayons sont lancés entre le pixel  $\mathbf{p}_0$  et les sources lumineuses de la scène, ce qui a été formalisé par VEACH [138] sous la forme de l'intégrale suivante

$$I = \int_{\mathcal{P}} f(\bar{\mathbf{x}}) d\bar{\mathbf{x}} \quad (\text{A.10})$$

où  $\bar{\mathbf{x}} = (\mathbf{p}_0, \dots, \mathbf{p}_k) \in \mathcal{P}$  est un chemin constitué de  $k$  segments avec  $\mathbf{p}_0$  un point du pixel et  $\mathbf{p}_k$  un point d'une source lumineuse, et  $\mathcal{P}$  est l'espace de tous les chemins possibles.

Afin de construire les chemins des rayons, une approche consiste à construire explicitement un nombre limité de chemins qui sont considérés importants et représentatifs des conditions d'éclairage. Dans le cas du rendu atmosphérique, nous pouvons ainsi construire des chemins qui partent des pixels et sont diffusés ou déviés par l'atmosphère ou la surface au plus une fois (*single scattering* [10]) en direction du soleil. La distance avant que chaque rayon ne soit diffusé (*free flight*) peut être discrétisée en pas régulier (*raymarching*), tandis que la transmittance peut être évaluée par quadrature.

Mais si discrétiser la profondeur des événements de diffusion en pas réguliers a l'avantage d'être simple et d'éviter la variance d'un échantillonnage, le rendu produit est biaisé et sensible à l'aliasage de la densité. Une approche plus générique consiste à échantillonner stochastiquement l'espace de tous les chemins possibles par méthode de MONTE CARLO (figure A.1 page suivante).

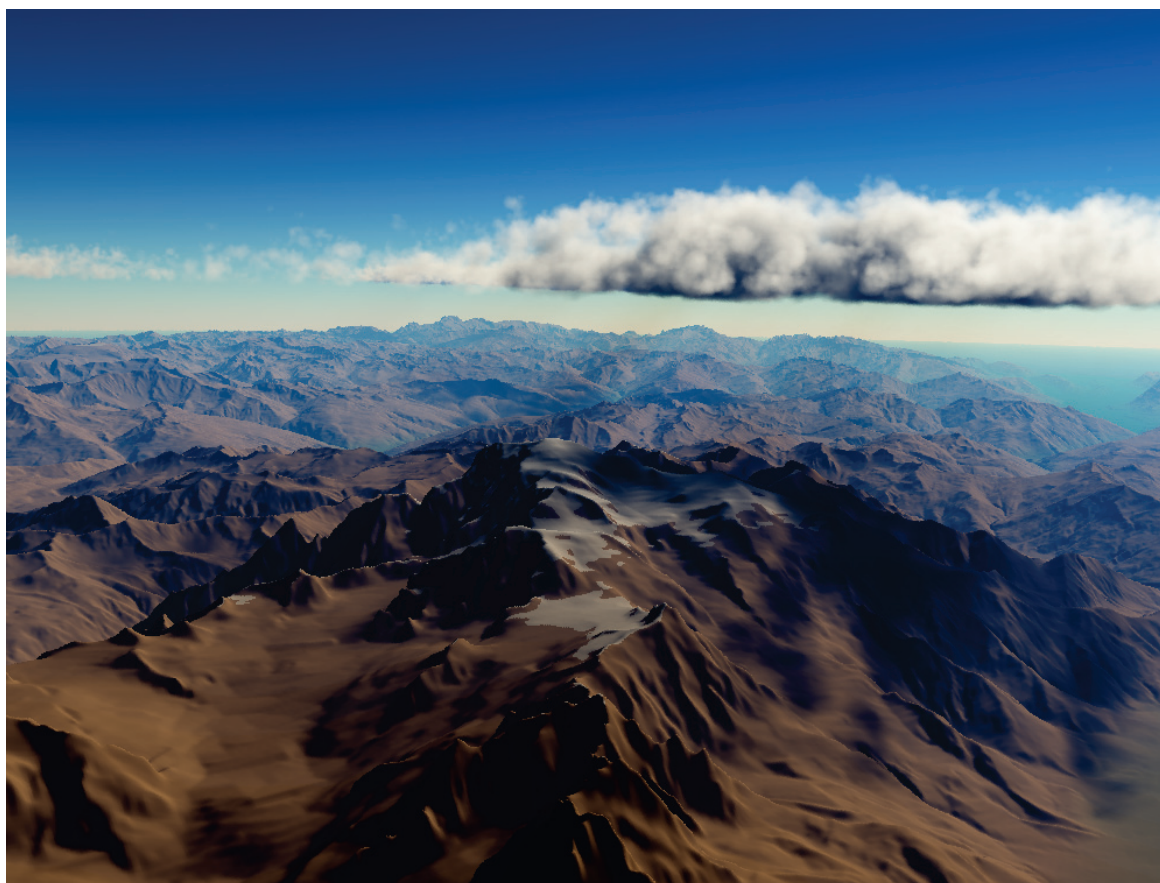
**Définition 17 (Fonction de visibilité)** La fonction de visibilité exprime l'absence d'obstacle obstruant le passage d'un rayon entre deux positions

$$V(\mathbf{p}, \mathbf{p}') = \begin{cases} 0 & \text{si un obstacle existe entre } \mathbf{p} \text{ et } \mathbf{p}' \\ 1 & \text{sinon} \end{cases} \quad (\text{A.11})$$

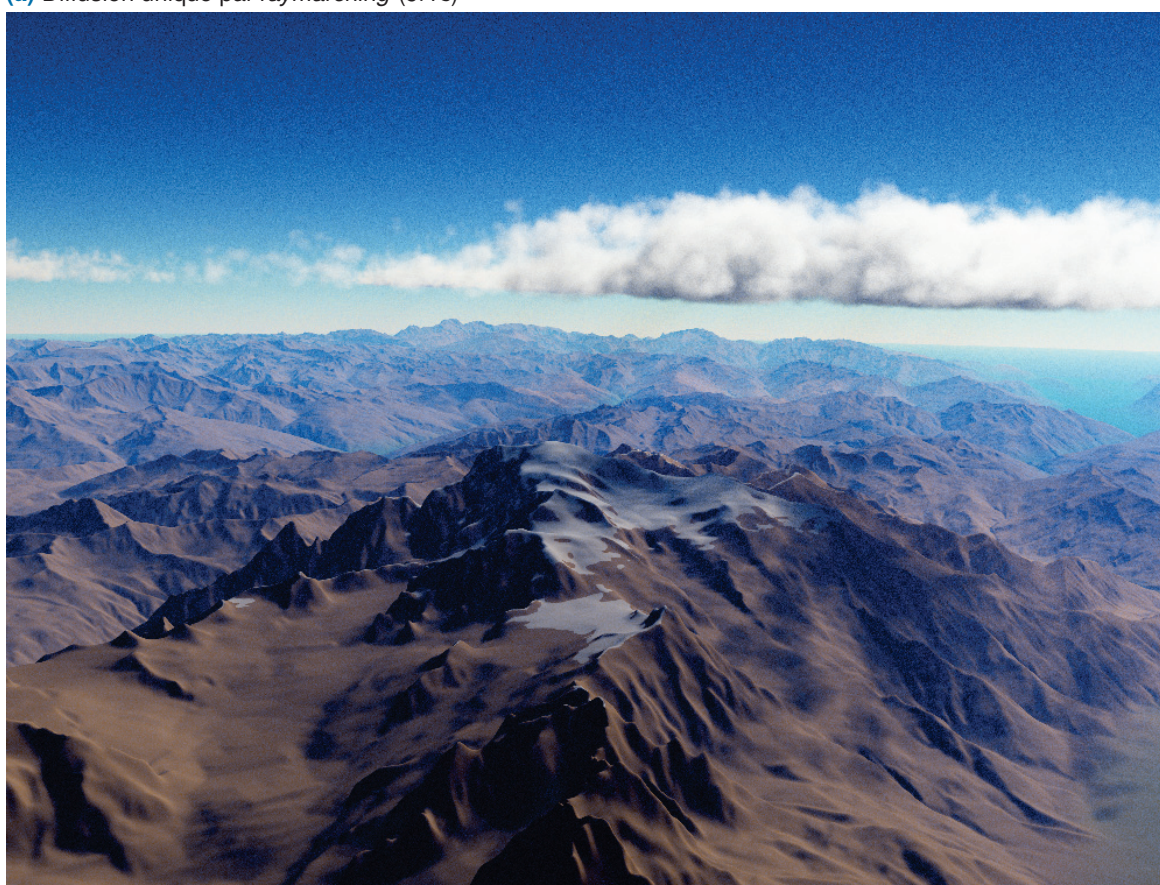
### A.5.1 Approche par *single scattering* et *raymarching*

En ne considérant qu'un seul événement de diffusion (*single scattering*) par chemin, un rayon primaire est lancé depuis chaque pixel dans la direction d'observation, et un rayon secondaire est lancé dans la direction du soleil à chaque pas le long du rayon primaire. L'évaluation de la transmittance des rayons d'un même pixel peut ainsi être factorisée efficacement sur leur portion commune le long du rayon primaire.





(a) Diffusion unique par *raymarching* (3.4 s)



(b) Multiples diffusions par *path tracing* (281 s, 200 échantillons par pixel)

**FIGURE A.1.** – Comparaison des méthodes de lancer de rayon sur une même scène.

Soit  $\omega$  la direction du rayon primaire, et  $[\mathbf{p}_0, \mathbf{p}_n]$  le segment atmosphérique du rayon primaire découpé en  $n$  pas ; la valeur du pixel associé est estimée par :

$$\mathbf{I} \approx V(\mathbf{p}_0, \mathbf{p}_n) \cdot \mathbf{I}_0 + \frac{1}{n+1} \left[ (1 - V(\mathbf{p}_0, \mathbf{p}_n)) \cdot \mathbf{I}'_0 + \sum_{i=1}^n V(\mathbf{p}_0, \mathbf{p}_i) \cdot \mathbf{I}_i \right] \quad (\text{A.12})$$

où  $I_0$  et  $I'_0$  sont les contributions du rayon primaire d'ordre zéro et un, tandis que les  $I_i$  sont les contributions des rayons secondaires d'ordre un.

### Ordre zéro

En l'absence d'intersection avec la surface, la contribution du rayon primaire à la valeur de son pixel représente l'éclairage solaire direct, *id est* l'éclairage d'ordre zéro. Si la direction du rayon primaire  $\omega$  coïncide avec la direction d'illumination du champ solaire  $\omega_s$ , l'éclairage direct est proportionnel à la transmittance atmosphérique et à l'intensité du rayonnement solaire  $\mathbf{I}_s$  :

$$\mathbf{I}_0 = \begin{cases} \mathbf{T}_{atm}(\mathbf{p}_0, \mathbf{p}_n) \circ \mathbf{I}_s & \text{si } \omega \cdot (-\omega_s) \approx 1 \\ \mathbf{0} & \text{sinon} \end{cases} \quad (\text{A.13})$$

### Ordre un

Dans le cas où un rayon rencontre la surface du terrain, la contribution du rayon à l'éclairage est déplacée à un ordre supérieur.

Ainsi, si c'est un rayon primaire qui rencontre la surface, sa contribution après un rebond participe plutôt à l'éclairage d'ordre un. En cas de seconde intersection avec le terrain, la contribution du rayon primaire concerne l'éclairage d'ordre deux et est donc ignorée dans l'approximation de diffusion unique (*single scattering*). Soit  $\mathbf{p}_0$  l'origine du rayon primaire,  $\mathbf{p}_t$  le point d'intersection avec le terrain, et  $\mathbf{p}_f$  le point de sortie du milieu atmosphérique après intersection ; la contribution du rayon primaire à l'éclairage de premier ordre est

$$\mathbf{I}'_0 = \left[ \alpha_t \cdot f_t(\omega, -\omega_s) \cdot V(\mathbf{p}_t, \mathbf{p}_f) \right] \cdot \left[ \mathbf{T}_{atm}(\mathbf{p}_t, \mathbf{p}_f) \circ \mathbf{T}_{atm}(\mathbf{p}_0, \mathbf{p}_t) \circ \mathbf{I}_s \right] \quad (\text{A.14})$$

où  $\alpha_t$  et  $f_t$  sont l'albédo et la fonction bidirectionnelle de diffusion (*BSDF*) de la surface du terrain.

La contribution de chaque rayon secondaire a une forme similaire à l'équation A.14, mais au lieu d'être dévié par la surface, le rayon est plutôt diffusé par le milieu. Soit  $\mathbf{p}_i$  et  $\mathbf{p}_f$  le point

de diffusion forcée du  $i^{\text{e}}$  rayon secondaire et son point de sortie du milieu atmosphérique ; soit  $\alpha$  l'albédo du milieu ; la contribution du rayon secondaire est

$$\mathbf{I}_i = \left[ \alpha \cdot f(\omega, -\omega_s) \cdot V(\mathbf{p}_i, \mathbf{p}_f) \right] \cdot \left[ \mathbf{T}_{atm}(\mathbf{p}_0, \mathbf{p}_i) \circ (\mathbf{1} - \mathbf{T}_{atm}(\mathbf{p}_{i-1}, \mathbf{p}_i)) \circ \mathbf{T}_{atm}(\mathbf{p}_i, \mathbf{p}_f) \circ \mathbf{I}_s \right] \quad (\text{A.15})$$

La contribution est ainsi d'autant plus importante que le milieu est dense au point de diffusion forcée après le dernier pas, *id est* le segment  $[\mathbf{p}_{i-1}, \mathbf{p}_i]$ .

## Évaluation de la transmittance par quadrature

La transmittance entre les deux extrémités  $\mathbf{p}_0$  et  $\mathbf{p}_n$  d'un rayon peut être estimée par quadrature en avançant et en prenant à chacun des  $n$  pas de longueur  $l_i$  un échantillon de densité (*raymarching*) :

$$\mathbf{T}_{atm}(\mathbf{p}_0, \mathbf{p}_n) \approx e^{-\sum_{i=1}^n \mu_t(\mathbf{p}_i) l_i} \quad (\text{A.16})$$

$$\approx \prod_{i=1}^n e^{-\mu_t(\mathbf{p}_i) l_i} \quad (\text{A.17})$$

Par simplicité, la longueur de chaque pas peut être constante, ou être proportionnelle à la densité afin de limiter l'erreur d'intégration [110]. Elle peut aussi augmenter en fonction de la distance de manière à intégrer plus précisément les détails de la densité la plus proche de l'observateur.

### A.5.2 Approche par échantillonnage de l'espace des chemins

En présence d'un milieu participatif comme l'atmosphère, en plus de pouvoir rencontrer une surface qui le dévie, chaque rayon peut être diffusé après une certaine distance (*free flight*) et dans une nouvelle direction.

Bien qu'il soit possible de construire un chemin échantillon dans sa globalité, il est commun de le construire de proche en proche en échantillonnant récursivement la profondeur avant et la direction après chaque événement de déviation.

#### Échantillonnage de la profondeur

La profondeur à laquelle un événement de diffusion se produit est une variable aléatoire dont la fonction de répartition est donnée par la transmittance. La transmittance entre deux points exprime ainsi la probabilité qu'un événement de diffusion se produise entre ces deux points. Dans un milieu de densité homogène, la profondeur optique est proportionnelle à la distance et l'expression de la transmittance en fonction de la profondeur  $t$  est simplement  $1 - e^{-\mu_t t}$ . Cette expression est assez simple pour appliquer la méthode de la transformée



inverse. Soit  $X$  une variable aléatoire de loi uniforme sur  $[0, 1]$ , la profondeur de diffusion s'exprime alors par  $-\frac{\ln(1-X)}{\mu t}$ .

Dans un milieu hétérogène la profondeur optique dépend du chemin parcouru et l'expression de la transmittance est inconnue ou trop complexe pour être inversée. La profondeur optique d'un segment peut facilement être estimée par méthode de MONTE CARLO en prenant des échantillons uniformément sur le segment, mais cette estimation ne peut être utilisée pour inverser la transmittance car  $\mathbb{E}(T) \neq e^{\mathbb{E}(\tau)}$  (inégalité de JENSEN).

Une solution consiste à appliquer la méthode du rejet pour reproduire indirectement la distribution de la profondeur de diffusion. Pour ce faire, un majorant de la probabilité d'interaction est utilisé :  $\mu_{\max} \geq \max(\mu(\mathbf{p}))$ . Ce majorant constant permet de se ramener au cas d'un milieu homogène dans lequel nous savons échantillonner la profondeur de diffusion directement selon sa fonction de répartition. Après avoir tiré une profondeur  $t$  de diffusion dans le milieu fictif, il ne reste plus qu'à faire une tentative stochastique de collision selon la probabilité  $\frac{\mu(t)}{\mu_{\max}}$ . Ce principe donne lieu à l'algorithme de *Woodcock tracking* aussi appelé *delta tracking* qui calcule la profondeur du prochain événement stochastique de diffusion (algorithme 1).

---

**Algorithme 1** : Pseudo-code de l'algorithme de *delta tracking* évaluant stochastiquement la profondeur du prochain événement de diffusion d'un rayon monochrome d'origine  $\mathbf{p}$ , de direction  $\omega$  et de profondeur maximale  $t_{\max}$  dans un milieu hétérogène.

---

```

1 DeltaTracking ( $\mathbf{p}, \omega, t_{\max}$ ) :
2    $t \leftarrow 0$ 
3   while  $t < t_{\max}$  do
4      $r \leftarrow \text{rand}()$ 
5      $t \leftarrow t - \frac{\ln(1-r)}{\mu_{\max}}$ 
6     if  $X \leq \frac{\mu(\mathbf{p}+t\omega)}{\mu_{\max}}$  then break
7   end
8   return  $t$ 
9 end

```

---

Dans notre cas, le milieu est constitué des nuages et des gaz de l'atmosphère que nous considérons comme indépendants. Le majorant choisi doit alors s'appliquer à la somme entre la probabilité d'interaction des deux milieux. De plus, nous considérons le caractère spectral de l'atmosphère qui peut être traité en appliquant l'algorithme de *delta tracking* séparément pour chaque longueur d'onde souhaitée. Procéder ainsi représente cependant un certain gaspillage car chaque chemin construit pour une longueur d'onde donnée pourrait servir au reste du spectre. C'est justement l'idée derrière l'algorithme de *spectral tracking* [112] qui réalise un échantillonnage préférentiel multiple de la profondeur de diffusion selon les probabilités de diffusion des différentes longueurs d'onde (algorithme 2 page suivante). Le majorant doit dans ce cas s'appliquer à toutes les longueurs d'onde considérées. De plus, chaque tentative de collision spectrale doit repondérer le flux (*throughput*) qui représente l'importance d'un chemin échantillonné.

---

**Algorithme 2** : Pseudo-code de l'algorithme de *spectral tracking* évaluant stochastiquement la profondeur du prochain événement de diffusion d'un rayon spectral d'origine  $\mathbf{p}$ , de direction  $\omega$  et de profondeur maximale  $t_{\max}$  dans un milieu hétérogène. Chaque tentative de rejet répondre le flux spectral  $\mathbf{W}$ .

---

```

1 SpectralTracking ( $\mathbf{p}, \omega, t_{\max}, \mathbf{W}$ ) :
2    $t \leftarrow 0$ 
3   while  $t < t_{\max}$  do
4      $r_1 \leftarrow \text{rand}()$ 
5      $t \leftarrow t - \frac{\ln(1-r_1)}{\mu_{\max}}$ 
6      $P \leftarrow \text{avg}_{\lambda}(\mu(\mathbf{p} + t \cdot \omega) \circ \mathbf{W})$ 
7      $P_n \leftarrow \text{avg}_{\lambda}((\mu_{\max} - \mu(\mathbf{p} + t \cdot \omega)) \circ \mathbf{W})$ 
8      $P \leftarrow \frac{P}{P + P_n}$ 
9      $P_n \leftarrow \frac{P_n}{P + P_n}$ 
10     $r_2 \leftarrow \text{rand}()$ 
11    if  $r_2 \leq P$  then
12       $\mathbf{W} \leftarrow \mathbf{W} \circ \frac{\mu(\mathbf{p} + t \cdot \omega)}{\mu_{\max} \cdot P}$ 
13      break
14    else
15       $\mathbf{W} \leftarrow \mathbf{W} \circ \frac{\mu_{\max} - \mu(\mathbf{p} + t \cdot \omega)}{\mu_{\max} \cdot P_n}$ 
16    end
17  end
18  return ( $\mathbf{W}, t$ )
19 end

```

---



## Échantillonnage de la direction

La direction de diffusion peut être choisie par échantillonnage uniforme de la sphère des directions. Lorsque la fonction de phase du milieu est hautement anisotrope, il peut cependant être intéressant d'appliquer un échantillonnage préférentiel d'après la fonction de phase. Les fonctions de phase complexes, comme dans le cas de la diffusion de Mie n'ont cependant généralement pas une expression assez simple pour appliquer la méthode de la transformée inverse.

## Évaluation de la transmittance par *ratio tracking*

Une estimation stochastique non biaisée de la transmittance sur un segment peut être obtenue en utilisant l'algorithme de *delta tracking*. La transmittance est alors estimée à zéro en cas de collision, et à un en l'absence de toute collision. Bien que la transmittance puisse être évaluée plus précisément en calculant la moyenne de plusieurs estimations binaires, la densité du milieu doit être récupérée de nombreuses fois en de nombreux points différents sur le segment.

Il existe une autre approche qui fournit directement une estimation flottante de la transmittance et exploite mieux les échantillons de densité : le *ratio tracking* (algorithme 3). À la différence du *delta tracking*, le *ratio tracking* force la traversée de tout le segment tant qu'aucune tentative de collision ne tombe pas en un point où la collision est certaine (si la probabilité locale est égale au majorant). À chaque tentative de collision, la transmittance est multipliée par la probabilité de rejet.

---

**Algorithme 3** : Pseudo-code de l'algorithme de *ratio tracking* évaluant stochastiquement le vecteur de transmittance spectrale à la traversée d'un milieu hétérogène par un rayon d'origine  $\mathbf{p}$ , de direction  $\omega$  et de profondeur maximale  $t_{\max}$ .

---

```
1 RatioTracking ( $\mathbf{p}$ ,  $\omega$ ,  $t_{\max}$ ) :
2    $\mathbf{T} \leftarrow 1$ 
3    $t \leftarrow 0$ 
4   while  $t < t_{\max}$  do
5      $r \leftarrow \text{rand}()$ 
6      $t \leftarrow t - \frac{\ln(1-r)}{\mu_{\max}}$ 
7      $\mathbf{T} \leftarrow \mathbf{T} \circ \left( 1 - \frac{\mu(\mathbf{p}+t\cdot\omega)}{\mu_{\max}} \right)$       /* où  $\circ$  est le produit terme à terme */
8
9     if  $\mathbf{T} = 0$  then break
10  end
11  return  $\mathbf{T}$ 
12 end
```

---



# Tables des symboles

## Contexte général

$C$	Classe de continuité.
$\tilde{I}$	Fonction d'approximation.
$d$	Fonction de distance.
$\mathcal{I}$	Fonction d'interpolation.
$\mathbb{E}$	Espérance.
$\circ$	Produit terme à terme.
$U$	Loi uniforme.
$\bar{\varphi}$	Colatitute.
$\omega$	Vecteur direction.
$\varphi$	Latitude.
$\theta$	Longitude.
$n$	Nombre entier.
$\mathbf{p}$	Position.
$r$	Rayon.
$t$	Temps.
$\Delta t$	Pas de temps.

## Surfaces implicites

$\Sigma$	Opérateur de mélange.
$g$	Fonction d'atténuation.
$\cap$	Opérateur d'intersection.
$f$	Fonction de potentiel.
$\cup$	Opérateur d'union.
$\mathcal{C}$	Primitive.
$D$	Densité maximale de primitive.
$R$	Rayon de primitive.
$\mathcal{S}$	Squelette.
$S$	Surface.
$T$	Isovaleur.

## Bruits

$a$	Amplitude.
$n$	Fonction de bruit.
$f$	Fréquence.
$l$	Lacunarité.
$o$	Nombre d'octaves.
$\rho$	Persistance.

## Rendu

$\theta_a$	Section transversale d'absorption.
$\alpha$	Albédo.
$\theta$	Section transversale.
$\rho$	Densité.
$\mu_a$	Probabilité d'une absorption.
$\mu_t$	Probabilité d'atténuation.
$L_e$	Luminance émise.
$\mu$	Probabilité d'une interaction entre particule et rayon.
$\tau$	Profondeur optique.
$f$	Fonction de phase.
$f_g$	Fonction de phase des gaz atmosphériques.
$f_c$	Fonction de phase des nuages.
$f_t$	Fonction bidirectionnelle de diffusion du terrain.
$L$	Luminance.
$L_{atm}$	Luminance atmosphérique spectrale.
$f$	Mesure.
$\mu_{sg}$	Probabilité spectrale d'une diffusion par les gaz atmosphériques.
$\mu_{sc}$	Probabilité d'une diffusion par les nuages.
$\mu_s$	Probabilité d'une diffusion.
$T$	Transmittance.
$T_g$	Transmittance spectrale des gaz atmosphériques.
$T_{atm}$	Transmittance atmosphérique spectrale.
$T_c$	Transmittance des nuages.
$V$	Fonction de visibilité.
$\bar{x}$	Chemin d'illumination.
$\mathcal{P}$	Domaine des chemins d'illumination.
$\theta_s$	Section transversale de diffusion.
$W$	Flux.

## Modèle d'atmosphère

$e$	Élévation de la base nuageuse.
$\mathcal{T}$	Type nuageux.
$\mathcal{C}$	Primitive de contrôle.
$a$	Champ atmosphérique de densité.
$c$	Fonction de contrôle.
$C$	Fonction de couche.
$s$	Fonction de texture nuageuse.
$\sigma$	Fonction de normalisation.
$\rho$	Fonction de présence.
$\times$	Opérateur de rétrécissement–estompage.
$b$	Élévation minimale de la base nuageuse.
$a$	Élévation minimale de la base nuageuse.
$C$	Centre de primitive.
$\Omega$	Support compact de primitive.
$R_H$	Rayon horizontal de primitive.
$R_V$	Rayon vertical de primitive.

$\alpha$	Facteur de rétrécissement–estompage.
$r$	Extension verticale.

## Métamorphose

$\kappa$	Fonction de coût.
$\kappa_h$	Fonction de coût spécifique à l'altitude du terrain.
$K$	Fonctionnelle de coût d'un chemin.
$\kappa_s$	Fonction de coût spécifique à la pente du terrain.
$\kappa_v$	Fonction de coût spécifique au vent.
$h$	Carte de hauteur.
$C_{\leftarrow j}$	Primitive interpolante faisant apparaître une portion de la $j^{\text{e}}$ primitive finale.
$\overline{B_j}$	Primitive fantôme associée à la $j^{\text{e}}$ primitive finale..
$B$	Image clef finale.
$L_{\leftarrow}$	Liens de correspondance antichronologique.
$B_j$	Primitive statique finale d'indice $j$ .
$n_B$	Nombre de primitives finales.
$t_B$	Temps associé à l'image clef finale.
$M$	Fonctionnelle de masse.
$d^+$	Degré sortant d'un nœud.
$\Gamma$	Fonction de trajectoire.
$\tilde{h}$	Carte de hauteur lissée.
$\Gamma^*$	Fonction de plus court chemin.
$\kappa_{OT}$	Fonctionnelle de coût de transport.
$\mathbf{v}$	Champ vectoriel de vent.
$\kappa_{\max}$	Valeur de seuil pour la relaxation.
$C_{i \rightarrow}$	Primitive interpolante faisant disparaître une portion de la $i^{\text{e}}$ primitive initiale.
$\overline{A_i}$	Primitive fantôme associée à la $i^{\text{e}}$ primitive initiale..
$A$	Image clef initiale.
$L_{\rightarrow}$	Liens de correspondance chronologique.
$A_i$	Primitive statique initiale d'indice $i$ .
$n_A$	Nombre de primitives initiales.
$t_A$	Temps associé à l'image clef initiale.
$L$	Liens de correspondance.
$w$	Poids de correspondance.
$h_{\max}$	Hauteur maximale.
$v_{\max}$	Vitesse de vent maximale.
$h_{\min}$	Hauteur minimale.
$C_{i,j}$	Primitive interpolante entre la $i^{\text{e}}$ primitive initiale et la $j^{\text{e}}$ primitive finale.
$C$	Ensemble des primitives interpolantes.
$\tilde{t}$	Temps normalisé sur l'intervalle temporel de métamorphose.
$\mathbf{M}^*$	Plan de transport optimal.
$\overline{L_{\leftarrow}}$	Liens de correspondance antichronologique invalidés par seuillage.
$\overline{L_{\rightarrow}}$	Liens de correspondance chronologique invalidés par seuillage.
$\overline{L}$	Liens de correspondance invalidés par seuillage.
$\Omega_{\Gamma}$	Domaine des chemins.
$\mathcal{G}_{\Gamma}$	Graphe représentant le domaine discret des chemins.
$\Omega$	Domaine spatial.



$T$	Domaine temporel.
$M$	Plan de transport.
$m$	Masse.
$\tilde{m}$	Masse normalisée.

# Bibliographie

- [1] Bart ADAMS *et al.* Adaptively Sampled Particle Fluids. *ACM Transactions on Graphics* 26.3 (juil. 2007). DOI : [10.1145/1276377.1276437](https://doi.org/10.1145/1276377.1276437). URL : <https://core.ac.uk/download/pdf/34424595.pdf> (cf. p. 50).
- [2] Gladimir Valerio Guimaraes BARANOSKI *et al.* Simulating the aurora borealis. *Pacific Conference on Computer Graphics and Applications*. 2000, p. 422-432. DOI : [10.1109/PCCGA.2000.883852](https://doi.org/10.1109/PCCGA.2000.883852). URL : [https://www.researchgate.net/profile/Trond\\_Trondsen/publication/3875546\\_Simulating\\_the\\_aurora\\_borealis/links/02bfe50c8abdd5803f000000/Simulating-the-aurora-borealis.pdf](https://www.researchgate.net/profile/Trond_Trondsen/publication/3875546_Simulating_the_aurora_borealis/links/02bfe50c8abdd5803f000000/Simulating-the-aurora-borealis.pdf) (cf. p. 21).
- [3] Gladimir Valerio Guimaraes BARANOSKI *et al.* Simulating the aurora. *The Journal of Visualizattion and Computer Animation banner* (fév. 2003). DOI : [10.1002/vis.304](https://doi.org/10.1002/vis.304). URL : <http://pedrinho.cs.uwaterloo.ca/~gvgsbaran/cgpub/aurora1.pdf> (cf. p. 21).
- [4] Gladimir Valerio Guimaraes BARANOSKI *et al.* Simulating the Dynamics of Auroral Phenomena. *ACM Transactions on Graphics* 24.1 (jan. 2005), 37-59. DOI : [10.1145/1037957.1037960](https://doi.org/10.1145/1037957.1037960). URL : <https://cs.uwaterloo.ca/~jwlwan/papers/BaranoskiWanETAL05.pdf> (cf. p. 21).
- [5] Charles Welton Ferreira BARBOSA, Yoshinori DOBASHI et Tsuyoshi YAMAMOTO. Adaptive Cloud Simulation Using Position Based Fluids. *Computer Animation and Virtual Worlds* 26.3-4 (mai 2015), 367-375. DOI : [10.1002/cav.1657](https://doi.org/10.1002/cav.1657) (cf. p. 50, 63).
- [6] Michael F. BARNESLEY *et al.* *The Science of Fractal Images*. 1988. ISBN : 978-1-4612-8349-2. DOI : [10.1007/978-1-4612-3784-6](https://doi.org/10.1007/978-1-4612-3784-6) (cf. p. 57, 63).
- [7] A. R. BATISTA-TOMÁS *et al.* Classification and dynamics of tropical clouds by their fractal dimension. *Quarterly Journal of the Royal Meteorological Society* 142.695 (2016), 983-988. DOI : [10.1002/qj.2699](https://doi.org/10.1002/qj.2699) (cf. p. 57).
- [8] Philippe BLASI, Bertrand SAEC et Christophe SCHLICK. A Rendering Algorithm for Discrete Volume Density Objects. *Computer Graphics Forum* 12.3 (1993), 201-210. DOI : [10.1111/1467-8659.1230201](https://doi.org/10.1111/1467-8659.1230201) (cf. p. 37, 43).
- [9] James Frederick BLINN. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics* 1.3 (juil. 1982), 235-256. DOI : [10.1145/357306.357310](https://doi.org/10.1145/357306.357310). URL : <https://cumincad.architexturez.net/system/files/pdf/6094.content.pdf> (cf. p. 65, 68, 69).
- [10] James Frederick BLINN. Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. *SIGGRAPH Computer Graphics* 16.3 (juil. 1982), 21-29. DOI : [10.1145/965145.801255](https://doi.org/10.1145/965145.801255). URL : <https://design.osu.edu/carlson/history/PDFs/blinn-dusty.pdf> (cf. p. 30, 31, 43, 139).
- [11] Nicolas BONNEEL *et al.* Displacement Interpolation Using Lagrangian Mass Transport. *ACM Transactions on Graphics* 30.6 (2011). URL : <http://www.cs.ubc.ca/labs/imager/tr/2011/DisplacementInterpolation/DisplacementSigAsia.pdf> (cf. p. 114).

- [12] Antoine BOUTHORS et Fabrice NEYRET. Modeling Clouds Shape. *Eurographics Short Papers*. Août 2004. URL : [https://hal.inria.fr/inria-00537462/file/BN04\\_clouds.pdf](https://hal.inria.fr/inria-00537462/file/BN04_clouds.pdf) (cf. p. 60, 63).
- [13] Antoine BOUTHORS et Fabrice NEYRET. Real-time realistic rendering of clouds. Thèse de doct. Université Joseph Fourier, juin 2008. URL : <http://evasion.imag.fr/Publications/2008/BN08a/thesis.pdf> (cf. p. 26).
- [14] Antoine BOUTHORS, Fabrice NEYRET et Sylvain LEFEBVRE. Real-time realistic illumination and shading of stratiform clouds. *Eurographics Workshop on Natural Phenomena*. 2006. URL : <http://www-evasion.imag.fr/Publications/2006/BNL06/bnl06-elec.pdf> (cf. p. 26, 41-43).
- [15] Antoine BOUTHORS *et al.* Interactive multiple anisotropic scattering in clouds. *ACM Symposium on Interactive 3D Graphics and Games*. 2008, p. 173-182. ISBN : 978-1-59593-983-8. DOI : 10.1145/1342250.1342277. URL : <http://evasion.imag.fr/Publications/2008/BNMBC08/clouds.pdf> (cf. p. 42, 43, 53).
- [16] Thomas BRESSON. © ⓘ Creative Commons - Attribution 2.0. 25 mai 2009. URL : [https://upload.wikimedia.org/wikipedia/commons/8/85/Thomas\\_Bresson\\_-\\_Eclairs-1\\_%28by%29.jpg](https://upload.wikimedia.org/wikipedia/commons/8/85/Thomas_Bresson_-_Eclairs-1_%28by%29.jpg) (cf. p. 20).
- [17] Eric BRUNETON et Fabrice NEYRET. Precomputed Atmospheric Scattering. *Computer Graphics Forum* 27.4 (2008), 1079-1086. DOI : 10.1111/j.1467-8659.2008.01245.x. URL : [http://www-ljk.imag.fr/Publications/Basilic/com.lmc.publi.PUBLI\\_Article@11e7cdda2f7\\_f64b69/article.pdf](http://www-ljk.imag.fr/Publications/Basilic/com.lmc.publi.PUBLI_Article@11e7cdda2f7_f64b69/article.pdf) (cf. p. 27, 41, 43).
- [18] Subrahmanyan CHANDRASEKHAR. *Radiative Transfer*. 1950. ISBN : 0-486-60590-6. URL : <https://ia800308.us.archive.org/24/items/RadiativeTransfer/Chandrasekhar-RadiativeTransfer.pdf> (cf. p. 31, 36, 137).
- [19] Junping CHEN, Yunchi CEN et Xiaohui LIANG. Sketch-based Cloud Model Retrieval for Cumulus Cloud Scene Construction. *International Conference on Digital Signal Processing* (2018), 166-170. DOI : 10.1145/3193025.3193043 (cf. p. 56).
- [20] Yu CHUNG-MIN et Wang CHUNG-MING. An Effective Framework for Cloud Modeling, Rendering, and Morphing. *Journal of Information Science and Engineering* 27.3 (2011), 891-913. URL : [http://www.iis.sinica.edu.tw/page/jise/2011/201105\\_06.pdf](http://www.iis.sinica.edu.tw/page/jise/2011/201105_06.pdf) (cf. p. 129).
- [21] Peter COLLIENNE *et al.* Physically based rendering of the martian atmosphere. *Virtuelle und Erweiterte Realität* 10 (2013), 97-108. URL : [http://elib.dlr.de/86477/1/Collienne\\_GI\\_VRAR\\_2013.pdf](http://elib.dlr.de/86477/1/Collienne_GI_VRAR_2013.pdf) (cf. p. 35, 43).
- [22] Yoshinori DOBASHI, Tomoyuki NISHITA et Tsuyoshi OKITA. Animation of Clouds Using Cellular Automaton. *Computer Graphics and Imaging*. 1998, p. 251-256 (cf. p. 48).
- [23] Yoshinori DOBASHI, Yusuke SHINZO et Tsuyoshi YAMAMOTO. Modeling of Clouds from a Single Photograph. *Computer Graphics Forum* 29.7 (2010), 2083-2090. DOI : 10.1111/j.1467-8659.2010.01795.x. URL : [http://nis-ei.eng.hokudai.ac.jp/~doba/projects/photo\\_cloud\\_modeling/photoCloudModeling.pdf](http://nis-ei.eng.hokudai.ac.jp/~doba/projects/photo_cloud_modeling/photoCloudModeling.pdf) (cf. p. 54, 61, 63).
- [24] Yoshinori DOBASHI, Tsuyoshi YAMAMOTO et Tomoyuki NISHITA. Efficient Rendering of Lightning Taking into Account Scattering Effects Due to Cloud and Atmospheric Particles. *Pacific Conference on Computer Graphics and Applications*. 2001, p. 390-399. ISBN : 0-7695-1227-5. URL : [http://nishitalab.org/user/nis/cdrom/pg/pg01\\_lightning.pdf](http://nishitalab.org/user/nis/cdrom/pg/pg01_lightning.pdf) (cf. p. 20).



- [25] Yoshinori DOBASHI, Tsuyoshi YAMAMOTO et Tomoyuki NISHITA. Interactive Rendering of Atmospheric Scattering Effects Using Graphics Hardware. *SIGGRAPH/Eurographics Workshop on Graphics Hardware*. 2002, p. 99-108. ISBN : 1-58113-580-7. DOI : [10.2312/EGGH/EGGH02/099-108](https://doi.org/10.2312/EGGH/EGGH02/099-108). URL : <http://nis-ei.eng.hokudai.ac.jp/~doba/papers/HWW02.pdf> (cf. p. 27).
- [26] Yoshinori DOBASHI, Tsuyoshi YAMAMOTO et Tomoyuki NISHITA. A Controllable Method for Animation of Earth-scale Clouds. *Computer Animation and Social Agents*. Juil. 2006, p. 43-52. URL : <https://pdfs.semanticscholar.org/2a6a/6e003d6380ad8cb87988be7c59dcb9de1535.pdf> (cf. p. 48, 61, 63).
- [27] Yoshinori DOBASHI *et al.* Modeling of Clouds from Satellite Images Using Metaballs. *Pacific Conference on Computer Graphics and Applications*. 1998, p. 53-60. ISBN : 0-8186-8620-0. DOI : [10.1109/PCCGA.1998.731998](https://doi.org/10.1109/PCCGA.1998.731998). URL : <http://nis-ei.eng.hokudai.ac.jp/~doba/papers/pg98.pdf> (cf. p. 54, 61, 63).
- [28] Yoshinori DOBASHI *et al.* Using metaballs to modeling and animate clouds from satellite images. *The Visual Computer* 15.9 (déc. 1999), 471-482. DOI : [10.1007/s003710050193](https://doi.org/10.1007/s003710050193). URL : <http://nishitalab.org/user/nis//cdrom/cad/90150471.pdf> (cf. p. 54).
- [29] Yoshinori DOBASHI *et al.* A Simple, Efficient Method for Realistic Animation of Clouds. *SIGGRAPH*. 2000, p. 19-28. ISBN : 1-58113-208-5. DOI : [10.1145/344779.344795](https://doi.org/10.1145/344779.344795) (cf. p. 27, 48, 49, 63).
- [30] Yoshinori DOBASHI *et al.* Feedback Control of Cumuliform Cloud Formation based on Computational Fluid Dynamics. *ACM Transactions on Graphics* 27.3 (août 2008). DOI : [10.1145/1360612.1360693](https://doi.org/10.1145/1360612.1360693). URL : [https://ime.ist.hokudai.ac.jp/~doba/projects/cloud\\_control/control\\_cloud.pdf](https://ime.ist.hokudai.ac.jp/~doba/projects/cloud_control/control_cloud.pdf) (cf. p. 53, 55, 63).
- [31] David S. EBERT. Volumetric Modeling with Implicit Functions : A Cloud is Born. *SIGGRAPH Visual*. 1997. ISBN : 0-89791-921-1. DOI : [10.1145/259081.259233](https://doi.org/10.1145/259081.259233). URL : [https://engineering.purdue.edu/~ebertd/papers/S97\\_sketch.pdf](https://engineering.purdue.edu/~ebertd/papers/S97_sketch.pdf) (cf. p. 59, 63).
- [32] David S. EBERT et Richard E. PARENT. Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques. *SIGGRAPH Computer Graphics* 24.4 (1990), 357-366. DOI : [10.1145/97880.97918](https://doi.org/10.1145/97880.97918). URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.7193&rep=rep1&type=pdf> (cf. p. 57, 63).
- [33] David S. EBERT *et al.* *Texturing and Modeling*. 1994. ISBN : 978-0-12-228760-2. DOI : [10.1016/C2009-0-22378-6](https://doi.org/10.1016/C2009-0-22378-6) (cf. p. 58, 61, 63).
- [34] David S. EBERT *et al.* *Texturing and Modeling*. 3<sup>e</sup> éd. 2002. ISBN : 978-1558608481. URL : [http://ismusicmake.googlecode.com/svn/trunk/rc\\_Paper/1558608486/Morgan%20Kaufmann%20Publishers%20-%20Texturing%20and%20Modeling.%20A%20Procedural%20Approach,%203rd%20Edition.pdf](http://ismusicmake.googlecode.com/svn/trunk/rc_Paper/1558608486/Morgan%20Kaufmann%20Publishers%20-%20Texturing%20and%20Modeling.%20A%20Procedural%20Approach,%203rd%20Edition.pdf) (cf. p. 59).
- [35] Oskár ELEK *et al.* Interactive Cloud Rendering Using Temporally-Coherent Photon Mapping. *Computers & Graphics* 36.8 (déc. 2012), 1109-1118. DOI : [10.1016/j.cag.2012.10.002](https://doi.org/10.1016/j.cag.2012.10.002). URL : [http://people.mpi-inf.mpg.de/~oelek/Papers/CloudRendering/CAG\\_2012\\_ElekRitschelWilkieSeidel.pdf](http://people.mpi-inf.mpg.de/~oelek/Papers/CloudRendering/CAG_2012_ElekRitschelWilkieSeidel.pdf) (cf. p. 40, 43).
- [36] Akila ELHADDAD *et al.* Real-Time Cloud Simulation Using Lennard-Jones Approximation. *Computer Animation and Social Agents*. 2016, p. 131-137. ISBN : 978-1-4503-4745-7. DOI : [10.1145/2915926.2915942](https://doi.org/10.1145/2915926.2915942) (cf. p. 51, 63).

- [37] FAMARTIN. © ⓘ Creative Commons - Attribution-ShareAlike 3.0. 10 mar. 2014. URL : [https://upload.wikimedia.org/wikipedia/commons/7/78/2014-03-10\\_08\\_47\\_44\\_Stratus\\_clouds\\_and\\_virga\\_in\\_Elko%2C\\_Nevada.JPG](https://upload.wikimedia.org/wikipedia/commons/7/78/2014-03-10_08_47_44_Stratus_clouds_and_virga_in_Elko%2C_Nevada.JPG) (cf. p. 88).
- [38] Eric GALIN et Samir AKKOUCHE. Blob Metamorphosis based on Minkowski Sums. *Computer Graphics Forum* 15.3 (1996), 143-153. DOI : 10.1111/1467-8659.1530143. URL : <http://liris.cnrs.fr/~egaline/Pdf/1996-blob-morphing.pdf> (cf. p. 114).
- [39] Eric GALIN et al. Procedural Generation of Roads. *Computer Graphics Forum* 29.2 (2010), 429-438. URL : [https://perso.liris.cnrs.fr/adrien.peytavie/publications/articles/EG2010\\_ProceduralGenerationOfRoads.pdf](https://perso.liris.cnrs.fr/adrien.peytavie/publications/articles/EG2010_ProceduralGenerationOfRoads.pdf) (cf. p. 111).
- [40] Geoffrey Y. GARDNER. Simulation of Natural Scenes Using Textured Quadric Surfaces. *SIGGRAPH Computer Graphics* 18.3 (juil. 1984). DOI : 10.1145/964965.808572 (cf. p. 56, 61).
- [41] Geoffrey Y. GARDNER. Visual simulation of clouds. *SIGGRAPH Computer Graphics* 19.3 (juil. 1985), 297-304. DOI : 10.1145/325165.325248. URL : <https://www.cs.drexel.edu/~david/Classes/Papers/p297-gardner.pdf> (cf. p. 39, 43, 56, 58, 61, 63).
- [42] Kshitiz GARG et Shree K. NAYAR. Photorealistic Rendering of Rain Streaks. *ACM Transactions on Graphics* 25.3 (juil. 2006), 996-1002. DOI : 10.1145/1141911.1141985. URL : [http://www1.cs.columbia.edu/CAVE/publications/pdfs/Garg\\_TOG06.pdf](http://www1.cs.columbia.edu/CAVE/publications/pdfs/Garg_TOG06.pdf) (cf. p. 22, 23).
- [43] Alexander GERST. Flying Through an Aurora. 29 août 2014. URL : <https://www.nasa.gov/sites/default/files/thumbnails/image/iss040e117941.jpg> (cf. p. 21).
- [44] Andrew GLASSNER. Computer-generated solar halos and sun dogs. *IEEE Computer Graphics and Applications* 16.2 (mar. 1996), 77-81. DOI : 10.1109/38.486687. URL : <https://www.glassner.com/wp-content/uploads/2014/04/CG-CGA-PDF-96-03-Solar-Halos-and-Sun-Dogs-2-Mar96.pdf> (cf. p. 25).
- [45] Andrew GLASSNER. Solar halos and sun dogs. *IEEE Computer Graphics and Applications* 16.1 (jan. 1996), 83-87. DOI : 10.1109/38.481627. URL : <https://www.glassner.com/wp-content/uploads/2014/04/CG-CGA-PDF-96-01-Solar-Halos-and-Sun-Dogs-1-Jan96.pdf> (cf. p. 25).
- [46] Andrew GLASSNER. The Digital Ceraunoscope : Synthetic Lightning and Thunder, Part 1. *IEEE Computer Graphics and Applications* 20.2 (mar. 2000), 89-93. DOI : 10.1109/38.824552. URL : <http://www.glassner.com/wp-content/uploads/2014/04/CG-CGA-PDF-00-03-The-Digital-Ceraunoscope-1-Mar00.pdf> (cf. p. 20).
- [47] Andrew GLASSNER. The Digital Ceraunoscope : Synthetic Lightning and Thunder, Part 2. *IEEE Computer Graphics and Applications* 20.3 (mai 2000), 92-96. DOI : 10.1109/38.844376. URL : <http://www.glassner.com/wp-content/uploads/2014/04/CG-CGA-PDF-00-05-The-Digital-Ceraunoscope-2-May00.pdf> (cf. p. 20).
- [48] GLG. © ⓘ Creative Commons - Attribution-ShareAlike 2.0. 26 mai 2005. URL : [https://upload.wikimedia.org/wikipedia/commons/6/66/Img20050526\\_0007\\_at\\_tannheim\\_cumulus.jpg](https://upload.wikimedia.org/wikipedia/commons/6/66/Img20050526_0007_at_tannheim_cumulus.jpg) (cf. p. 88).
- [49] Jean-Christophe GONZATO et Sylvain MARCHAND. Photo-Realistic Simulation and Rendering of Halos. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. Fév. 2001, p. 106-113. URL : <https://hal.archives-ouvertes.fr/hal-00308264/document> (cf. p. 25).



- [50] Cindy M. GORAL *et al.* Modeling the Interaction of Light Between Diffuse Surfaces. *SIGGRAPH Computer Graphics* 18.3 (juil. 1984), 213-222. DOI : [10.1145/964965.808601](https://doi.org/10.1145/964965.808601). URL : <https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S07/lectures/goral.pdf> (cf. p. 35).
- [51] Prashant GOSWAMI et Fabrice NEYRET. *Real-time Landscape-size Convective Clouds Simulation*. Rapp. tech. RR-8919. INRIA, fév. 2016. URL : <https://hal.inria.fr/hal-01325905/file/RR-8919.pdf> (cf. p. 50, 51, 61, 63).
- [52] Robert GREENLER. *Rainbows, Halos and Glories*. 1980. ISBN : 978-0521388658 (cf. p. 25).
- [53] Dave GRUBB. © ⓘ ⓘ Creative Commons - Attribution-ShareAlike 2.0. 18 mai 2015. URL : [https://live.staticflickr.com/5337/17235035983\\_c5b553acf2\\_o\\_d.jpg](https://live.staticflickr.com/5337/17235035983_c5b553acf2_o_d.jpg) (cf. p. 24).
- [54] Jakub GRUDZIŃSKI et Adrian DĘBOWSKI. Clouds and Atmospheric Phenomena Simulation in Real-Time 3D Graphics. *International Conference on Computer Vision/Computer Graphics Collaboration Techniques (MIRAGE)*. 2007, p. 11-127. ISBN : 978-3-540-71456-9. URL : <https://pdfs.semanticscholar.org/6986/57ff0d07070ace30ca028c11bc102c96de94.pdf> (cf. p. 53, 61, 63).
- [55] A. GUILLAUME *et al.* Horizontal and Vertical Scaling of Cloud Geometry Inferred from CloudSat Data. *Journal of the Atmospheric Sciences* 75.7 (2018), 2187-2197. DOI : [10.1175/JAS-D-17-0111.1](https://doi.org/10.1175/JAS-D-17-0111.1) (cf. p. 57).
- [56] Diego GUTIERREZ *et al.* Chasing the Green Flash : A Global Illumination Solution for Inhomogeneous Media. *Spring Conference on Computer Graphics*. 2004, p. 97-105. ISBN : 1-58113-914-4. DOI : [10.1145/1037210.1037225](https://doi.org/10.1145/1037210.1037225). URL : [http://giga.cps.unizar.es/~diegog/ficheros/pdf\\_papers/Chasing\\_the\\_green\\_flash\\_A\\_global\\_illumination\\_solution\\_for\\_inhomogeneous\\_media.pdf](http://giga.cps.unizar.es/~diegog/ficheros/pdf_papers/Chasing_the_green_flash_A_global_illumination_solution_for_inhomogeneous_media.pdf) (cf. p. 28).
- [57] Diego GUTIERREZ *et al.* Simulation of atmospheric phenomena. *Computers & Graphics* 30.6 (2006), 994-1010. DOI : [10.1016/j.cag.2006.05.002](https://doi.org/10.1016/j.cag.2006.05.002). URL : <https://pdfs.semanticscholar.org/1115/8de61b64b39f4b8ad46a78105b99115551e5.pdf> (cf. p. 28).
- [58] Jörg HABER, Marcus MAGNOR et Hans-Peter SEIDEL. Physically-based Simulation of Twilight Phenomena. *ACM Transactions on Graphics* 24.4 (oct. 2005), 1353-1373. DOI : [10.1145/1095878.1095884](https://doi.org/10.1145/1095878.1095884). URL : <http://graphics.tu-bs.de/static/people/magnor/publications/tog05.pdf> (cf. p. 27, 29, 36, 41, 43).
- [59] Mark Jason HARRIS et Anselmo LASTRA. Real-Time Cloud Rendering. *Computer Graphics Forum* 20.3 (2001), 76-85. DOI : [10.1111/1467-8659.00500](https://doi.org/10.1111/1467-8659.00500). URL : [http://www.markmark.net/PDFs/RTClouds\\_HarrisEG2001.pdf](http://www.markmark.net/PDFs/RTClouds_HarrisEG2001.pdf) (cf. p. 40, 43).
- [60] Mark Jason HARRIS *et al.* Simulation of Cloud Dynamics on Graphics Hardware. *SIGGRAPH/Eurographics Conference on Graphics Hardware*. 2003, p. 92-101. ISBN : 1-58113-739-7. URL : <http://www.markmark.net/cloudsim/harrisGH2003.pdf> (cf. p. 47, 61, 63).
- [61] L. G. HENYAY et J. L. GREENSTEIN. Diffuse radiation in the Galaxy. *The Astrophysical Journal* 93 (jan. 1941), 70. DOI : [10.1086/144246](https://doi.org/10.1086/144246). URL : [http://articles.adsabs.harvard.edu/cgi-bin/nph-iarticle\\_query?db\\_key=AST&bibcode=1941ApJ...93...70H&letter=.&classic=YES&defaultprint=YES&whole\\_paper=YES&page=70&epage=70&send=Send+PDF&filetype=.pdf](http://articles.adsabs.harvard.edu/cgi-bin/nph-iarticle_query?db_key=AST&bibcode=1941ApJ...93...70H&letter=.&classic=YES&defaultprint=YES&whole_paper=YES&page=70&epage=70&send=Send+PDF&filetype=.pdf) (cf. p. 137).

- [62] Sung Min HONG et Gladimir Valerio Guimaraes BARANOSKI. *A Study on Atmospheric Halo Visualization*. Rapp. tech. University of Waterloo, sept. 2003. URL : <http://www.npsg.uwaterloo.ca/resources/docs/tr/halo-report.pdf> (cf. p. 25).
- [63] Lukas HOSEK et Alexander WILKIE. An Analytic Model for Full Spectral Sky-Dome Radiance. *ACM Transactions on Graphics* 3.4 (juil. 2012). DOI : 10.1145/2185520.2185591. URL : [http://cgg.mff.cuni.cz/projects/SkylightModelling/HosekWilkie\\_SkylightModel\\_SIGGRAPH2012\\_Preprint.pdf](http://cgg.mff.cuni.cz/projects/SkylightModelling/HosekWilkie_SkylightModel_SIGGRAPH2012_Preprint.pdf) (cf. p. 34, 43).
- [64] Przemyslaw IDZKIEWICZ. © ⓘ ⓘ Creative Commons - Attribution-ShareAlike 2.0. 15 juil. 2005. URL : [https://upload.wikimedia.org/wikipedia/commons/0/0b/Cirrus\\_over\\_Warsaw%2C\\_June\\_26%2C\\_2005.jpg](https://upload.wikimedia.org/wikipedia/commons/0/0b/Cirrus_over_Warsaw%2C_June_26%2C_2005.jpg) (cf. p. 88).
- [65] Brocken INAGLORY. © ⓘ ⓘ Creative Commons - Attribution-ShareAlike 4.0. 13 juil. 2007. URL : [https://upload.wikimedia.org/wikipedia/commons/6/60/Fogbow\\_glory\\_spectre\\_bridge\\_edit\\_1.jpg](https://upload.wikimedia.org/wikipedia/commons/6/60/Fogbow_glory_spectre_bridge_edit_1.jpg) (cf. p. 25).
- [66] Brocken INAGLORY. © ⓘ ⓘ Creative Commons - Attribution-ShareAlike 3.0. 18 mai 2007. URL : [https://upload.wikimedia.org/wikipedia/commons/7/7b/Hot\\_road\\_mirage.jpg](https://upload.wikimedia.org/wikipedia/commons/7/7b/Hot_road_mirage.jpg) (cf. p. 27).
- [67] Brocken INAGLORY. © ⓘ ⓘ Creative Commons - Attribution-ShareAlike 3.0. 11 jan. 2009. URL : [https://upload.wikimedia.org/wikipedia/commons/2/2d/Sunset\\_Mirage\\_and\\_green\\_blue\\_flashes\\_1-11-09.jpg](https://upload.wikimedia.org/wikipedia/commons/2/2d/Sunset_Mirage_and_green_blue_flashes_1-11-09.jpg) (cf. p. 27).
- [68] Masa INAKAGE. An Illumination Model for Atmospheric Environments. *Computer Graphics International*. 1989, p. 533-548. ISBN : 978-4-431-68095-6. DOI : 10.1007/978-4-431-68093-2\_34 (cf. p. 24, 25).
- [69] Tomokazu ISHIKAWA *et al.* Modeling of Aurora Borealis Using the Observed Data. *Spring Conference on Computer Graphics*. 2011, p. 13-16. ISBN : 978-1-4503-1978-2. DOI : 10.1145/2461217.2461220. URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.228.4680&rep=rep1&type=pdf> (cf. p. 21).
- [70] Kei IWASAKI, Yoshinori DOBASHI et Makoto OKABE. Example-based Synthesis of Three-dimensional Clouds from Photographs. *Computer Graphics International*. 2017. ISBN : 978-1-4503-5228-4. DOI : 10.1145/3095140.3095168 (cf. p. 55, 63).
- [71] Dietmar JACKËL et Bertram WALTER. Modelling and Rendering of the Atmosphere Using Mie-Scattering. *Computer Graphics Forum* 16.4 (1997), 201-210. DOI : 10.1111/1467-8659.00180 (cf. p. 24, 25).
- [72] Xiaogang JIN *et al.* Blob-based liquid morphing. *Computer Animation and Virtual Worlds* 16.3-4 (2005), 391-403. DOI : 10.1002/cav.84. URL : <http://homepage.tudelft.nl/h05k3/pubs/CASABlobby.pdf> (cf. p. 129).
- [73] James T. KAJIYA. The Rendering Equation. *SIGGRAPH Computer Graphics* 20.4 (août 1986), 143-150. DOI : 10.1145/15886.15902. URL : [http://www.cse.chalmers.se/edu/year/2011/course/TDA361/2007/rend\\_eq.pdf](http://www.cse.chalmers.se/edu/year/2011/course/TDA361/2007/rend_eq.pdf) (cf. p. 37).
- [74] James T. KAJIYA et Brian von HERZEN. Ray Tracing Volume Densities. *SIGGRAPH Computer Graphics* 18.3 (jan. 1984), 165-174. DOI : 10.1145/964965.808594. URL : [https://www.researchgate.net/profile/Brian\\_Von\\_Herzen3/publication/234800181\\_Ray\\_Tracing\\_Volume\\_Densities/links/0deec5340c9062b99a000000/Ray-Tracing-Volume-Densities.pdf](https://www.researchgate.net/profile/Brian_Von_Herzen3/publication/234800181_Ray_Tracing_Volume_Densities/links/0deec5340c9062b99a000000/Ray-Tracing-Volume-Densities.pdf) (cf. p. 31, 36, 37, 43, 46, 63).


- [75] Simon KALLWEIT *et al.* Deep Scattering : Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks. *ACM Transactions on Graphics* (2017). DOI : [10.1145/3130800.3130880](https://doi.org/10.1145/3130800.3130880). URL : <http://drz.disneyresearch.com/~jnovak/publications/DeepScattering/DeepScattering.pdf> (cf. p. 39, 43).
- [76] Kazufumi KANEDA *et al.* Photorealistic image synthesis for outdoor scenery under various atmospheric conditions. *The Visual Computer* 7.5 (1991), 248-258. DOI : [10.1007/BF01905690](https://doi.org/10.1007/BF01905690) (cf. p. 29, 34, 43).
- [77] Theodore KIM et Ming C. LIN. Fast Animation of Lightning Using an Adaptive Mesh. *IEEE Transactions on Visualization and Computer Graphics* 13.2 (mar. 2007), 390-402. DOI : [10.1109/TVCG.2007.38](https://doi.org/10.1109/TVCG.2007.38). URL : [http://gamma.cs.unc.edu/FAST\\_LIGHTNING/lightning\\_tvcg\\_2007.pdf](http://gamma.cs.unc.edu/FAST_LIGHTNING/lightning_tvcg_2007.pdf) (cf. p. 20).
- [78] Michal KLAJBAN.  Creative Commons - Attribution-ShareAlike 4.0. 22 sept. 2018. URL : [https://upload.wikimedia.org/wikipedia/commons/d/da/Rain\\_over\\_Beinn\\_Eich%2C\\_Luss\\_Hills%2C\\_Scotland.jpg](https://upload.wikimedia.org/wikipedia/commons/d/da/Rain_over_Beinn_Eich%2C_Luss_Hills%2C_Scotland.jpg) (cf. p. 22).
- [79] Robert Victor KLASSEN. Modeling the Effect of the Atmosphere on Light. *ACM Transactions on Graphics* 6.3 (juil. 1987), 215-237. DOI : [10.1145/35068.35071](https://doi.org/10.1145/35068.35071) (cf. p. 29, 31, 32, 34, 43).
- [80] Joe KNISS, Gordon KINDLMANN et Charles HANSEN. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 8.3 (juil. 2002), 270-285. DOI : [10.1109/TVCG.2002.1021579](https://doi.org/10.1109/TVCG.2002.1021579) (cf. p. 32).
- [81] Joe KNISS *et al.* A Model for Volume Lighting and Modeling. *IEEE Transactions on Visualization and Computer Graphics* 9.2 (avr. 2003), 150-162. DOI : [10.1109/TVCG.2003.1196003](https://doi.org/10.1109/TVCG.2003.1196003). URL : <https://www.cs.utah.edu/~shirley/papers/volshade.pdf> (cf. p. 32, 43).
- [82] Peter KUTZ *et al.* Spectral and Decomposition Tracking for Rendering Heterogeneous Volumes. *ACM Transactions on Graphics* (2017). DOI : [10.1145/3072959.3073665](https://doi.org/10.1145/3072959.3073665). URL : <http://drz.disneyresearch.com/~jnovak/publications/SDTracking/SDTracking.pdf> (cf. p. 38, 43, 135).
- [83] Felipe Micaroni LALLI.  Creative Commons - Attribution-ShareAlike 2.5. 15 fév. 2004. URL : [https://upload.wikimedia.org/wikipedia/commons/d/d7/Campinas\\_Estrada\\_040.jpg](https://upload.wikimedia.org/wikipedia/commons/d/d7/Campinas_Estrada_040.jpg) (cf. p. 24).
- [84] M. S. LANGER *et al.* A spectral-particle hybrid method for rendering falling snow. *Eurographics Symposium on Rendering*. 2004. ISBN : 3-905673-12-6. DOI : [10.2312/EGWR/EGSR04/217-226](https://doi.org/10.2312/EGWR/EGSR04/217-226). URL : [http://www.cim.mcgill.ca/~langer/MY\\_PAPERS/Langer-EGSR04.pdf](http://www.cim.mcgill.ca/~langer/MY_PAPERS/Langer-EGSR04.pdf) (cf. p. 22, 23).
- [85] Orion Sky LAWLOR et Jon D. GENETTI. Interactive Volume Rendering Aurora on the GPU. *Journal of WSCG* 19.1 (2011), 25-32. URL : <https://dspace5.zcu.cz/bitstream/11025/1242/1/Lawlor.pdf> (cf. p. 21).
- [86] J. P. LEWIS. Algorithms for Solid Noise Synthesis. *SIGGRAPH Computer Graphics* 23.3 (1989), 263-270. DOI : [10.1145/74334.74360](https://doi.org/10.1145/74334.74360). URL : <http://www.scribblethink.org/Work/Crunge/crunge.pdf> (cf. p. 57).
- [87] Jonathan M. LILLY *et al.* Fractional Brownian motion, the Matérn process, and stochastic modeling of turbulent dispersion. *Nonlinear Processes in Geophysics* 24.3 (21 août 2017), 481-514. DOI : [10.5194/npg-24-481-2017](https://doi.org/10.5194/npg-24-481-2017). URL : <https://arxiv.org/pdf/1605.01684.pdf> (cf. p. 102).

- [88] Andrei LINTU, Jörg HABER et Marcus MAGNOR. Realistic Solar Disc Rendering. *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*. 2005, p. 79-86. ISBN : 80-903100-7-9. URL : [http://halma.mpi-inf.mpg.de/intranet/nwg3/nwg3publ.nsf/0/d37aeff0f73a82e4c1256fa100627318/\\$FILE/wscg05.pdf](http://halma.mpi-inf.mpg.de/intranet/nwg3/nwg3publ.nsf/0/d37aeff0f73a82e4c1256fa100627318/$FILE/wscg05.pdf) (cf. p. 28).
- [89] Shengjun LIU, Xiaogang JIN et Charlie Chang Ling WANG. Target Shape Controlled Cloud Animation. *Computer Graphics International*. 2006, p. 578-585. ISBN : 978-3-540-35639-4. DOI : [10.1007/11784203\\_54](https://doi.org/10.1007/11784203_54) (cf. p. 60, 63, 129).
- [90] Shengjun LIU *et al.* Ellipsoidal-blob Approximation of 3D Models and Its Applications. *Computers & Graphics* 31.2 (avr. 2007), 243-251. DOI : [10.1016/j.cag.2006.12.004](https://doi.org/10.1016/j.cag.2006.12.004). URL : <https://pdfs.semanticscholar.org/1c2c/2d2a16e706311f761c924918ed2cd2d660c1.pdf> (cf. p. 129).
- [91] William E. LORENSEN et Harvey E. CLINE. Marching Cubes : A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Computer Graphics* 21.4 (juil. 1987), 163-169. DOI : [10.1145/37402.37422](https://doi.org/10.1145/37402.37422) (cf. p. 59).
- [92] Shaun LOVEJOY. Area-Perimeter Relation for Rain and Cloud Areas. *Science* 216.4542 (1982), 185-187. DOI : [10.1126/science.216.4542.185](https://doi.org/10.1126/science.216.4542.185). URL : <http://www.physics.mcgill.ca/~gang/eprints/eprintLovejoy/neweprint/Lovejoy.Science.1982.pdf> (cf. p. 57).
- [93] Miles MACKLIN et Matthias MÜLLER. Position Based Fluids. *ACM Transactions on Graphics* 32.4 (2013). URL : [http://mmacklin.com/pbf\\_sig\\_preprint.pdf](http://mmacklin.com/pbf_sig_preprint.pdf) (cf. p. 50).
- [94] Nelson L. MAX. Atmospheric Illumination and Shadows. *SIGGRAPH Computer Graphics* 20.4 (1986), 117-124. DOI : [10.1145/15886.15899](https://doi.org/10.1145/15886.15899) (cf. p. 26).
- [95] Nelson L. MAX. Light diffusion through clouds and haze. *Computer Vision, Graphics, and Image Processing* 33.3 (1986), 280-292. DOI : [10.1016/0734-189X\(86\)90177-5](https://doi.org/10.1016/0734-189X(86)90177-5) (cf. p. 31, 43, 57, 63).
- [96] Nelson L. MAX. Efficient Light Propagation for Multiple Anisotropic Volume Scattering. *Eurographics Workshop on Rendering*. 1994, p. 87-104. ISBN : 978-3-642-87825-1. DOI : [10.1007/978-3-642-87825-1\\_7](https://doi.org/10.1007/978-3-642-87825-1_7). URL : <https://pdfs.semanticscholar.org/9d41/5b137234d6edb84f522f6138ec5219d3de4f.pdf> (cf. p. 36, 37, 43).
- [97] Nelson L. MAX, Roger CRAWFIS et Dean WILLIAMS. Visualizing Wind Velocities by Advecting Cloud Textures. *Conference on Visualization*. 1992, p. 179-184. ISBN : 0-8186-2896-0 (cf. p. 58, 61, 63).
- [98] Ryo MIYAZAKI, Yoshinori DOBASHI et Tomoyuki NISHITA. Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics. *Eurographics Short Presentations*. 2002, p. 405-410. URL : <https://pdfs.semanticscholar.org/f398/7e9e83fe0a969be249ef23ad187ec713dc94.pdf> (cf. p. 46, 47, 61, 63).
- [99] Ryo MIYAZAKI *et al.* A method for modeling clouds based on atmospheric fluid dynamics. *Pacific Conference on Computer Graphics and Applications*. 2001, p. 363-372. DOI : [10.1109/PCCGA.2001.962893](https://doi.org/10.1109/PCCGA.2001.962893) (cf. p. 49, 61, 63).
- [100] Aindrila MUKHOPADHYAY. © Creative Commons - Attribution 2.0. 8 avr. 2004. URL : [https://upload.wikimedia.org/wikipedia/commons/e/ea/Aindrila\\_-\\_lightning\\_-\\_jet\\_air\\_%28by%29.jpg](https://upload.wikimedia.org/wikipedia/commons/e/ea/Aindrila_-_lightning_-_jet_air_%28by%29.jpg) (cf. p. 20).



- [101] S. MURAKAMI et H. ICHIHARA. On a 3D Display Method by Metaball Technique. *Journal of papers given by at the Electronics Communication J70-D.8* (1987), 1607-1615 (cf. p. 71).
- [102] Forest Kenton MUSGRAVE. Prisms and Rainbows : a Dispersion Model for Computer Graphics. *Graphics Interface*. 1989, p. 227-234. DOI : 10.20380/GI1989.30. URL : <http://graphicsinterface.org/wp-content/uploads/gi1989-30.pdf> (cf. p. 24).
- [103] Forest Kenton MUSGRAVE. A Note on Ray Tracing Mirages. *IEEE Computer Graphics and Applications* 10.6 (1990), 10-12. DOI : 10.1109/38.62692 (cf. p. 28).
- [104] NATIONAL OCEANIC et ATMOSPHERIC ADMINISTRATION. Public-Domain. URL : <https://upload.wikimedia.org/wikipedia/commons/1/16/Hailstorm.jpg> (cf. p. 22).
- [105] Fabrice NEYRET. Qualitative Simulation of Convective Cloud Formation and Evolution. *Computer Animation and Simulation*. Sept. 1997, p. 113-124. ISBN : 978-3-211-83048-2. DOI : 10.1007/978-3-7091-6874-5\_8. URL : <http://evasion.imag.fr/Membres/Fabrice.Neyret/publis/EWAS97.pdf> (cf. p. 50, 63).
- [106] Shawn NICKEL. How to shoot Northern Lights; unique duty station offers unique photos. Public-Domain. 15 nov. 2014. URL : <https://media.defense.gov/2014/Nov/24/2000984512/-1/-1/0/141115-F-FT438-034.JPG> (cf. p. 21).
- [107] Tomoyuki NISHITA, Yoshinori DOBASHI et Eihachiro NAKAMAE. Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light. *SIGGRAPH*. 1996, p. 379-386. ISBN : 0-89791-746-4. DOI : 10.1145/237170.237277 (cf. p. 35, 43, 55, 59, 60, 63).
- [108] Tomoyuki NISHITA, Yasuhiro MIYAWAKI et Eihachiro NAKAMAE. A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources. *SIGGRAPH Computer Graphics* 21.4 (1987), 303-310. DOI : 10.1145/37402.37437 (cf. p. 26).
- [109] Tomoyuki NISHITA et Eihachiro NAKAMAE. Continuous Tone Representation of Three-dimensional Objects Illuminated by Sky Light. *SIGGRAPH Computer Graphics* 20.4 (1986), 125-132. DOI : 10.1145/15886.15900 (cf. p. 33, 34, 43).
- [110] Tomoyuki NISHITA et al. Display of the Earth Taking into Account Atmospheric Scattering. *SIGGRAPH*. 1993, p. 175-182. ISBN : 0-89791-601-8. DOI : 10.1145/166117.166140. URL : [http://nishitalab.org/user/nis/cdrom/sig93\\_nis.pdf](http://nishitalab.org/user/nis/cdrom/sig93_nis.pdf) (cf. p. 29, 32, 35, 37, 41, 43, 142).
- [111] Tomoyuki NISHITA et al. Display Method of the Sky Color Taking into Account Multiple Scattering. *Pacific Conference on Computer Graphics and Applications*. 1996, p. 117-132. URL : <http://nishitalab.org/user/nis/cdrom/skymul.pdf> (cf. p. 34, 37, 43).
- [112] Jan NOVÁK, Andrew SELLE et Wojciech JAROSZ. Residual Ratio Tracking for Estimating Attenuation in Participating Media. *ACM Transactions on Graphics* 33.6 (nov. 2014). DOI : 10.1145/2661229.2661292. URL : <https://cs.dartmouth.edu/~wjarosz/publications/novak14residual.pdf> (cf. p. 38, 43, 135, 143).
- [113] Jan NOVÁK et al. Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum* 37.2 (2018). URL : <https://cs.dartmouth.edu/~wjarosz/publications/novak18monte.pdf> (cf. p. 135).



- [114] Derek OVERBY, Zeki MELEK et John KEYSER. Interactive Physically-Based Cloud Simulation. *Pacific Conference on Computer Graphics and Applications*. Oct. 2002. ISBN : 0-7695-1784-6. DOI : [10.1109/PCCGA.2002.1167904](https://doi.org/10.1109/PCCGA.2002.1167904). URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.574.3983&rep=rep1&type=pdf> (cf. p. 47, 61, 63).
- [115] Carlos Jiménez de PARGA et Sebastián Rubén Gómez PALOMO. Efficient Algorithms for Real-Time GPU Volumetric Cloud Rendering with Enhanced Geometry. *Symmetry* 10.4 (2018). DOI : [10.3390/sym10040125](https://doi.org/10.3390/sym10040125). URL : <https://www.mdpi.com/2073-8994/10/4/125/pdf> (cf. p. 61, 63).
- [116] Darwyn R. PEACHEY. Solid Texturing of Complex Surfaces. *SIGGRAPH Computer Graphics* 19.3 (juil. 1985), 279-286. DOI : [10.1145/325165.325246](https://doi.org/10.1145/325165.325246) (cf. p. 56).
- [117] Richard PEREZ, R. SEALS et Joseph J. MICHALSKY. All-weather model for sky luminance distribution—Preliminary configuration and validation. *Solar Energy* 50.3 (1993), 235-245. DOI : [10.1016/0038-092X\(93\)90017-I](https://doi.org/10.1016/0038-092X(93)90017-I) (cf. p. 34, 43).
- [118] Ken PERLIN. An Image Synthesizer. *SIGGRAPH Computer Graphics* 19.3 (juil. 1985), 287-296. DOI : [10.1145/325165.325247](https://doi.org/10.1145/325165.325247). URL : <http://www.cs.jhu.edu/~misha/Fall17/Readings/Perlin85.pdf> (cf. p. 56, 83).
- [119] Ken PERLIN et Eric M. HOFFERT. Hypertexture. *SIGGRAPH Computer Graphics* (juil. 1989), 253-262. DOI : [10.1145/74334.74359](https://doi.org/10.1145/74334.74359). URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.438.4926&rep=rep1&type=pdf> (cf. p. 57).
- [120] Arcot J. PREETHAM, Peter SHIRLEY et Brian SMITS. A Practical Analytic Model for Daylight. *SIGGRAPH*. 1999, p. 91-100. ISBN : 0-201-48560-5. DOI : [10.1145/311535.311545](https://doi.org/10.1145/311535.311545). URL : [https://www.cs.duke.edu/courses/cps124/spring08/assign/07\\_papers/p91-preetham.pdf](https://www.cs.duke.edu/courses/cps124/spring08/assign/07_papers/p91-preetham.pdf) (cf. p. 34, 43).
- [121] Manoj Kumar RACHERLA.  Creative Commons - Attribution-ShareAlike. 30 avr. 2013. URL : <https://upload.wikimedia.org/wikipedia/commons/d/dd/Godrays.jpg> (cf. p. 26).
- [122] Kirk RILEY *et al.* Visually Accurate Multi-Field Weather Visualization. *IEEE Visualization*. 2003. ISBN : 0-7695-2030-8. DOI : [10.1109/VISUAL.2003.1250383](https://doi.org/10.1109/VISUAL.2003.1250383) (cf. p. 33, 43).
- [123] Kirk RILEY *et al.* A System For Realistic Weather Rendering. *American Meteorological Society Annual Meeting*. Jan. 2004. URL : <https://ams.confex.com/ams/pdfpapers/73147.pdf> (cf. p. 33).
- [124] Kirk RILEY *et al.* Efficient Rendering of Atmospheric Phenomena. *Eurographics Symposium on Rendering*. 2004, p. 375-386. ISBN : 3-905673-12-6. DOI : [10.2312/EGWR/EGSR04/375-386](https://doi.org/10.2312/EGWR/EGSR04/375-386). URL : [http://wwwvis.informatik.uni-stuttgart.de/~kraus/preprints/egsr2004\\_riley.pdf](http://wwwvis.informatik.uni-stuttgart.de/~kraus/preprints/egsr2004_riley.pdf) (cf. p. 24-26, 33, 43).
- [125] Holly E. RUSHMEIER et Kenneth E. TORRANCE. The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium. *SIGGRAPH Computer Graphics* (1987), 293-302. DOI : [10.1145/37401.37436](https://doi.org/10.1145/37401.37436) (cf. p. 35, 36, 43).
- [126] Iman SADEGHI *et al.* Physically-based Simulation of Rainbows. *ACM Transactions on Graphics* 31.1 (fév. 2012). DOI : [10.1145/2077341.2077344](https://doi.org/10.1145/2077341.2077344). URL : <http://www.cs.dartmouth.edu/~wjarosz/publications/sadeghi11physically.pdf> (cf. p. 25).
- [127] Georgios SAKAS. Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer* 9.4 (1993), 200-212. DOI : [10.1007/BF01901724](https://doi.org/10.1007/BF01901724) (cf. p. 58).

- [128] Georgios SAKAS et Rüdiger WESTERMANN. A Functional Approach to the Visual Simulation of Gaseous Turbulence. *Computer Graphics Forum* 11.3 (mai 1992), 107-117. DOI : [10.1111/1467-8659.1130107](https://doi.org/10.1111/1467-8659.1130107) (cf. p. 58, 63, 97).
- [129] Yoann SALLAZ-DAMAZ.  Creative Commons - Attribution-ShareAlike 3.0. 28 jan. 2008. URL : <https://upload.wikimedia.org/wikipedia/commons/f/f8/MirageAntarctique.jpg> (cf. p. 27).
- [130] Joshua SCHPOK *et al.* A Real-Time Cloud Modeling, Rendering, and Animation System. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2003, p. 160-166. ISBN : 1-58113-659-5. URL : <https://www.sci.utah.edu/publications/schpok03/scacclouds.pdf> (cf. p. 52, 59-61, 63, 129).
- [131] Francisco José SERÓN *et al.* Implementation of a method of curved raytracing for inhomogeneous atmospheres. *Computers & Graphics* 29.1 (2005), 95-108. DOI : [10.1016/j.cag.2004.11.010](https://doi.org/10.1016/j.cag.2004.11.010) (cf. p. 28).
- [132] Jaroslav SLOUP. A Survey of the Modelling and Rendering of the Earth's Atmosphere. *Spring Conference on Computer Graphics*. 2002, p. 141-150. ISBN : 1-58113-608-0. DOI : [10.1145/584458.584482](https://doi.org/10.1145/584458.584482) (cf. p. 136).
- [133] Jaroslav SLOUP. Visual Simulation of Refraction Phenomena in the Earth's Atmosphere. *International Conference on Information Visualization*. 2003, p. 452. ISBN : 0-7695-1988-1. DOI : [10.1109/IV.2003.1218024](https://doi.org/10.1109/IV.2003.1218024) (cf. p. 27, 28).
- [134] V. V. SURKOV et M. HAYAKAWA. Underlying mechanisms of transient luminous events : a review. *Annales Geophysicae* 30 (2012), 1185-1212. DOI : [10.5194/angeo-30-1185-2012](https://doi.org/10.5194/angeo-30-1185-2012). URL : <https://www.ann-geophys.net/30/1185/2012/angeo-30-1185-2012.pdf> (cf. p. 20).
- [135] Natalya TATARCHUK. Artist-directable Real-time Rain Rendering in City Environments. *SIGGRAPH Courses*. 2006, p. 23-64. ISBN : 1-59593-364-6. DOI : [10.1145/1185657.1185828](https://doi.org/10.1145/1185657.1185828) (cf. p. 23).
- [136] TENBERGEN.  Creative Commons - Attribution-ShareAlike 4.0. 1<sup>er</sup> fév. 2018. URL : [https://upload.wikimedia.org/wikipedia/commons/5/51/20180201\\_Winnipeg\\_Sundog\\_and\\_halo.jpg](https://upload.wikimedia.org/wikipedia/commons/5/51/20180201_Winnipeg_Sundog_and_halo.jpg) (cf. p. 25).
- [137] Lars TIEDE.  Creative Commons - Attribution-ShareAlike 2.0. 23 jan. 2011. URL : [https://upload.wikimedia.org/wikipedia/commons/d/d1/Northern\\_lights\\_on\\_Kval%C3%B8ya\\_2012-01-23.jpg](https://upload.wikimedia.org/wikipedia/commons/d/d1/Northern_lights_on_Kval%C3%B8ya_2012-01-23.jpg) (cf. p. 21).
- [138] Eric VEACH. Robust Monte Carlo Methods for Light Transport Simulation. Thèse de doct. Stanford University, 1997. URL : [http://graphics.stanford.edu/papers/veach\\_thesis/thesis.pdf](http://graphics.stanford.edu/papers/veach_thesis/thesis.pdf) (cf. p. 139).
- [139] Antoine WEBANCK. All rights reserved. 15 juil. 2006 (cf. p. 26).
- [140] Antoine WEBANCK. All rights reserved. 25 mai 2009 (cf. p. 20).
- [141] Antoine WEBANCK. All rights reserved. 1<sup>er</sup> jan. 2018 (cf. p. 22).
- [142] Antoine WEBANCK *et al.* Procedural Cloudscapes. *Computer Graphics Forum* 37.2 (mai 2018), 431-442. DOI : [10.1111/cgf.13373](https://doi.org/10.1111/cgf.13373). URL : <https://hal.archives-ouvertes.fr/hal-01730789/file/Procedural-Cloudscapes-EG2018.pdf> (cf. p. 104).
- [143] Yoann WEBER *et al.* A Multiscale Model for Rain Rendering in Real-time. *Computers & Graphics* 50.C (août 2015), 61-70. DOI : [10.1016/j.cag.2015.04.007](https://doi.org/10.1016/j.cag.2015.04.007). URL : [http://www.unilim.fr/pages\\_perso/guillaume.gilet/publications/pdf/CG2015.pdf](http://www.unilim.fr/pages_perso/guillaume.gilet/publications/pdf/CG2015.pdf) (cf. p. 22, 23).

- [144] Alexander WILKIE et Lukas HOSEK. Predicting Sky Dome Appearance on Earth-like Extrasolar Worlds. *Spring Conference on Computer Graphics*. Mai 2013, p. 145-152. ISBN : 978-1-4503-2480-9. DOI : [10.1145/2508244.2508263](https://doi.org/10.1145/2508244.2508263). URL : [http://cgg.mff.cuni.cz/projects/SkylightModelling/sccg\\_2013\\_alien\\_sun\\_preprint.pdf](http://cgg.mff.cuni.cz/projects/SkylightModelling/sccg_2013_alien_sun_preprint.pdf) (cf. p. 35, 43).
- [145] Jamie WITHER, Antoine BOUTHORS et Marie-Paule CANI. Rapid sketch modeling of clouds. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. 2008, p. 113-118. ISBN : 978-3-905674-07-1. DOI : [10.2312/SBM/SBM08/113-118](https://doi.org/10.2312/SBM/SBM08/113-118). URL : [http://evasion.imag.fr/Publications/2008/WBC08/rapid\\_sketch\\_modeling\\_of\\_clouds.pdf](http://evasion.imag.fr/Publications/2008/WBC08/rapid_sketch_modeling_of_clouds.pdf) (cf. p. 53, 63).
- [146] WORLD METEOROLOGICAL ORGANIZATION. *International Cloud Atlas*. T. I. 1975. ISBN : 978-9263104076. URL : [http://library.wmo.int/pmb\\_ged/wmo\\_407\\_en-v1.pdf](http://library.wmo.int/pmb_ged/wmo_407_en-v1.pdf) (cf. p. 87).
- [147] WORLD METEOROLOGICAL ORGANIZATION. *International Cloud Atlas*. T. II. 1987. ISBN : 92-63-12407-8. URL : [http://library.wmo.int/pmb\\_ged/wmo\\_407\\_en-v2.pdf](http://library.wmo.int/pmb_ged/wmo_407_en-v2.pdf) (cf. p. 87).
- [148] WORLD METEOROLOGICAL ORGANIZATION. *International Cloud Atlas*. 2017. URL : <https://cloudatlas.wmo.int> (cf. p. 87).
- [149] Brian WYVILL, Andrew GUY et Eric GALIN. Extending The CSG Tree. Warping, Blending and Boolean Operations in an Implicit Surface Modeling System. *Computer Graphics Forum* 18.2 (juin 1999), 149-158. DOI : [10.1111/1467-8659.00365](https://doi.org/10.1111/1467-8659.00365). URL : <https://pages.cpsc.ucalgary.ca/~blob/ps/blobtree.pdf> (cf. p. 65, 66).
- [150] Brian WYVILL et Geoff WYVILL. Field functions for implicit surfaces. *The Visual Computer* 5.1 (jan. 1989), 75-82. DOI : [10.1007/BF01901483](https://doi.org/10.1007/BF01901483) (cf. p. 68).
- [151] Geoff WYVILL, Craig MCPHEETERS et Brian WYVILL. Data structure for soft objects. *The Visual Computer* 2.4 (août 1986), 227-234. DOI : [10.1007/BF01900346](https://doi.org/10.1007/BF01900346). URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.881.7066&rep=rep1&type=pdf> (cf. p. 49, 71).
- [152] Chunqiang YUAN et Jing Guo. An efficient framework for modeling clouds from Landsat8 images. *International Conference on Graphic and Image Processing*. T. 9443. 2015. DOI : [10.1117/12.2179444](https://doi.org/10.1117/12.2179444) (cf. p. 55, 63).
- [153] Chunqiang YUAN *et al.* Modeling Large Scale Clouds from Satellite Images. *Pacific Conference on Computer Graphics and Applications*. 2013. ISBN : 978-3-905674-50-7. DOI : [10.2312/PE.PG.PG2013short.047-052](https://doi.org/10.2312/PE.PG.PG2013short.047-052) (cf. p. 55).
- [154] Chunqiang YUAN *et al.* Modelling Cumulus Cloud Shape from a Single Image. *Computer Graphics Forum* 33.6 (2014), 288-297. DOI : [10.1111/cgf.12350](https://doi.org/10.1111/cgf.12350) (cf. p. 55, 56).
- [155] Egor Yusov. High-performance Rendering of Realistic Cumulus Clouds Using Pre-computed Lighting. *High Performance Graphics*. 2014, p. 127-136. DOI : [10.2312/hpg.20141101](https://doi.org/10.2312/hpg.20141101). URL : <https://pdfs.semanticscholar.org/0a58/7b387059d66ef1ec2a78ab43fd734ae124.pdf> (cf. p. 42, 43).
- [156] Yiming ZHANG, Zili ZHANG et Jiayue Hou andx XIAOHUI Hou. Estimating Cumulus Cloud Shape from a Single Image. *Advances in Image and Graphics Technologies*. 2018. ISBN : 978-981-10-7389-2. DOI : [10.1007/978-981-10-7389-2\\_15](https://doi.org/10.1007/978-981-10-7389-2_15) (cf. p. 55).

- [157] Zili ZHANG *et al.* Modeling Cumulus Cloud Scenes from High-resolution Satellite Images. *Computer Graphics Forum* 36.7 (2017), 229-238. DOI : [10.1111/cgf.13288](https://doi.org/10.1111/cgf.13288) (cf. p. [55](#), [63](#)).