

UNIVERSITÉ DE MONTRÉAL

**VISUALISATION D'INCENDIE BASÉE SUR
SIMULATION NUMÉRIQUE**

DAVID DUBÉ

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
DÉCEMBRE 2008



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-48914-7

Our file Notre référence

ISBN: 978-0-494-48914-7

NOTICE:

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

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

AVIS:

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

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

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

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

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

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

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

VISUALISATION D'INCENDIE BASÉE SUR SIMULATION NUMÉRIQUE

présenté par : DUBÉ David

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

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

Mme CHERIET Farida, Ph.D., présidente

M. OZELL Benoit, Ph.D., membre et directeur de recherche

M. REGGIO Marcelo, Ph.D., membre et codirecteur

M. CAMARERO Ricardo, Ph.D., membre

*Pour ma famille, qui me répète sans cesse
qu'il ne faut jamais abandonner.*

REMERCIEMENTS

J'aimerais remercier mon directeur de recherche pour sa patience sans borne et son soutien dans l'élaboration de ce mémoire de recherche, ma famille qui n'a jamais cessé de me pousser à terminer l'écriture du mémoire et ma petite amie pour ses rappels à l'ordre lorsque je décidais de ne pas écrire. Finalement je remercie mon grand-père pour m'avoir rappeler que la vie est précieuse.

RÉSUMÉ

La modélisation et la visualisation par ordinateur sont des outils essentiels pour comprendre l'évolution et le comportement des incendies. Cette recherche propose une nouvelle méthode hybride conciliant les techniques numériques et approximatives pour la visualisation de la fumée et de la chaleur d'une simulation d'incendie.

Les données de simulation, produites par le simulateur *Fire Dynamic Simulator* (FDS), sont d'abord traitées pour être utilisées en nuanceurs. Une technique à base de plans de coupe sert de squelette pour la visualisation. La fumée est visualisée sur les plans de coupe grâce à une fonction de transfert unidimensionnelle. Une technique basée sur la variance des indices de réfraction des points du volume de données permet de construire l'effet de chaleur. La variance est combinée à une texture gaussienne qui est appliquée dans l'espace écran.

Le moteur 3D Phoenix développé sur mesure réalise les deux effets d'incendie. Une représentation 3D de la scène de simulation est importée et combinée aux effets. La capacité de se déplacer à la première personne et d'obtenir des résultats animés contribuent à la qualité des résultats observés ainsi qu'au sentiment d'immersion. L'intégration d'une scène 3D permet de mettre les résultats visualisés en contexte et d'avoir une visualisation de simulation complète.

ABSTRACT

Modeling and visualization tools are essentials in understanding the evolution and behaviour of fire simulation. This research proposes a new method based on the strengths of both numerical and artificial visualization techniques for the rendering of smoke and heat shimmering on GPU.

Simulations data, produced by the simulator Fire Dynamic Simulator (FDS), enter a processing stage before graphic cards shaders can use them. A slicing plane technique is then used as the skeleton for visualization. Smoke is visualized on the slicing planes using a one-dimensional transfer function. The heat shimmering is based on the computation of the per-pixel variance of refraction indices at each point in the data volume. The variance values are then combined to a Gaussian noise texture for a post-processing effect applied in screen space.

A 3D engine codenamed Phoenix was developed specifically for the rendering of the smoke and heat shimmering effect. A 3D representation of the simulation scene is used and combined to the effects. The resulting simulation is visualized from a first person perspective that contributes greatly to the quality of the observed results and the immersion factor.

TABLE DES MATIÈRES

REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX.....	xi
LISTE DES FIGURES.....	xii
LISTE DES SIGLES ET ABRÉVIATIONS	xv
CHAPITRE 1 - INTRODUCTION	1
1.1 Contexte	1
1.2 Modéliser sans se brûler.....	2
1.3 Problématique et objectifs.....	3
1.4 Aperçu des différents chapitres.....	6
CHAPITRE 2 - TRAVAUX ANTÉRIEURS	7
2.1 Fumée.....	7
2.1.1 Simulations physiquement réalistes basées sur la MFD	7
2.1.2 Simulations temps réel	10
2.2 Mirages et distorsions	13
2.2.1 Phénomène physique.....	13
2.2.2 Méthodes ad-hoc	14
2.2.3 Méthodes basées sur le lancer de rayons.....	16
2.3 Feu.....	18
2.3.1 Approches par acquisitions de données expérimentales	18
2.3.2 Approches par simulations numériques	19
2.3.3 Approches approximatives.....	23
CHAPITRE 3 - FIRE DYNAMIC SIMULATOR (FDS).....	25
3.1 Modèle de combustion	25

3.2 Limites du modèle et robustesse	26
3.3 Spécification de la simulation et données	27
CHAPITRE 4 - VISUALISATION DE LA FUMÉE	29
4.1 Élaboration de la méthodologie de visualisation	29
4.2 Algorithme et implémentation	32
4.2.1 Prétraitement des valeurs de densité	32
4.2.2 Génération des plans de coupe	35
4.2.2.1 Préparation	37
4.2.2.2 Détermination de la distance minimale et maximale de la boîte englobante	38
4.2.2.3 Optimisations d'usage	38
4.2.2.4 Génération des plans de coupe	39
4.3 Caractérisation de l'illumination et fonction de transfert	43
4.4 Rendu et animations	46
4.4.1 Traitement des nuanceurs	46
4.4.2 Animations	49
CHAPITRE 5 - DISTORSION VISUELLE DUE À LA CHALEUR INTENSE	50
5.1 Élaboration de la méthodologie de visualisation	50
5.2 Hypothèses simplificatrices	53
5.3 Quantification de l'impact sur la lumière	54
5.3 Algorithme et implémentation CPU	58
5.3.1 Prétraitement des valeurs de températures	58
5.3.2 Échantillonnage du volume de données	59
5.4 Rendu, composition et animations	60
5.4.1 Première passe : calculs de $\Sigma(\Delta E_N)$ et $\Sigma(\Delta E_N)^2$	61
5.4.2 Deuxième passe : bruitage et composition	63
5.4.3 Animations	67
CHAPITRE 6 - RÉSULTATS	68

6.1 Moteur de rendu PHX	68
6.2 Plateforme et simulations	70
6.3 Plans de coupe	73
6.4 Effets de fumée	75
6.5 Effets de distorsion de chaleur	82
CHAPITRE 7 - DISCUSSION	89
7.1 Résultats	89
7.2 Paramétrisation des effets	92
7.3 Améliorations et perspective	95
CONCLUSION	97
BIBLIOGRAPHIE	98
ANNEXES	103

LISTE DES TABLEAUX

Tableau 3.1 Variables récupérées dans une simulation FDS	28
Tableau 4.1 Pseudo-code de génération de plans de coupe sur CPU.....	36
Tableau 4.2 Algorithme de génération des triangles coplanaires d'un plan p_i	40
Tableau 4.3 Nuanceur de sommets HLSL	47
Tableau 4.4 Nuanceur de fragments HLSL.....	47
Tableau 5.1 Valeurs d'indice de réfraction pour différents matériaux pour $\lambda = 589.29$ nm, $T = 0^\circ\text{C}$ et $P = 101,325 \text{ kPa}$	56
Tableau 5.2 Nuanceurs HLSL de sommets et de fragments utilisés pour la première passe de l'effet de chaleur.....	61
Tableau 5.3 Nuanceurs HLSL de sommets et de fragments utilisés pour la seconde passe de l'effet de chaleur	64
Tableau 6.1 Plateforme de visualisation	70
Tableau 6.2 Scénarios simulés sur FDS.....	70

LISTE DES FIGURES

Figure 1.1 Représentation schématique du pipeline de simulation.....	5
Figure 2.1 Phénomène de mirage inférieur	14
Figure 2.2 Simulation d'incendie Barrero et al. (2003) (a) retour de flammes (b) feu de maison	22
Figure 4.1 Idée générale de rendu volumétrique par plan de coupes.....	31
Figure 4.2 Pipeline du traitement des valeurs de densité pour chaque temps t	32
Figure 4.3 Représentation des données dans le format PLOT3D (a) et RAW (b).....	32
Figure 4.4 Plans de coupe sur la direction de $\vec{V}_{CAM}$, tirée de Ikits et al. (2004).....	35
Figure 4.5 Représentation de la distance s entre deux plans de coupe à travers une grille volumétrique	37
Figure 4.6 Positions de diverses caméras vis à vis la boîte englobante B	38
Figure 4.7 Droite normale aux plans de coupe p_i	40
Figure 4.8 Ordonnancement des sommets à partir d'un point d'ancrage (rouge) sur un plan à trois points d'intersection (bleus)	41
Figure 4.9 Rayon de lumière en milieu participant. (a) Absorption (b) Émission (c) Dispersion vers l'externe (d) Dispersion vers l'interne	44
Figure 4.10 Fonction de transfert pour de la fumée blanche.....	45
Figure 4.11 Traitement des nuances de sommets et de fragments pour le rendu de la fumée.....	46
Figure 5.1 Étapes de la génération de l'effet de chaleur.....	51
Figure 5.2 Indice de réfraction en fonction de la température et de la pression à partir de la relation de Lide (2003)	55
Figure 5.3 Rendu de l'effet de distorsion de chaleur	60
Figure 5.4 Application d'un bruit gaussien sur une image	65
Figure 5.5 Variance par pixel des ΔE_{Ni}	65

Figure 6.1 Rendu C107	71
Figure 6.2 Rendu Cuisine.....	72
Figure 6.3 Plans de coupe (a) Volume 5m x 5m x 5m $s_f = 1.0$ (b) Volume C107 $s_f = 1.0$ (c) Volume C107 $s_f = 0.25$	73
Figure 6.4 Textures alpha 8 bits de densité de fumée simulation C107 128x128x32 pixels au temps 200.01s (a) tranche 8 (b) tranche 9.....	75
Figure 6.5 Textures alpha 8 bits de densité de fumée simulation Cuisine 64x128x64 pixels au temps 130.0s (a) tranche 29 (b) tranche 31 (c) tranche 54.....	76
Figure 6.6 Échelle de couleurs de l'effet de fumée	76
Figure 6.7 Effet de fumée simulation C107 textures 8 bits alpha 128x128x32 avec une résolution 640x480 vue externe sans scène $s_f = 2.0$	77
Figure 6.8 Effet de fumée simulation C107 textures 8 bits alpha 128x128x32 avec une résolution 640x480 (a) Vue interne $s_f = 2.0$ (b) Vue interne $s_f = 1.0$ (c) Vue interne $s_f = 0.25$	78
Figure 6.9 Effet de fumée simulation Cuisine textures 8 bits alpha 64x128x64 avec une résolution de 640x480 vue externe sans scène $s_f = 2.0$	79
Figure 6.10 Effet de fumée simulation Cuisine textures 8 bits alpha 64x128x64 avec une résolution de 640x480 (a) Vue interne $s_f = 2.0$ (b) Vue interne $s_f = 1.0$ (c) Vue interne $s_f = 0.25$	80
Figure 6.11 Variance par pixels résolution 640x480 sur canal rouge (a) C107 proche (b) Cuisine (c) C107 éloigné.....	82
Figure 6.12 Texture de bruit gaussien 128 x 128 8 bits luminance	84
Figure 6.13 Effet de chaleur simulation C107 résolution 640x480 textures alpha 8 bits s_f $= 1.0$ (a) Scène sans effet (b) avec effet de chaleur (c) avec effet de chaleur et fumée.....	85
Figure 6.14 Effet de chaleur simulation Cuisine résolution 640x480 textures alpha 8 bits $s_f = 1.0$ (a) Scène sans effet (b) avec effet de chaleur (c) avec effet de chaleur et fumée	87

Figure 7.1 Artefact de la technique par plans de coupe (a) vue externe (b) intersection avec des surfaces	90
--	----

LISTE DES SIGLES ET ABRÉVIATIONS

CFD	Computational Fluid Dynamic
MDF	Mécanique des fluides
ENS	Équations de Navier-Stokes
GPU	Graphic Processing Unit
HRR	Heat Release Rate
LES	Large Eddy Simulation
LBM	Lattice Boltzmann Model
MSP	Ministère de la Sécurité Publique
NIST	National Institute of Standards and Technologies
TDC	Taux de dégagement de chaleur
API	Application programming interface

CHAPITRE 1 - INTRODUCTION

1.1 Contexte

Chez nos voisins du sud, les incendies coûtent 100 milliards de dollars à l'économie américaine en dommages structuraux, gestion de départements de protection contre les incendies et coûts d'assurances. (Federal Emergency Management [FEM], 1998). Plus graves encore sont les pertes humaines, pour lesquelles on ne peut mettre de prix : selon la FEM, environ 4000 personnes meurent chaque année et 23000 autres sont blessées. Plus de 80% d'entre eux mourront à l'intérieur de leur domicile.

Au Québec, 10 770 incendies ont eu lieu au cours de l'année 2002 (Ministère de la Sécurité publique, 2003). De ce nombre, 78% ont eu lieu dans le secteur résidentiel et 47% sont expliqués par l'utilisation négligente d'un appareil à cuisson ou de chauffage. L'inhalation de fumée et de gaz toxique est la cause principale des décès lors d'incendies mortels. Malgré l'instauration de normes de conception en bâtiment plus strictes et l'établissement de codes de sécurité, la propagation de la fumée est encore un problème majeur qui requiert une bonne connaissance du comportement des incendies.

Le National Institute of Standards and Technologies (NIST) aux États-unis est un organisme gouvernemental impliqué dans le développement de modèles et de simulations d'incendies sur le terrain. Ils étudient aussi bien des paramètres simples comme la taille des flammes ou la vitesse des courants que des paramètres plus complexes comme la distribution de la fumée et de la température dans un environnement clos.

Les essais d'incendies à grande échelle sont coûteux : ils peuvent, selon le NIST, dépasser rapidement les 50 000 dollars US par essai. De plus, le temps requis pour

mettre en place ces tests ainsi que le nombre limité d'endroits disponibles ne facilitent pas la réalisation de nombreux essais. La simulation d'incendie est donc l'alternative à favoriser.

1.2 Modéliser sans se brûler

Au cours des années 70, le NIST développe des modèles d'incendies par zones qui décrivent comment le feu évolue à l'intérieur de compartiments. Ces modèles divisent la région d'incendie en deux zones spatialement uniformes. La zone supérieure représente la fumée et les produits de combustion provenant du feu et la zone inférieure contient de l'air à température ambiante. Ce modèle d'incendie a été le premier à être modélisé et visualisé grâce à l'informatique et l'infographie.

La modélisation et la visualisation par ordinateur sont devenues par la suite des outils importants pour comprendre l'évolution et le comportement des incendies et permettent de simplifier grandement l'approche des tests sur le terrain. Dans le cas où la simulation est une reproduction d'un incendie passé, on peut vouloir une approximation des différents paramètres pour fin de validation. D'autre part, on peut effectuer le test d'incendie sur le terrain et par la suite réaliser informatiquement des scénarios alternatifs qui n'ont pas pu être effectués sur le terrain. Dans le monde de la simulation informatique, une maison peut effectivement brûler un nombre infini de fois!

1.3 Problématique et objectifs

Le modèle par zones s'exécute aujourd'hui en temps réel. Depuis le développement de ce modèle, d'autres modèles d'incendie ont fait leur apparition, notamment ceux dérivant des équations employées en mécanique des fluides (MDF). Chaque modèle de simulation utilise une ou plusieurs méthodes de visualisation. Ces méthodes peuvent être classées en deux catégories, soit les méthodes visuelles scientifiques et les méthodes visuelles approximatives.

Les méthodes de visualisation à caractère scientifique, notamment le *marching* tétraèdre de Bourke (1997), nous renseignent sur le caractère numérique concernant le phénomène étudié, mais ces dernières sont généralement pauvres au niveau du réalisme visuel. Par contre, certaines sont, au contraire, très réalistes comme par exemple la technique de placage de photons (*photon mapping*) de Jensen & Christensen (1998) utilisé par Fedkiw & Stam & Jensen (2001) pour le rendu de fumée. Toutefois, on ne peut pas, à l'aide de ces dernières, visualiser en temps réel les résultats de la simulation.

Les méthodes visuelles approximatives, comme les systèmes de particules de Reeves (1983) ou leur application aux flammes par Nguyen (2004) tentent d'imiter le phénomène physique par l'utilisation de plusieurs artifices. Ces méthodes donnent des résultats visuels impressionnants et peuvent s'exécuter en temps réel. Toutefois, certains aspects du phénomène physique, notamment l'effet « tourbillon » de la fumée de cigarette, sont très complexes à réaliser avec ces méthodes et requièrent beaucoup de manipulations manuelles. De plus, celles-ci ne font qu'approximer le phénomène en question et la plupart n'ont pas de sources de données physiques.

Compte tenu des avantages et inconvénients des méthodes de visualisation scientifiques et approximatives, cette recherche a pour but de concilier les deux types

pour développer une méthode hybride pour la visualisation de deux phénomènes reliés aux incendies : la fumée et l'effet de distorsion visuel causé par la chaleur intense.

Ce projet de recherche prend ses sources dans le logiciel de simulation d'incendie *Fire Dynamic Simulator* (FDS), conçu par le *National Institute of Standards and Technology* (NIST) aux États-unis. Les simulations FDS sont basées sur les équations de Navier-Stokes et les différentes données numériques produites sont utilisées pour générer le modèle graphique de la simulation

Ce projet a trois objectifs spécifiques qui sont complémentaires :

- rehausser le degré de réalisme visuel des simulations physiques d'incendies, notamment au niveau de la génération de fumée et de la distorsion due à la chaleur tout en maintenant un bon compromis sur la qualité de l'information physique donnée;
- permettre une visualisation en temps réel des résultats de la simulation en exploitant les capacités des unités de traitement graphique (GPU) du matériel;
- développer un pipeline efficace comprenant la simulation, le traitement des données et la visualisation.

Le dernier objectif est un objectif intégrateur. Une fois les deux premiers objectifs atteints, il s'agit alors de mettre sur pied un « pipeline » de simulation-visualisation. La figure 1.1 montre les étapes de celui-ci.

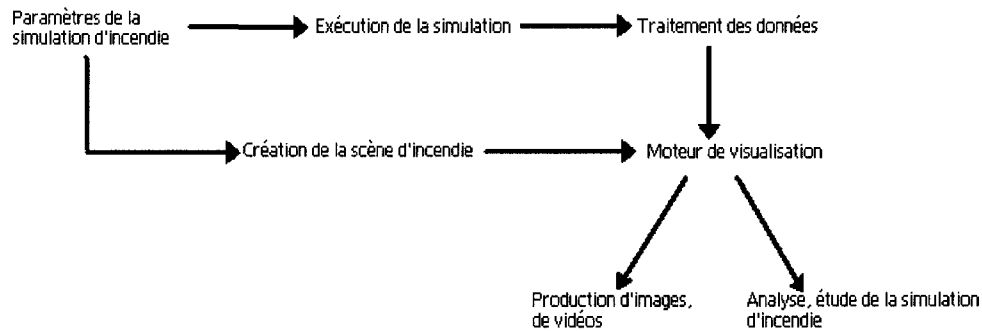


Figure 1.1 Représentation schématique du pipeline de simulation

Les paramètres de simulation sont d'abord fournis au simulateur d'incendie, FDS. Le simulateur est une boîte noire et est interchangeable. Une fois la simulation exécutée, un traitement des données est nécessaire pour d'une part ne retirer que les informations essentielles et d'autre part, rendre l'information compatible avec la prochaine étape, soit le moteur de visualisation. Ce moteur reçoit deux entrées : une scène d'incendie et les données nécessaires pour la représentation des deux phénomènes à visualiser. Finalement, à partir des images et des animations du moteur de visualisation, étudier l'évolution et le déroulement de la simulation d'incendie.

Le problème à résoudre pour la visualisation de la fumée est le suivant : à partir des valeurs de densité de fumée ρY_s données pour tout point de grille d'une simulation, développer la capacité d'explorer les résultats volumétriques de manière interactive et obtenir une bonne représentation visuelle de celle-ci avec une scène 3D.

La problématique de la visualisation de la distorsion due à la chaleur est semblable au problème de visualisation de la fumée. Il s'agit de mettre au point une technique de visualisation basée sur les valeurs calculées de température aux points de grille d'une simulation FDS. Plus précisément, développer une méthode de visualisation

simple qui s'intègre bien à un rendu à base de polygones et à la technique de visualisation de fumée.

1.4 Aperçu des différents chapitres

Le chapitre 2 de ce mémoire présente une revue de littérature sur les travaux précédents réalisés sur la fumée, l'effet de distorsion et mirage ainsi que le feu. Une attention particulière est portée aux travaux dérivant de la MDF et autres techniques de simulations produisant des données volumétriques dans un espace 3D. La revue sur le feu n'est fournie que par souci de complétude, étant un élément majeur d'un incendie.

Le chapitre 3 porte sur FDS, étant donné qu'il est le simulateur d'incendie utilisé pour obtenir les données volumétriques de ce projet de recherche. Le modèle numérique, les limites du simulateur ainsi qu'un court survol de la spécification d'une simulation sont décrits dans ce chapitre.

La fumée est abordée en détail au chapitre 4. Ce chapitre décrit la méthodologie de visualisation choisie et son implémentation au niveau du processeur et du GPU. La caractérisation de l'illumination de la fumée est aussi présentée dans ce chapitre. L'effet de distorsion est abordé en détail au chapitre 5. De manière similaire à la fumée, la méthodologie de visualisation choisie y est décrite ainsi que les algorithmes impliqués et leurs implémentations.

Le chapitre 6 présente les résultats obtenus pour deux simulations types, soit une salle de cours et une cuisine. L'intégration de la fumée et de l'effet de distorsion dans le pipeline de simulation montré à la figure 1.1 est aussi discutée dans ce chapitre. Finalement, le chapitre 7 parle des modèles d'effets présentés au chapitre précédent.

CHAPITRE 2 - TRAVAUX ANTÉRIEURS

Ce chapitre présente la revue de la littérature sur trois aspects d'un incendie : la fumée, l'effet de distorsion et le feu. Pour chaque aspect, nous avons porté notre attention sur deux sujets. D'une part, les travaux dérivés de la MDF sont essentiels étant donné la source des données générées par FDS. D'autre part, comme on cherche à réaliser une méthode hybride de visualisation, il importe de regarder ce qui a été fait dans le domaine temps réel où les aspects physiques sont représentés par un ou plusieurs artifices visuels non simulés.

2.1 Fumée

2.1.1 Simulations physiquement réalistes basées sur la MFD

Une méthode naturelle pour modéliser la fumée est de le faire à partir des équations disponibles dans le domaine de la MDF. Les premiers auteurs à modéliser la fumée en infographie dans le domaine de la MDF furent Kajiyama et Herzen (1984). Étant données les capacités de calcul de l'époque, les travaux de Kajiyama et Herzen (1984) n'ont porté que sur de petites grilles de simulation.

Il faudra attendre plus de dix ans pour que de réels progrès dans le domaine soient réalisés. Les travaux de Foster et Metaxas (1996) intègrent ainsi un bon mouvement tourbillon à la fumée en trois dimensions, bien qu'encore une fois la grille de simulation demeure petite. Comme leur modèle utilise une technique d'intégration explicite, leurs simulations sont stables uniquement lorsque le pas de temps est très petit. Ceci résulte en des temps de calcul relativement longs. Pour corriger le problème, Stam (1999) proposa un modèle qui est inconditionnellement stable. Le mouvement général de la fumée ainsi produit apparaît de manière fluide et réaliste, mais souffre d'une trop

grande dissipation numérique ce qui fait disparaître certains aspects de la fumée trop rapidement.

Fedkiw & Stam & Jensen (2001) modélisent la vitesse de la fumée grâce à l'équation d'Euler des fluides incompressibles. Ils résolvent les équations avec une méthode d'intégration semi-Lagrangienne suivie par une équation de Poisson pour la pression. Ceux-ci pallient au problème de dissipation numérique de Stam (1999) en mettant de l'avant une technique d'intégration numérique avec dissipation réduite. Ils réintroduisent la dissipation numérique dans le fluide, qui se traduit en énergie perdue, en ajoutant une nouvelle force basée sur les paramètres de confinement de vorticité. Le rendu de la fumée est réalisé par deux modules distincts. Le premier module a pour but de donner une approximation rapide aux animateurs. Dans un premier temps, ils déterminent la quantité de lumière qui atteint chacun des voxels de la grille volumétrique. À toutes les fois où un rayon traverse un voxel, la transparence de celui-ci est déterminée à partir de sa densité. Dans un second temps, la visualisation des densités est réalisée par une technique de plans de coupe dans le volume de simulation selon l'axe le plus aligné avec la direction de vue.

Fedkiw et al. (2001) proposent un second module, cette fois-ci pour obtenir de meilleurs résultats visuels. En effet, la modélisation d'une fumée plus réelle doit tenir compte d'un phénomène optique supplémentaire, soit la dispersion multiple des rayons de lumière (*multiple light scattering*). Ce dernier survient lorsque les rayons de lumière traversent un milieu participant (homogène ou non), comme de la fumée. Les rayons peuvent alors être déviés et dispersés à l'intérieur du milieu, ce qui affecte grandement l'illumination et l'aspect de celui-ci. Fedkiw et al. (2001) utilise donc une technique de placage de photons (*photon mapping*) appliquée aux milieux participants, initialement proposé par Jensen & Christensen (1998) et qui résout en partie l'équation décrivant l'état de la lumière en présence d'un milieu participant de Chandrasekhar (1960).

Treuille, McNamara, Popovic et Stam (2003) s'appuient sur les travaux de Fedkiw et al. (2001) pour développer une simulation de fumée où il y a un plus grand contrôle sur la fumée elle-même. En effet, les modèles présentés plus tôt offrent tous un contrôle de base des paramètres de génération de fumée comme la quantité, le point de génération ou encore la densité cible. Toutefois, la manipulation de ces paramètres procure peu de contrôle direct sur la simulation. Treuille et al. (2003) proposent donc l'utilisation de trames clés (key frame), qui vont ainsi diriger la simulation.

Pour atteindre le comportement désiré, Treuille et al. (2003) résolvent un problème d'optimisation quasi-Newton continu pour trouver la force appropriée à appliquer au(x) champ(s) vectoriel(s) de vitesse de la simulation. Ils formulent une fonction qui décrit comment et à quel degré correspond une trame clé par rapport à l'état instantané de la simulation. Le problème consiste alors à résoudre les dérivées appropriées pour minimiser la fonction. Le degré de contrôle ainsi obtenu est impressionnant mais requiert deux heures de calcul pour une grille 2D de 50x50 sur un Pentium IV 2GHz.

Dans le même esprit de flexibilité sur les simulations de fumée, Fattal & Lishinski (2004) s'appuient sur les travaux de Treuille et al. (2003). Le résultat principal de leurs travaux est le niveau de contrôle donné aux animateurs pour la modélisation de la fumée via des paramètres plus intuitifs. Pour ce faire, ils introduisent deux concepts : une force directrice qui transforme la fumée vers une densité cible et un terme d'attraction qui empêche la fumée de se dissiper trop rapidement (problème de dissipation numérique). Le premier terme est ajouté à l'équation. Ils représentent une fonction qui amène la densité initiale de la fumée vers une densité cible. Ce terme ne dépend que de l'état courant instantané du système. Le terme d'attraction est ajouté à l'équation d'advection, qui décrit le transport de la fumée par le courant du fluide. Il sert à contre balancer partiellement la dissipation numérique qui cause la diffusion trop rapide de la fumée, comme mentionné par Fedkiw et al. (2001). Leurs simulations sont

en moyenne deux fois plus rapides que celles proposées par Treuille et al. (2003) sur du matériel comparable. Le niveau de contrôle donné aux animateurs leur permet de spécifier des trames clés sensiblement différentes et ainsi réaliser un effet de morphage avec de la fumée. De plus, il est possible d'effectuer des animations 3D volumétriques de fumée, en spécifiant un nombre raisonnable de trames clés. Malheureusement, aucun détail n'est donné sur la technique de rendu utilisé pour la fumée elle-même.

2.1.2 Simulations temps réel

La complexité numérique des techniques de simulation précédentes (Fedikw et al. 2001 ; Treuille et al. 2003 ; Fattal et Lishinski 2004) les rend inaptes à fonctionner dans un environnement interactif. Plusieurs modèles simplifiés de simulation ont été proposés au fil des années (Reeves 1983 ; Cabal & Cam & Foran 1994).

Un modèle récent est celui proposé par Nguyen (2004). Celui-ci utilise un système de particules, initialement proposé par Reeve (1983) pour réaliser son effet de fumée. La fumée est représentée par des textures 2D, soit des lutins (*sprites*), dont la génération a été réalisée en pré-traitement. L'illumination a été retouchée manuellement pour les besoins de la scène, puisque la réalisation d'un éclairage physiquement correct de la fumée est un sujet complexe.

Le mouvement des particules est spécifié que par une seule force directrice. Comme les systèmes de particules sont liés à des objets dynamiques, Nguyen (2004) pallie au problème de particules dites « solitaires » en ajoutant un système de poids à celles-ci. Ainsi, lorsqu'une particule est créée, elle reste attachée à son émetteur si celui-ci bouge. Lorsque la particule vieillit, le poids diminue de façon linéaire et l'émetteur parent a moins d'impact sur la particule. Pour donner plus de réalisme à la fumée, l'auteur ajoute un mouvement circulaire selon l'axe des x, ce qui donne l'impression que la fumée s'enroule sur elle-même. Finalement, l'auteur opte pour un rendu avec

composition alpha (alpha blending), étant donné que la composition additive sature les couleurs beaucoup trop rapidement.

Il est très difficile de contrôler l'évolution dynamique de ces simulations comme elles apparaissent dans la réalité. Krüger et Westermann (2005) proposent donc un modèle de simulation temps réel hybride, qui exploite les dernières fonctionnalités offertes sur GPU.

Krüger et Westermann (2005) résolvent sur GPU les équations de Navier-Stokes (ENS) qui gouvernent le mouvement de fluides newtoniens non turbulents. Comme la résolution numérique de ces équations sur des grilles 3D de taille considérable consomme beaucoup de puissance de calcul, la simulation physique n'est réalisée que sur des plans de coupes du domaine 3D, qu'ils appellent des « proxy ». Les valeurs volumétriques dans le domaine 3D sont récupérées par interpolation linéaire sphérique.

Pour visualiser la simulation, Krüger et Westermann (2005) combinent deux modèles de rendu exploitant les fonctionnalités offertes sur GPU. Le premier est un modèle basé sur le rendu volumétrique à base de texture, comme proposé par Cabral et al. (1994) et Westermann (1998). Au lieu de reconstruire le volume de données au complet pour le rendu, ils construisent simplement des plans de coupe alignés avec la direction de la vue et dessinés dans un certain ordre. Ce module est particulièrement adapté pour la visualisation de valeur de densité.

Le second module est à base de particules, comme décrit dans Reeves (1983) et sert à représenter le flux simulé. Les positions initiales des particules sont stockées dans les composantes RGB d'une texture conventionnelle. Pour chaque particule, le nuanceur de fragments récupère la position de la particule de la texture et reconstruit le champ vectoriel de la position de la particule grâce aux informations contenues dans les plans de coupe. Finalement, les nouvelles positions de chacune des particules sont calculées avec une intégration d'Euler dans la direction du flux dans le nuanceur de fragments. À

partir d'un tableau de sommets statique, ils mettent à jour dans le nuanceur de sommets les positions des particules et dessinent celles-ci grâce à l'utilisation de lutins à point, qui requièrent moins de traitement qu'une série de triangles.

Krüger et Westermann (2005) coupent ainsi les ponts avec la simulation physique grâce à la simulation partielle sur de la géométrie proxy. Leur véritable innovation se situe toutefois en l'utilisation de « patrons » de vitesse et de pression. Ces patrons perturbent localement les champs de pression et de vitesse pour donner un plus grand contrôle aux animateurs.

2.2 Mirages et distorsions

Bien que l'on s'attaque uniquement à l'effet de distorsion générée par la chaleur intense dégagée par un incendie, il importe de retracer les travaux réalisés dans le cas de mirages. En effet, le phénomène physique impliqué, soit la variation de l'indice de réfraction de l'air dû au réchauffement du sol, est similaire. Comme pour la fumée, on s'intéresse à la technique de rendu utilisée, étant donné que l'on possède déjà les résultats d'une simulation d'incendie.

2.2.1 Phénomène physique

Le phénomène de distorsion de la visualisation dû à la chaleur est similaire à ce qui se produit lors de mirage. Dans le cas de mirage, le sol chauffé par le soleil augmente la température de l'air à proximité, ce qui lui donne un indice de réfraction différent de l'air ambiant plus froid. Les rayons de lumières traversant la région plus chaude dévient et arrivent ainsi sous un différent angle à l'œil. La figure 2.1 illustre le phénomène de mirage inférieur, fréquemment observé dans le désert ou les routes de pays chauds.

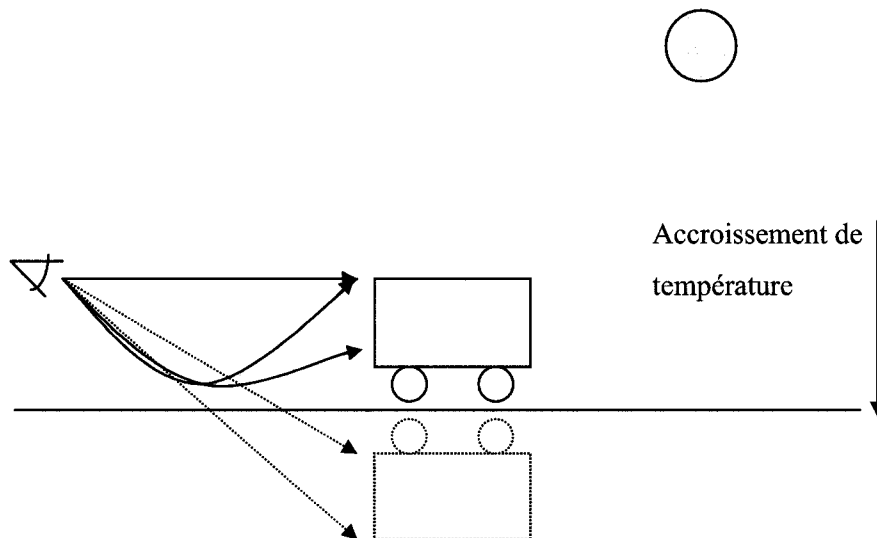


Figure 2.1 Phénomène de mirage inférieur

Un mirage inférieur survient lorsque la température descend rapidement alors que la hauteur augmente. Un rayon suffisamment haut sera droit, alors que les rayons de lumière qui frappent le bas du camion de la figure 2.1 seront courbes. Dans le cas de mirage inférieur, une image inversée se crée sur le sol. Il existe d'autres types de mirage, comme par exemple le mirage supérieur où une image double apparaît au-dessus des éléments observés.

2.2.2 Méthodes ad-hoc

L'effet de distorsion peut être réalisé par des méthodes dites « ad hoc ». Ces méthodes, comme celles à base de bruit, sont de nature qualitative et ont peu en commun avec le phénomène physique. De manière générale, celles-ci requièrent la manipulation manuelle de leurs paramètres pour obtenir l'effet désiré.

Un exemple de ces méthodes ad hoc est celle proposée par Nguyen (2004). C'est une technique de rendu basé sur une texture de bruit gaussien. À partir d'un système de particules orientées, il dessine d'abord dans un tampon les particules avec la texture de bruit. Le tampon est ensuite récupéré et réutilisé dans un nuanceur pour la composition finale de l'image. Lors de cette composition, les canaux rouge et vert du tampon, devenu une texture de « chaleur », sont utilisés pour faire varier les coordonnées de textures u et v lors de la récupération des texels de la texture contenant la scène. Le résultat final est une image contenant de la distorsion là où les particules étaient dessinées.

2.2.3 Méthodes basées sur le lancer de rayons

Le lancer de rayons (*ray tracing*) a été utilisé pour simuler plusieurs phénomènes optiques comme la réfraction, la réflexion, ainsi que les ombres. Les premiers algorithmes, comme ceux développés par Whitted (1980) et Cook (1986) proposent des approximations au modèle d'illumination globale et en partie de l'illumination locale. Cook (1986) se distingue par la simulation des réfractions spéculaires imparfaites.

Les mirages et la distorsion de chaleur font partie de ces phénomènes naturels où il est aisé d'effectuer un rapprochement avec les méthodes à lancer de rayons. C'est Berger (1990) qui fut le premier à s'attaquer au phénomène de réfraction / réflexion atmosphérique en infographie, qui est la cause des mirages et de la distorsion. Berger (1980) étend les méthodes de lancer de rayons pour y intégrer l'effet de mirage. Son équipe modélise la variation continue du spectre de température en divisant l'espace concerné par couche. Le mirage est alors une boîte virtuelle où chaque couche possède un indice n de réfraction différent.

Lorsqu'un rayon pénètre cet objet virtuel, celui-ci est dévié à chaque couche intersectée. La déviation est calculée grâce à la loi de la réfraction de Snell-Descartes. Le rayon finira par quitter la boîte et continuer sa course dans le reste de la scène. Suite aux travaux de Berger (1990), Musgrave (1991) affirma que le principal agent de déviation des rayons de lumière lors de mirages est la réflexion totale interne et non pas la réfraction.

Plus récemment, les travaux de Seron et al. (2004) ont porté sur une méthode de lancer de rayons curvilignes capable de représenter des phénomènes atmosphériques apparaissant sous certaines conditions. Les travaux précédents supposaient que la

lumière traversait l'environnement en ligne droite et dans un milieu homogène. Seron et al. (2004) résolvent le problème de la propagation de la lumière dans un milieu non homogène en utilisant une équation générale basée sur le principe de Fermat. Ils appliquent leur méthode sur les phénomènes apparaissant en haute atmosphère comme les doubles soleils, les soleils entrecoupés ou encore les soleils écrasés.

Zhao et al (2006) ont quant à eux modélisé les transferts de chaleur entre les sources de chaleur et l'air ambiant. La modélisation physique du phénomène de transfert leur donne un avantage sur les autres méthodes (Berger, 1990; Musgrave, 1991; Seon et al., 2004) où l'air ambiant n'interagit pas avec les sources ou n'est pas considéré. Ils proposent un modèle de transfert de chaleur hybride Boltzmann (HTLBM) combiné à un algorithme Boltzmann sur réseau à relaxation multiple de temps (multiple-relaxation-time LBM). La résolution de celui-ci donne un volume de distribution de température de l'environnement qui est utilisé pour déterminer le nouvel indice de réfraction de l'air n à pas discret dans le volume. Le volume de température ainsi généré sera utilisé dans le cadre d'un lanceur de rayons non-linéaire, où les rayons de lumière partant de l'oeil seront déviés ou réfléchis en fonction du nouvel indice de réfraction de l'air n et de la loi de Snell-Descartes. La distribution de la température sur les sources de chaleur, comme un sol chaud, est contenue dans ce qu'ils appellent une texture de température. Celle-ci est appliquée directement à la géométrie et permet la résolution du modèle sur le processeur graphique (GPU). Il est à noter que le modèle de Zhao et al. (2006) fonctionne uniquement à travers un médium homogène, comme l'air ambiant.

2.3 Feu

Parmi les phénomènes naturels, le feu est souvent perçu comme le plus difficile à réaliser, car notre compréhension de celui-ci, encore aujourd'hui, est limitée. Le feu requiert trois éléments selon Wei et al. (2002) : un carburant, de l'oxygène et de la chaleur.

Le carburant peut être sous forme solide, liquide ou gazeux. Toutefois, pour pouvoir brûler, il doit être décomposé en gaz ou en vapeur. Ceci arrive sous l'action de la chaleur. La chaleur est la mesure de l'activité moléculaire se déroulant à l'intérieur d'un objet. Avec suffisamment de chaleur, les molécules d'un objet peuvent bouger si rapidement qu'il se détache du reste du corps. C'est ainsi que le carburant est transformé en gaz. Il existe trois façons de transférer de la chaleur à un objet : par advection, par radiation ou par conduction.

Cette recherche ne s'attarde pas spécifiquement à la modélisation de flammes. Elle s'intéresse plutôt aux phénomènes secondaires résultant de l'incendie, comme la production de fumée ou les dégagements de chaleur. Les sous-sections qui suivent présentent toutefois une courte revue sur la réalisation, la modélisation et la visualisation de flammes sur ordinateur.

2.3.1 Approches par acquisitions de données expérimentales

Il est possible de mesurer, avec des équipements spécialisés, les valeurs expérimentales reliées à un incendie. Rushmeier et al. (1995) mesurent les transferts par radiation via des sondes qui mesurent à très courte portée. Ces sondes, refroidies à l'eau, en paires et opposées mesurent selon une direction radiale semi-cylindrique des valeurs comme la température instantanée, la concentration de suie ou la concentration de CO_2

dans trois longueurs d'ondes : 900 nm, 1000 nm et 4350 nm. L'incendie est un puits d'essence à base d'heptane de 10 cm de diamètre.

Une fois les données acquises, la visualisation de ces données de flammes est réalisée grâce à un algorithme de lancer de rayons. Les données étant dans un volume cylindrique, les rayons traversent des couches le long de la direction radiale. Les rayons partent ainsi de l'œil et les valeurs de température, de concentration de suie ainsi que de pression partielle du CO_2 sont récupérées à chaque intersection de couches. L'intensité de la lumière est ensuite déterminée grâce à l'utilisation d'une formule d'intégration qui procure la radiance dans le spectre visible.

Rushmeier et al. (1995) ont conclu que la mesure des données de flammes requiert énormément de points de mesure et que les techniques de visualisation, comme le lancer de rayons, aident beaucoup à l'étude des données mesurées.

2.3.2 Approches par simulations numériques

C'est sans surprise que l'approche privilégiée pour la réalisation de flammes soit la simulation numérique du feu lui-même. Comme mentionné au chapitre premier, l'exécution d'incendies sur le terrain et donc de flammes est peu pratique, coûteux et potentiellement dangereux.

Chiba et al. (1994) ainsi que Stam et Fiume (1995) simule le feu grâce à l'utilisation de distributions de carburant. Chiba et al. (1994) utilise un champ vectoriel de vitesse basé sur les vortex ainsi qu'une distribution de carburant 2D pour décrire la propagation des flammes. Stam et Fiume (1995) ont quant à eux construit, pour tous les objets inflammables, une distribution indiquant la densité de carburant et la température

pour chaque point de la surface. Ils utilisent par la suite une technique par différences finies pour simuler le feu.

Certains ont modélisé la propagation des flammes sur des surfaces polygonales. Perry et Picard (1994) et plus récemment Beaudoin et al. (2001) ont ainsi connecté plusieurs points d'échantillonnage pour représenter la frontière de leurs flammes. Lee et al. (2001), basant leur recherche sur les résultats de Perry et Picard (1994) et Beaudoin et al. (2001), ont modélisé à la fois l'évolution des flammes sur des géométries complexes et la fusion de plusieurs feux différents.

Wei et al. (2002) propose un modèle équilibré entre performance de la simulation et réalité physique. La simulation du feu est basée sur l'utilisation d'un LBM réalisé sur GPU, où le feu est considéré comme un gaz chaud. Le LBM permet de calculer un volume 3D de vitesses, qui sera utilisé pour déterminer la vitesse de chaque particule de flamme. Les particules de flamme sont représentées par une texture, qui est fonction de la chaleur et de la localisation dans le feu. La vitesse de chaque particule est déterminée par interpolation trilinéaire et celles-ci sont retirées au fur et à mesure que leur carburant est consommé. Au cours de l'année suivante, Wei et al. (2003) utilise un LBM pour générer les champs de vitesses d'objets en flamme à une vitesse interactive. Cette fois-ci, ils mettent l'emphasis sur les mouvements de fronts de flamme et leurs propagations. Un champ de distance, qui dicte la distance par rapport à une surface en tout point d'un espace fini, leur permet de s'assurer que les points de front de flamme « collent » à des surfaces virtuelles. Les particules de flamme sont émises à partir de la surface des objets en flamme et bougent selon le champ de vitesse déterminé par le modèle LBM. Le rendu des particules de flamme est réalisé de la même manière que Wei et al. (2002), soit à base de splat.

Parallèlement à la simulation de fumée de Fedkiw et al. (2001), il est possible d'utiliser les ENS de la MDF pour la simulation de flammes. Nguyen et al. (2002) simulent indépendamment le carburant sous forme vapeur et les produits gazeux chauds à l'aide des ENS. Ils modélisent l'expansion du carburant lorsque celui-ci entre sous forme vapeur lors de la transformation en produits gazeux chauds. Comme les ENS ne tiennent pas compte du phénomène d'expansion, ils proposent l'utilisation d'un modèle de flamme mince. Une surface implicite est utilisée pour représenter la zone de réaction où le carburant est converti en produits gazeux chauds.

Ils considèrent trois phénomènes visuels distincts dans leurs travaux. D'abord, ils modélisent le noyau bleu ou bleu-vert au centre des flammes. Cette couleur est due à l'émission d'espèces chimiques intermédiaires, comme les radicaux de carbone, durant la réaction chimique. Ensuite, ils s'attaquent aux radiations émises par les produits gazeux chauds et en particulier à la suie de carbone. C'est ce qui donne la couleur jaune-orange des flammes. Finalement, lorsque la température diminue, il y a production de suie. Ils modélisent celle-ci en conservant la densité de suie au cours de la simulation. Le rendu des flammes est réalisé avec un lanceur de rayons de type Monte Carlo, où la radiance de chaque rayon est approximée sur de courts segments, notamment grâce à l'utilisation de la fonction de phase de distribution sphérique de Henyey-Greenstein. De plus, leurs travaux tiennent compte du phénomène d'adaptation chromatique de l'œil et de l'interaction des flammes avec les objets et l'environnement.

Plusieurs travaux de recherches ont été entrepris à l'École polytechnique de Montréal sur les simulations d'incendies numériques. Barrero et al. (2003) ont travaillé sur l'intégration de la CAO, le transfert géométrique ainsi que sur la simulation de flux et de feu. Ils ont aussi développé plusieurs outils de visualisations de fumée et de flammes pour les simulations d'incendies dans les édifices. Les scénarios d'incendies sont d'abord modélisés en utilisant des outils standards. Les modèles sont par la suite exportés et découpés en voxels ou en maillages polygonaux. Les simulations d'incendies sont

réalisées par FDS. La technique de visualisation utilisée pour présenter les flammes est une version modifiée des blobs de Stam et Fiume (1993). Les blobs qui décrivent les flammes sont construits à l'aide d'une distribution de densité appliquée à une série de particules imbriquées basées sur les valeurs du taux de dégagement de chaleur (TDC) et de la fraction de mélange. Les blobs sont ensuite convertis en polygones et en textures 3D de petites tailles à travers le volume considéré. Le rendu des flammes est fonction de l'émission du corps noir et des propriétés de combustion des matériaux. Les figures 2.2a et 2.2b illustrent les résultats obtenus pour un retour de flammes et un feu de maison.

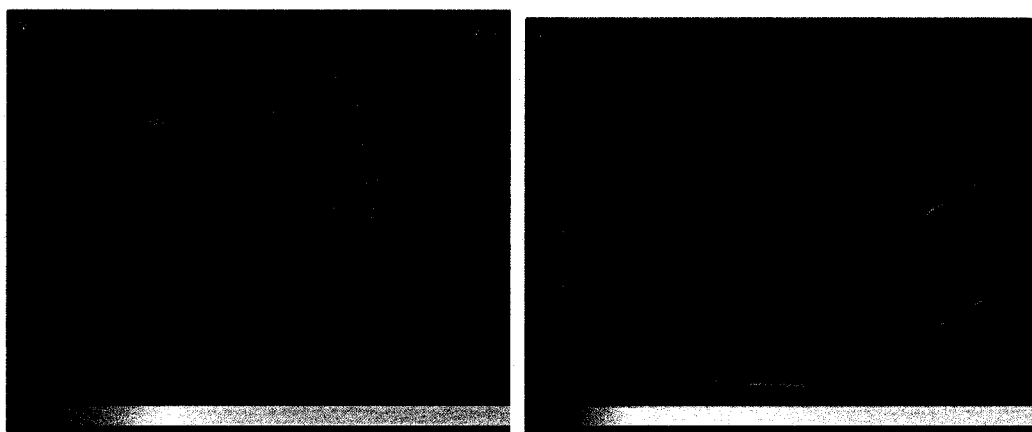


Figure 2.2 Simulation d'incendie Barrero et al. (2003) (a) retour de flammes (b) feu de maison

Abanto et al. (2005) de l'École Polytechnique de Montréal ont travaillé sur la propagation du feu et de la fumée dans un tunnel passant sous l'eau. Ils ont modélisé le tunnel Louis-Hyppolite-Lafontaine de Montréal et comparé les résultats de simulation obtenus avec un logiciel commercial (Fluent) et maison (Barrero et al. (2003)). La complexité et la taille du modèle les ont forcés à ne simuler qu'une partie du tunnel. Ils ont pu démontrer avec succès la viabilité des technologies à base de MFD dans la reproduction et la compréhension de la propagation des flammes et de la fumée dans les tunnels.

2.3.3 Approches approximatives

Certains auteurs délaissent la simulation numérique pour se diriger vers les simulations dites approximatives. En effet, la complexité du phénomène de combustion ainsi que les nombreux paramètres de combustion interdépendants rendent difficiles l'utilisation de simulations numériques pour la génération et l'animation de flammes.

Lamorlette et Foster (2002) reconnaissent ainsi les avantages de la simulation numérique mais démontrent que la complexité des flammes rend moins attrayante ce type de simulation. Leur méthode est basée sur l'observation que plusieurs aspects visuels d'une flamme sont de nature statistique. Ils ont donc développé une méthode d'animation de flammes basée sur un modèle stochastique pour les vacillements des flammes et l'effet de diffusion thermique flottante. Ce modèle est couplé à des champs de vents basés sur la physique et du bruit de Kolmogorov pour des animations contrôlées. Leur méthode tient compte des sources enflammées dynamiques, la propagation de flammes sur les objets, la séparation et la fusion de flammes ainsi que l'interaction avec les objets stationnaires. Finalement, ils mettent l'accent sur le contrôle de l'animation des flammes. Ainsi, un feu est spécifié par le profil désiré, la position générale, l'intensité, la durée de vie et l'évolution de la forme, la couleur, la taille et le comportement des flammes. Cette technique est très différente de celle de Nguyen (2002), où le feu est simulé numériquement à partir d'équations en mécanique des fluides où des données physiques sont impliquées.

La réalisation de flammes demeure un domaine complexe. Chaque méthode, que ce soit par mesures expérimentales, par simulations numériques ou par approximations demande une bonne compréhension des phénomènes physiques impliqués dans la production de flammes. Toutefois, les techniques élaborées par les différents chercheurs

ont toutes suffisamment de points d'extension pour pousser encore plus loin la réalisation de flammes sur ordinateur.

CHAPITRE 3 - FIRE DYNAMIC SIMULATOR (FDS)

Ce projet ne propose pas un nouveau modèle numérique pour simuler les incendies. Comme indiqué à la figure 1.1, la simulation numérique de l'incendie est une boîte noire qui produit les résultats à visualiser et FDS joue le rôle de simulateur d'incendie. Ce simulateur est développé par le NIST, qui est en charge des nombreux programmes d'études sur les incendies et est offert gratuitement à la communauté scientifique.

Ce chapitre donne une vue d'ensemble rapide de FDS. Le lecteur souhaitant en savoir davantage sur les fonctionnalités de celui-ci peut consulter le guide d'utilisateur ainsi que le guide technique offerts sur le site web du NIST sur <http://www.fire.nist.gov/fds/docs>.

3.1 Modèle de combustion

Le modèle de combustion assume que les phénomènes de convection et de transport radiatif peuvent être simulés directement mais que les processus physiques impliqués, dans le cas d'un pas de temps relativement petit, doivent être approximés.

Ces approximations dépendent de la qualité de la grille de simulation et de la limite de pas de temps autorisée pour la puissance de calcul disponible. Les processus chimiques impliqués dans la production d'énergie due à la combustion sont, dans la plupart des cas, inconnus pour un scénario d'incendie donné. C'est pourquoi FDS propose un modèle de combustion basé sur la fraction de mélange, $Z(x,t)$, des éléments chimiques impliqués.

Cette fraction de mélange est une fonction dépendante du temps et de la position dans l'espace de simulation et qui respecte les équations de la conservation. Elle représente, pour un temps t et une position x , la fraction de l'élément qui provient du carburant. À partir de la fraction de masse de l'oxygène et de la fraction de mélange, il est possible de calculer le taux de consommation locale de masse d'oxygène par l'incendie. Le taux local de dégagement de chaleur (HRR) est ensuite calculé à partir de la consommation locale d'oxygène, en supposant que celui-ci est directement proportionnel au taux de consommation d'oxygène indépendamment du carburant.

Le modèle assume donc que la réaction est entièrement contrôlée par la fraction de mélange et que la réaction du carburant et de l'oxygène est infiniment rapide. De plus, le modèle assume que la réaction ne dépend pas de la température ambiante des gaz et que la zone de réaction est une surface mince avec le carburant d'un côté et l'oxygène de l'autre.

La fraction de masse de tous les réactifs et les produits majeurs (carburant, O_2 , CO_2 , H_2O , N_2 , CO et suie) peut être déterminée à partir de la fraction de mélange carburant-oxygène en utilisant les relations d'état qui sont des expressions empiriques provenant de la combinaison d'analyses simplifiées et de mesures. En d'autres mots, la valeur de la fraction de mélange Z pour n'importe quelle cellule dicte la fraction de masse de tous les gaz donnés ci-dessus.

3.2 Limites du modèle et robustesse

Toute technique numérique utilisée pour résoudre un modèle donné est une source potentielle d'erreurs et la solution numérique développée par le NIST avec FDS n'en fait pas exception. Le plus important paramètre à considérer dans une simulation FDS est la taille de la grille de simulation. En général, plus fine est la grille de

simulation et plus précises sont les solutions des équations numériques. FDS est précis au second ordre dans le temps et dans l'espace, ce qui implique qu'une augmentation de l'ordre de 2 de la résolution de la grille de simulation réduit l'erreur de discrétisation des équations d'un facteur de 4.

Le modèle de combustion a été conçu au départ pour analyser des feux de grandes dimensions. Ainsi, FDS peut être utilisé de manière relativement fiable pour simuler des incendies lorsque la région d'incendie, généralement un édifice, est grande par rapport à la taille du feu. D'après la documentation du NIST sur FDS (2006), le modèle peut alors prédire les vitesses des gaz ainsi que la température dans un intervalle de $\pm 20\%$ des valeurs expérimentales. Toutefois, dans les cas où l'incendie est confiné, par exemple un incendie dans un garde-manger, l'incertitude du modèle est plus élevée et est due au manque de spécifications des propriétés des matériaux impliqués, à la chimie de la combustion et à une plus grande erreur numérique dans la simulation de la combustion et du transport radiatif de chaleur. Ces limites doivent donc être considérées dans l'élaboration des simulations pour cette recherche.

3.3 Spécification de la simulation et données

Une simulation d'incendie FDS requiert la spécification des conditions initiales comme la température et la pression ambiantes, les conditions aux frontières, les bouches d'aération, les frontières des objets (scène) ainsi que les propriétés physiques des solides et des gaz. Le chapitre 6 donne plus de détails sur les paramètres exacts fournis à FDS pour la production des résultats.

FDS peut tenir compte d'une vingtaine d'espèces chimiques mais seules les espèces du tableau 3.1 sont utilisées. En effet, les espèces du tableau ci-dessous seront celles exploitées dans le cadre de la visualisation de la simulation d'incendie. La densité de suie sera utilisée dans le cadre de la réalisation de la visualisation de la fumée au chapitre 4. La température des gaz ainsi que le HRRPUV sont quant à eux appliqués pour la génération de l'effet de distorsion.

Tableau 3.1 Variables récupérées dans une simulation FDS

Variables	Symboles	Unités
Température des gaz	T	°C
Densité de suie	ρY_s	mg / m ³
Taux de dégagement de chaleur par unité de volume (HRRPUV)	q^m	kW / m ³
Fraction de mélange	Z	kg/kg

Finalement, les données de la simulation FDS sont écrites à raison d'un fichier par pas de temps sous le format PLOT3D, initialement développé par la NASA pour la MDF.

CHAPITRE 4 - VISUALISATION DE LA FUMÉE

4.1 Élaboration de la méthodologie de visualisation

Beaucoup de phénomènes doivent être visualisés de façon volumétrique. En effet, les fluides, les nuages, le feu, la fumée et le brouillard sont des exemples de phénomènes difficiles à modéliser visuellement avec des primitives géométriques. À ce jour, il existe plusieurs méthodes de visualisation appliquées aux simulations de fumée dont le réalisme est saisissant mais qui ne permettent pas de visualisation interactive. On peut alors citer Fedkiw, Stam et Jensen (2001) qui ont utilisé une technique de placage de photons pour visualiser les effets subtils d'illumination de la fumée.

D'autres méthodes se sont avérées peu pratiques, vu la grande difficulté d'obtenir une bonne cohésion avec les primitives de rendu. Les systèmes de particules de Reeves (1983) ainsi qu'un exemple d'application sur la fumée tel que décrit par Nguyen (2004) donnent un bon réalisme et un niveau interactif acceptable mais le contrôle fin des particules est spécifié manuellement. Lorsqu'un tel contrôle fin est désiré, comme dans un environnement de production, la complexité mathématique des éléments de contrôle (champ de vitesse, de pression ou de force) et les temps de calcul requis sont prohibitifs (Treuille et al. 2003 ; Lamorlette et Foster 2002).

Parmi les deux méthodes de visualisation de données volumétriques les plus populaires, soit le lancer de rayons (type Monte Carlo, avant, arrière, distribué, bidirectionnel) et le rendu à base de textures, c'est cette dernière catégorie qui, à prime abord, s'approche le plus de nos objectifs. Les techniques de rendu volumétrique à base de textures ont les avantages suivants sur le lancer de rayons (Ikits et al. 2004) :

- elles peuvent plus facilement être intégrées avec des algorithmes traitant des polygones ;
- elles requièrent peu de passes de rendu ;
- elles offrent un bon niveau d'interactivité et ce sans sacrifier la qualité du rendu.

Le problème à résoudre est le suivant : à partir des valeurs de densité de fumée données pour tous les points de grille d'une simulation ρY_s , est-il possible de développer la capacité d'explorer les résultats volumétriques de manière interactive et obtenir une bonne représentation visuelle.

Pour ce faire, nous allons adapter au problème une technique de rendu volumétrique à base de textures directes. Le but de cette technique est de générer des images à partir de données volumétriques sans explicitement en retirer de la géométrie. (Levoy 1998). Les images sont créées par échantillonnage de plans sur le volume le long de tous les rayons dans la direction de la caméra. Le modèle émission-absorption est utilisé pour déterminer la couleur et l'opacité. La couleur C et l'opacité A sont accumulées et calculées pour n échantillons selon les équations 4.1 et 4.2. L'opacité A_j approxime l'absorption et la couleur C_i , pondérée par l'opacité, approxime l'émission et l'absorption le long du rayon entre les échantillons i et $i+1$.

$$C = \sum_{i=1}^n C_i \prod_{j=1}^{i-1} (1 - A_j) \quad (4.1)$$

$$A = 1 - \prod_{j=1}^n (1 - A_j) \quad (4.2)$$

L'idée générale de la technique est illustrée dans la figure 4.1. La première étape concerne l'initialisation du volume de données. Les valeurs de densité de suie ne seront

pas utilisées directement et une fonction de transfert unidimensionnelle est plutôt appliquée pour adapter les valeurs de densité et transformer celles-ci en propriétés optiques. Une fois transformées, ces valeurs sont entreposées dans des textures volumétriques qui sont alors utilisables par le processeur graphique.

La seconde étape porte sur la génération de la géométrie qui permettra de visualiser les textures volumétriques. Ce support, appelé géométrie proxy, coupe le volume de données en tranches orientées dans la direction du point de vue. Chaque tranche ou plan de coupe réfère, par l'intermédiaire de coordonnées de textures, à la texture volumétrique courante.

Finalement, la dernière étape concerne le rendu de la géométrie proxy utilisée ainsi que la composition de l'image finale à l'aide de nuanceurs sur le processeur graphique.

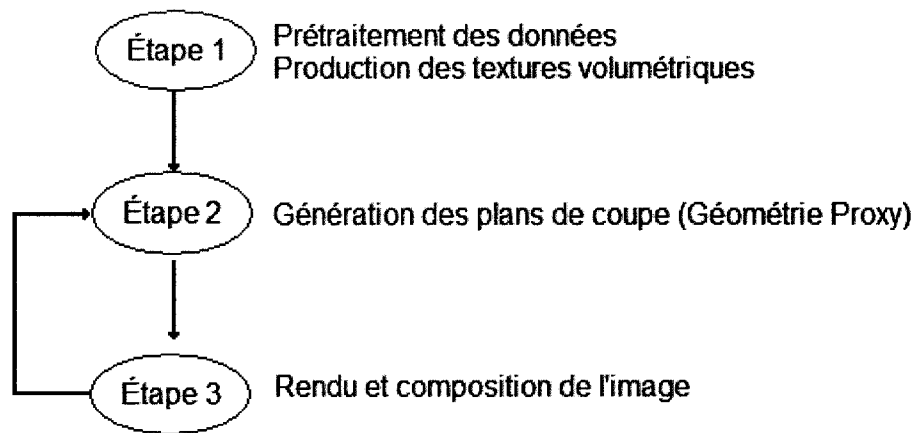


Figure 4.1 Idée générale de rendu volumétrique par plan de coupes

Les prochaines sections détaillent chacune des étapes de cette technique illustrée à la figure 4.1.

4.2 Algorithme et implémentation

4.2.1 Prétraitement des valeurs de densité

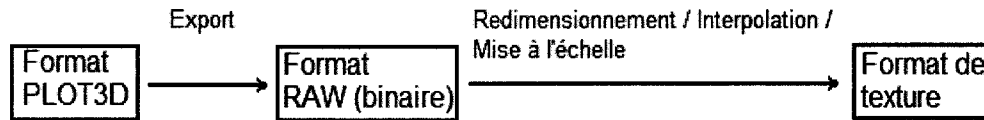


Figure 4.2 Pipeline du traitement des valeurs de densité pour chaque temps t

Un prétraitement des valeurs de densité de fumée est nécessaire pour les utiliser dans notre rendu graphique de la fumée, comme montré à la figure 4.2. Les valeurs de densité de fumée sont initialement distribuées dans un volume de données V_D de dimensions n_i , n_j et n_k au temps t et enregistrées, dans le cas de FDS, sous le format PLOT3D. Le format PLOT3D décrit, pour toute variable enregistrée Q_i (l'espèce chimique), la valeur de chaque point de grille D_{ijk} situé à i, j, k dans un volume V . D_{ijk} est une valeur réelle encodée sur 32 bits.

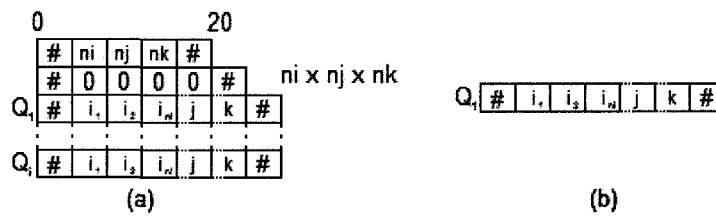


Figure 4.3 Représentation des données dans le format PLOT3D (a) et RAW (b)

Il est trivial d'exporter les données du format PLOT3D vers le format RAW, qui est simplement un format binaire brut où les données D_{ijk} sont stockées bout à bout. La figure 4.4 montre la représentation mémoire en octets d'un fichier en format PLOT3D et

RAW. Ce dernier nous permet de travailler plus facilement en mémoire (tableau unidimensionnel ou tridimensionnel) et ne contient pas les informations concernant les autres variables non utilisées pour le traitement de la fumée.

Le volume V_D produit par FDS n'est pas nécessairement dans les dimensions supportées par l'API ou le moteur pour réaliser le rendu. Ainsi, il importe d'offrir des méthodes d'interpolation et/ou de remplissage (ou « *padding* » en anglais) pour obtenir les dimensions voulues. Dans le cas présent, les textures volumétriques doivent avoir obligatoirement des dimensions en puissance de 2 pour être utilisables par le matériel graphique cible. L'outil *Teem* est utilisé pour redimensionner nos volumes de données, comme suggéré par Ikits et al (2004). Finalement, il est souvent pratique de pouvoir réaliser des mises à l'échelle linéaire pour faire ressortir les valeurs D_{ijk} qui sont autrement trop petites.

Il ne reste plus qu'à décrire la production des textures volumétriques qui servent en fait à stocker les informations de densité de fumée et qui seront alors accessibles par le matériel graphique. Le format des textures volumétriques dépend de l'espace disponible sur la mémoire vidéo pour chacune de ces textures et de la précision que l'on veut donner à la densité de fumée. L'espace disponible sur la mémoire vidéo est fonction du matériel graphique ainsi que de la charge mémoire courante utilisée par le moteur de rendu. Quant à la précision de la densité de fumée, elle ne peut pas dépasser la précision encodée sur 32 bits de la valeur flottante initiale. Ceci permet d'avoir une limite supérieure sur la taille de chaque texel, élément constitutif des textures. L'information de densité enregistrée par texel est donné par l'équation 4.3.

$$E_D = \min(d / d_{max}, 1.0) * bpc \quad (4.3) \text{ où}$$

E_D : Facteur de mise à l'échelle de la densité

d : Densité de fumée en mg / m^3

d_{max} : Densité maximale considérée en mg / m^3

bpc : Bits par canal supportés par la texture volumétrique

La densité de fumée est donc ramenée sur une échelle 0...1 à partir d'une densité maximale spécifiée. Cette valeur peut provenir d'études empiriques sur les simulations elles-mêmes ou bien simplement de la plage de valeurs de densité que l'on désire explorer et visualiser. La valeur est par la suite multipliée par bpc , qui correspond au nombre de bits par canal de la texture volumétrique. La variable bpc est généralement choisie parmi 2^8 , 2^{16} , 2^{24} ou 2^{32} et détermine en partie le format de texture qui sera utilisé pour stocker les valeurs E_D . La section 4.4 précise l'utilisation des formats de textures et les implications sur la performance et la qualité du rendu. Les textures volumétriques peuvent être générées avant la modélisation de la simulation ou pendant la simulation.

Une fois les textures volumétriques générées, à raison d'une texture volumétrique par temps t , nous pouvons passer à la prochaine étape qui est la génération de la géométrie proxy, soit les plans de coupe.

4.2.2 Génération des plans de coupe

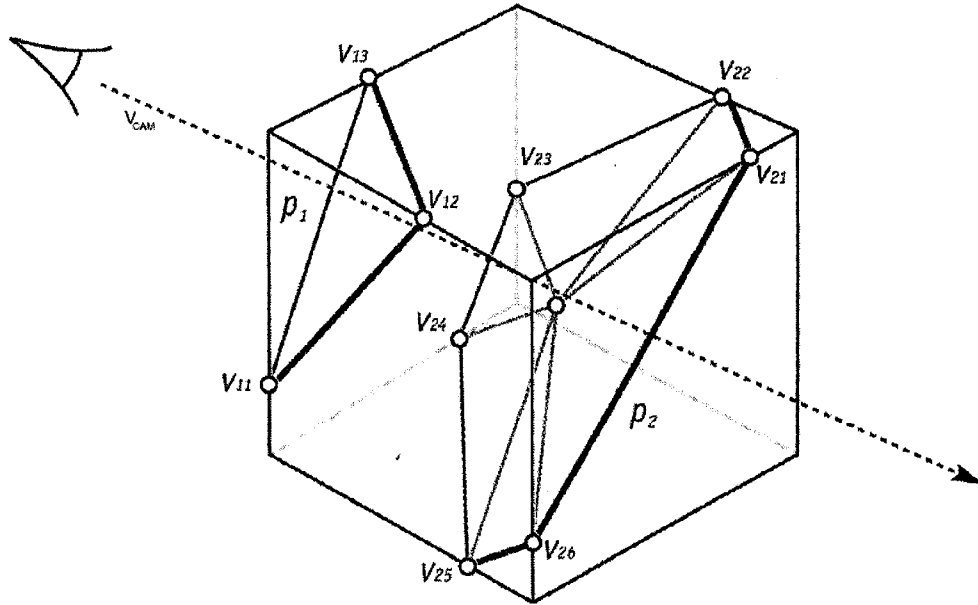


Figure 4.4 Plans de coupe sur la direction de $\vec{V}_{CAM}$, tirée de Ikits et al. (2004)

Un échantillonnage de plans de coupe sur l'espace volumétrique permet d'accumuler l'opacité correspondante à la densité de fumée. Pour ce faire, il faut découper le volume en tranches le long de la direction du vecteur de visée de la caméra. La figure 4.4 montre l'objectif poursuivi. Il s'agit de construire les plans p_i le long de la direction du vecteur de la visée de la caméra où $\vec{N}_{p_i} = -\vec{V}_{CAM}$.

Tableau 4.1 Pseudo-code de génération de plans de coupe sur CPU

Vecteur $B[8]$: Sommets de la boîte englobant l'espace volumétrique
 $Z_{\min}, Z_{\max}$: Position des plans proche et éloigné de la caméra
 Réel S : Distance normale désirée entre chaque plan de coupe
 Matrice M_{CAM} : Matrice de point de vue de la caméra
 Réel $B_{\max} = Z_{\min}$
 Réel $B_{\min} = Z_{\max}$

Étape 1 :
 Pour tous les sommets $B[i]$
 Transposer en espace caméra
 Trouver la composante z minimale et maximale ($B_{\min}$ et $B_{\max}$) de tous les $B[i]$

Étape 2 :
 La caméra regarde en direction de la boîte?
 La caméra est-elle dans la boîte?
 Le plan éloigné de la caméra est-il plus proche que le z maximal des $B[i]$?

Étape 3 :
 Vecteur $P_N(0,0,0,-1.0f)$: Normale aux plans de coupe dans l'espace de la caméra
 Entier $NbrePlan = (B_{\max} - B_{\min}) / S$

Pour tous les plans i jusqu'à $NbrePlan$
 Vecteur $P_P =$ Vecteur $(0,0,0, B_{\max} - i*S)$: Point sur le plan i
 Plan $P_i(P_N, P_P)$: Plan de coupe i spécifié par sa normale et un point
 Intersection($P_i, B[8]$) : Intersection entre P_i et la boîte englobante B
 Génération des triangles coplanaires du plan i
 Attribution des coordonnées de texture

4.2.2.1 Préparation

La génération des plans de coupe est réalisée en étapes (voir tableau 4.1) et est uniquement dépendante de la direction de la normale du plan de vue de la caméra. Ainsi, pour une position fixe, il n'est pas nécessaire de régénérer les plans de coupe, la normale des plans étant la même.

Pour construire les plans de coupe p_i , il faut d'abord avoir en main les informations sur la boîte englobante B de l'espace volumétrique ainsi que sur la caméra qui sera utilisée pour visualiser le volume. Un autre paramètre est nécessaire pour la construction des plans : la distance normale plan-plan s . Bien que l'on travaille uniquement avec la boîte englobante de l'espace volumétrique pour la construction des plans, l'espace V lui-même est une grille volumétrique de $n_i \times n_j \times n_k$ points de grille avec les données D_{ijk} distribuées sur chacun des points de grille. La figure 4.5 illustre la situation.

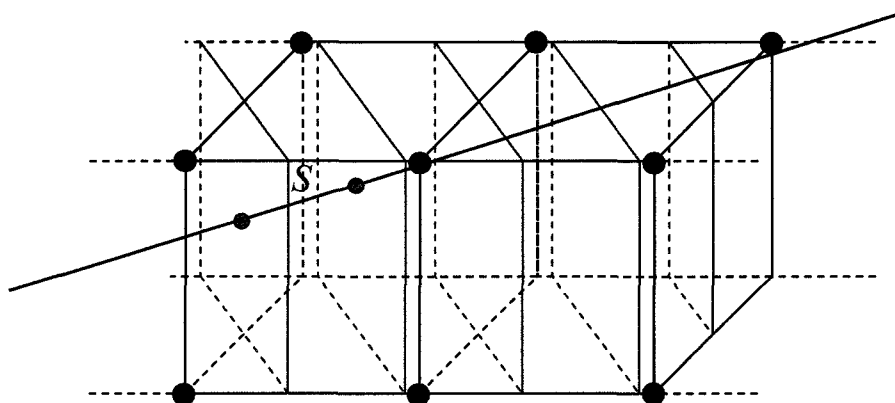


Figure 4.5 Représentation de la distance s entre deux plans de coupe à travers une grille volumétrique

Soit s_0 , la distance minimale s pour obtenir 1 plan de coupe p par élément de volume le long de la droite V_{CAM} . On définit alors s_f le rapport d'échantillonnage s / s_0 . Lorsque s_f est supérieure à 1.0, on suréchantillonne le volume de données et dans le cas contraire, on sous-échantillonne le volume. Le rapport d'échantillonnage s_f est une variable importante de cet algorithme et a des conséquences sur la qualité et la vitesse d'exécution (voir section 4.4).

4.2.2.2 Détermination de la distance minimale et maximale de la boîte englobante

La première étape de l'algorithme du tableau 4.1 est la détermination de la distance minimale et maximale de la boîte englobant le volume dans le repère de la caméra. Ceci est aisément accompli par un simple algorithme de recherche en $O(n)$ où chaque vertex $B[i]$ est d'abord transposé dans le repère de la caméra et ensuite testé par rapport aux valeurs minimale et maximale courantes sur la composante z .

4.2.2.3 Optimisations d'usage

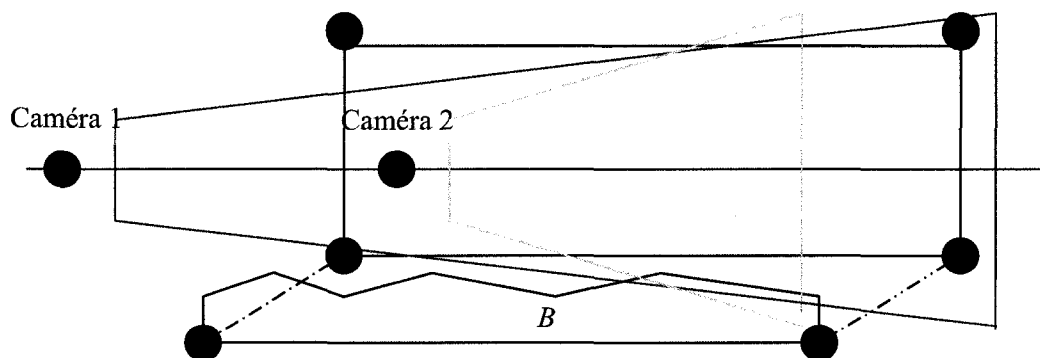


Figure 4.6 Positions de diverses caméras vis à vis la boîte englobante B

La seconde étape concerne les optimisations d'usage sur la position de la caméra. La figure 4.6 illustre les diverses positions possibles de la caméra par rapport à la boîte englobante B . Une caméra dont les plans de vue proche et éloigné enferme la boîte B (caméra 1 sur la figure 4.6) est le cas typique où tous les voxels peuvent potentiellement contribuer. Une caméra dont l'un ou l'autre des plans proche et éloigné est à l'intérieur de la boîte B (caméra 2 sur la figure 4.6) permet une optimisation au niveau de la génération des plans de coupe p_i . Ainsi, seuls les plans de coupe visibles sont nécessaires pour la visualisation : les plans derrière le plan de vue proche et en avant du plan éloigné ne sont pas utiles. Un test minimum / maximum sur les composantes z des $B[i]$ permet de réaliser ces optimisations.

4.2.2.4 Génération des plans de coupe

La troisième et dernière étape de l'algorithme décrit la construction des plans. Pour simplifier les calculs mathématiques, le repère utilisé sera celui de la caméra (U, R, V_{CAM}). Chaque plan p_i possède la même normale, soit $(0.0, 0.0, -1.0)$. Le nombre de plans à produire est simplement la distance maximale B_{MAX} moins la distance minimale B_{MIN} dans le repère de la caméra divisé par la distance normale plan-plan s . Pour complètement spécifier un plan, il est nécessaire d'avoir une normale et un point sur celui-ci. Un point peut facilement être déduit à partir de la droite le long de la normale au plan de vue de la caméra V_{CAM} . Chaque plan est espacé d'une distance normale s entre deux points le long de la droite, ce qui permet de retracer un point sur le plan là où la droite traverse le plan. La figure 4.7 illustre la position des points sur les plans.

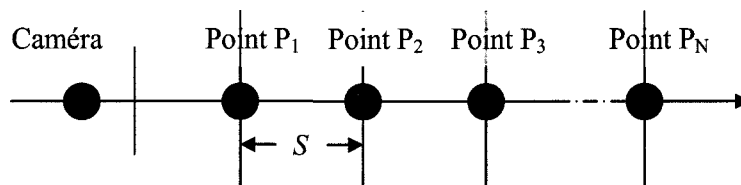


Figure 4.7 Droite normale aux plans de coupe p_i

L'intersection des plans avec la boîte englobante B est divisée en 12 tests d'intersection droite-plan, soit pour chacun des segments de la boîte. L'algorithme d'intersection droite-plan est donné à l'annexe A.1.

Une fois les tests d'intersection effectués, il faut générer les triangles coplanaires constitutifs des plans p_i à partir des points trouvés précédemment à l'aide de l'algorithme décrit au tableau 4.2. Cette étape est nécessaire, étant donné que la primitive du matériel graphique cible est le triangle.

Tableau 4.2 Algorithme de génération des triangles coplanaires d'un plan p_i

Fonction Génération des triangles coplanaires d'un plan p_i

$V[]$: Vecteur contenant les sommets d'intersection

V_{PLAN} : Sommet contenu à l'intérieur des points d'intersection sur le plan des triangles

Pour tous les points i d'intersection de V

Ramener $V[i]$ dans le repère du monde

Total += $V[i]$

$V_{PLAN} = \text{Total} / I$

Ordonnancement des sommets dans le sens des aiguilles d'une montre

Construction du tampon de sommets pour triangle en hélice

La première partie de cet algorithme trouve un sommet d'ancrage contenu à l'intérieur des sommets d'intersection, comme montré à la figure 4.8.

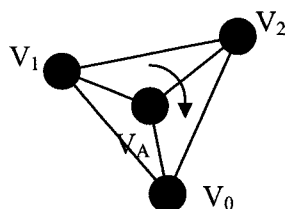


Figure 4.8 Ordonnement des sommets à partir d'un point d'ancrage (rouge) sur un plan à trois points d'intersection (bleus)

Ce nouveau sommet s'avère fort pratique. Celui-ci permet l'utilisation de la primitive à base d'hélice triangulaire (« triangle fan »), qui a pour avantage d'utiliser le minimum de sommets pour spécifier de nouveaux triangles. Soit $V_0, V_1 \dots V_n$ les sommets d'intersection triés et V_a le sommet du point d'ancrage. Pour construire les triangles spécifiés en hélice, chaque nouveau sommet V_i où $i \geq 2$ ajouté au triplet (V_a, V_0, V_1) ajoute un triangle (V_a, V_{i-1}, V_i) . Comme on veut fermer l'hélice, il faut répéter le premier sommet V_0 à la fin. Ceci donne donc un tampon de sommets de la forme $[V_a, V_0, V_1, V_2, \dots, V_n, V_0]$ lorsque les sommets sont triés dans le sens requis par l'API de rendu.

Un dernier travail est nécessaire pour représenter les triangles dans les APIs de rendu. En effet, il faut spécifier quelle face du triangle est devant et laquelle est considérée derrière. Cette étape est nécessaire étant donné que les sommets provenant des intersections droite-plan précédentes ne garantissent aucun ordre particulier. L'annexe A.2 donne le test nécessaire pour caractériser le sens de deux autres sommets à

partir d'un sommet d'ancrage. Le test n'est toutefois valide que si les sommets sont coplanaires. Il faut donc ramener les sommets dans l'espace de la caméra où les sommets d'intersection ont cette propriété, étant issus de l'intersection d'un plan avec une boîte. La liste de sommets triée dans le sens désiré à partir du point d'ancrage est obtenue en combinant le test de comparaison à un algorithme de tri à bulle. Cette liste permet de construire les triangles en hélice.

Finalement, il ne reste qu'à attribuer les coordonnées de texture (s,t,q) à chacun des sommets V_i . Les coordonnées seront utilisées pour référencer, à l'intérieur des nuanceurs, les valeurs de densité de fumée contenues dans les textures volumétriques décrites dans la section 4.2.1. Les plans de coupe servent à échantillonner un volume de données V aux limites définies par la boîte englobante B . Ainsi, si la boîte englobante B est placée pour qu'un de ses sommets coïncide avec l'origine du monde, il suffit de diviser les composantes x,y et z des sommets par la taille selon l'axe x,y et z de la boîte B . La génération des coordonnées de textures peut être faite dans le programme principal, dans le nuanceur de sommets ou encore réalisée automatiquement par l'API de rendu.

4.3 Caractérisation de l'illumination et fonction de transfert

Avant d'aborder le rendu de la fumée, il importe de caractériser celle-ci au niveau de l'illumination, c'est à dire lorsque la lumière traverse un nuage de fumée.

La fumée est un milieu participant. Pour reprendre la définition de Pharr et Humphreys (2004), il s'agit d'un milieu où des particules, distribuées dans un espace 3D, réagissent avec la lumière. Ainsi, lorsque la lumière traverse ce milieu, elle est atténuée et/ou dispersée. Un milieu participant est particulier : la radiance L n'est pas constante sur toute la distance parcourue par un rayon de lumière à l'intérieur du milieu.

Il existe deux types de milieu participant : les milieux dits homogènes et les milieux dits non homogènes. Un milieu homogène est un milieu qui possède certaines propriétés constantes sur une certaine distance. Par exemple, un brouillard londonien parfait disperse, de manière uniforme, la lumière provenant d'un lampadaire. Par contre, la fumée d'un incendie est un milieu participant non-homogène, où les propriétés optiques changent en fonction de la position p et de la direction ω de la lumière incidente.

Dans un milieu participant, il y a trois phénomènes à considérer qui affectent la distribution de lumière : l'absorption, l'émission et la dispersion de la lumière. La figure 4.9 ci-dessous montre l'effet des trois phénomènes sur un rayon de lumière et sa radiance traversant un milieu participant sur une distance d , à une position p avec une direction ω .

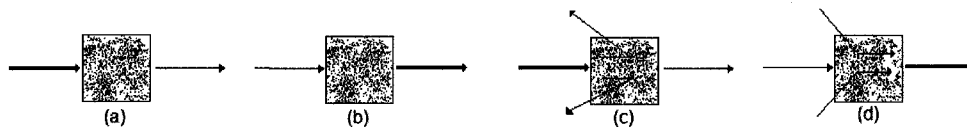


Figure 4.9 Rayon de lumière en milieu participant. (a) Absorption (b) Émission (c) Dispersion vers l'externe (d) Dispersion vers l'interne

L'absorption est généralement dépendante d'une valeur de densité. Elle est représentée dans la littérature par σ_a , qui est la fonction de densité de probabilité que la lumière soit absorbée par unité de distance. Cette probabilité est une fonction de la position p dans le volume participant et de la direction ω , cette dernière étant souvent négligée (Pharr et Humphreys (2004)). Un milieu où l'absorption prédomine est un milieu qui va obstruer la visibilité des objets derrière celui-ci et va posséder des zones sombres marquées.

L'émission augmente la radiance L d'un rayon qui traverse un milieu participant, comme montré à la figure 4.9 (b). Ceci peut être dû à une réaction chimique, nucléaire ou thermique qui convertit l'énergie en lumière visible. Ainsi, un milieu participant où l'émission prédomine aura pour caractéristique de paraître plus éclairé que les objets autour de lui.

Finalement, le dernier phénomène en milieu participant est la dispersion. Cette dispersion est de deux ordres. Comme montré à la figure 4.9 (c), un rayon est dévié, ce qui diminue la radiance du rayon sortant dans la direction incidente ω . Dans le cas de la figure 4.9 (d), un rayon sortant peut gagner en radiance dans la direction incidente ω par la déviation d'un autre rayon dans le volume participant.

Le modèle développé de représentation de la fumée est une approximation du phénomène d'absorption et d'émission. L'utilisation d'une fonction de transfert unidimensionnelle transforme les valeurs de densité enregistrées dans chaque texel (section 4.2.1) en information de couleur et d'opacité. Les valeurs E_D enregistrées dans les textures volumétriques sont utilisées pour indexer une seconde texture, qui correspond à la fonction de transfert. La fonction de transfert de la figure 4.10 transpose les valeurs E_D en information de couleur pour de la fumée blanche.



Figure 4.10 Fonction de transfert pour de la fumée blanche

De faibles valeurs E_D de densité donnent une contribution de couleur nulle (noire) alors que de fortes valeurs de densité donnent une couleur s'approchant du blanc. La fonction de transfert de la figure 4.10 ne fournit que l'information de couleur. Toutefois, il est essentiel de considérer l'opacité, ce qui permet de tenir compte du phénomène d'absorption. L'opacité est simplement les valeurs E_D transposées sur l'échelle $[0.0, 1.0]$, où la limite supérieure 1.0 correspond à la densité maximale étudiée.

4.4 Rendu et animations

Le prétraitement des valeurs de densité de fumée de la section 4.2.1 a permis d'obtenir les textures volumétriques qui sont utilisées par les nuanceurs. La génération des plans de la section 4.2.2 constituent le support d'échantillonnage, soit la géométrie proxy, qui est utilisée pour échantillonner les textures volumétriques le long de la direction du regard. La fonction de transfert de la section 4.3 sert à traduire les valeurs de densité de fumée en information de couleur et l'opacité est dépendante uniquement des valeurs de E_D . Tous les éléments sont en place pour réaliser la composition et le rendu de la fumée.

Pour ce faire, on utilise le pipeline programmable des unités graphiques. Sur le matériel cible, le pipeline est programmable au niveau des transformations des sommets (nuanceurs de sommets) et au niveau du traitement des fragments (nuanceurs de fragments). Le travail de rendu est donc divisé dans ces deux types de nuanceurs.

4.4.1 Traitement des nuanceurs

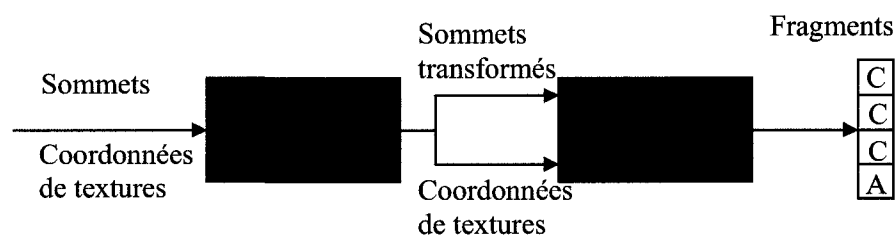


Figure 4.11 Traitement des nuanceurs de sommets et de fragments pour le rendu de la fumée

Les nuanceurs travaillent de concert (figure 4.11) pour le rendu de la fumée. Le nuanceur de sommets est simple (tableau 4.3) : tous les sommets envoyés sont transformés dans l'espace de projection. Les coordonnées de texture sont simplement passées au nuanceur de fragments et ce, sans transformation.

Tableau 4.3 Nuanceur de sommets HLSL

```
VS_OUTPUT VeffectSmokeVS( float4 inPosition : POSITION,
                          float3 inTexCoord : TEXCOORD0 )
{
    VS_OUTPUT Output;
    Output.Position = mul( inPosition, gsWorldViewProjection );
    Output.Texcoord = inTexCoord.xyz;
    Return Output;
}
```

Tableau 4.4 Nuanceur de fragments HLSL

```
PS_OUTPUT VeffectSmokePS( VS_OUTPUT In )
{
    PS_OUTPUT Output;
    // Récupération de la densité de fumée et interpolation linéaire de la valeur
    float smokeDensityA = tex3D(SmokeVolumeTextureSamplerA, In.Texcoord).a;
    float smokeDensityB = tex3D(SmokeVolumeTextureSamplerB, In.Texcoord).a;
    float smokeDensity = lerp(smokeDensityA, smokeDensityB, gLerpValue);
    // Correction Alpha
    smokeDensity = 1 - pow((1 - smokeDensity), 1.2f);
    // Récupération de la couleur de la fumée à partir de la fonction de transfert
    float3 smokeColor = tex1D(SmokeColorScaleSampler, smokeDensity);
    Output.RGBAColor = float4(smokeColor.rgb, smokeDensity);
    Return Output;
}
```

L'essentiel du travail est fait par le nuanceur de fragments du tableau 4.4. Le nuanceur de fragments reçoit en entrée les sommets transformés des plans de coupe ainsi que les coordonnées de texture du nuanceur de sommets. Ce nuanceur est appelé pour

tous les fragments générés de chacun des plans de coupe. Le premier travail du nuanceur est d'abord de récupérer l'information de densité dans les textures volumétriques chargées en mémoire vidéo à l'aide des coordonnées de texture associées à chaque sommet. Comme suggéré par Ikits et al. (2004), il faut corriger (équation 4.4) les valeurs d'opacité en fonction du rapport d'échantillonnage s_f pour obtenir une opacité constante. Par exemple, lorsque l'on utilise peu de plans de coupe, il faut rehausser l'opacité pour que l'intensité générale de la visualisation demeure la même. La correction est appliquée directement à la valeur de densité, puisque celle-ci est la valeur utilisée pour l'opacité. La variable `smokeDensity` du tableau 4.4 est par exemple ajustée avec $s_0 / s = s_f^{-1} = 1.2$ ce qui implique que le volume est sous échantillonné et que l'intensité doit être rehaussée.

$$A = 1 - (1 - A_0)^{s_0/s} \quad (4.4)$$

Ensuite, on récupère la couleur de la fumée grâce à la fonction de transfert de la figure 4.10. Cette fonction de transfert est transmise au nuanceur de fragments sous forme de texture 1D. Finalement, il ne reste qu'à constituer la couleur finale du fragment, qui correspond à la couleur de la fumée dans les canaux de couleurs RGB ainsi que l'opacité dans le canal alpha.

L'utilisation des nuanceurs des tableaux 4.3 et 4.4 ne suffit pas à réaliser la composition de l'image. Une application directe de ces nuanceurs avec un tampon Z ne permet pas d'obtenir le résultat des équations discrètes de rendu 4.1 et 4.2. À cette fin, il faut spécifier au matériel graphique de composer la couleur et l'opacité de chaque fragment pour tous les plans de coupe avec les valeurs du tampon arrière courant. Les plans de coupe ont été générés du plus éloigné au plus proche, ce qui assure un traitement correct de l'opacité. Les facteurs de composition alpha source et destination

sont respectivement de 1 et $(1 - \alpha_{\text{source}})$ pour correctement calculer la couleur et l'opacité.

4.4.2 Animations

Les simulations FDS sont calculées avec un certain pas de temps, plus grand ou égal à une seconde. Un rendu image par image ne donnera pas une visualisation agréable si le pas de temps est supérieur à $1/20$ de secondes. Une manière de contourner le problème est de visualiser la simulation environ vingt fois plus rapidement. Toutefois, il est possible de faire mieux en interpolant la densité de fumée entre deux échantillons. Pour ce faire, le nuanceur de fragments échantillonne deux textures volumétriques, soit les deux textures les plus proches du temps courant t . Le calcul de l'interpolation linéaire peut être fait dans le nuanceur de fragments directement avec l'instruction *lerp*.

L'interpolation entre les deux textures de densité permet d'obtenir un rendu plus fluide et moins brisé lors de la visualisation de la simulation. Ceci contribue grandement au réalisme de la simulation d'incendie. L'utilisation d'une seconde texture de densité pour le rendu d'une seule image a toutefois un coût non négligeable. Il faut d'abord envoyer une seconde texture volumétrique sur le GPU ce qui augmente le temps de préparation par image. De plus, cette deuxième texture est volumétrique et consomme ainsi une quantité considérable de mémoire sur le GPU. La consommation est particulièrement importante dans le cas de textures volumétriques de haute résolution et haute précision.

CHAPITRE 5 - DISTORSION VISUELLE DUE À LA CHALEUR INTENSE

5.1 Élaboration de la méthodologie de visualisation

Le 30 Mai 1999, Washington D.C., peu après minuit, un incendie fait rage dans une maison unifamiliale de la rue Cherry. Deux équipes de pompiers entrent à l'intérieur l'une au sous-sol par la porte glissante vitrée arrière et l'autre au rez-de-chaussée en utilisant l'entrée principale. L'équipe du rez-de-chaussée, équipée de boyaux, recherche à travers l'épaisse fumée noire des indices visuels de la présence d'incendie. Pendant ce temps, les pompiers au sous-sol remarquent les nombreux foyers d'incendie au sol. Ils se rendent compte rapidement que les flammes commencent à se propager le long des murs du sous-sol et sortent immédiatement. Quelques instants plus tard, un retour de flamme se produit et le sous-sol est envahi par les flammes. La première équipe évacue les lieux mais la seconde équipe au rez-de-chaussée n'a toutefois pas eu autant de chance : une intense explosion de chaleur envahit le premier étage, tuant 2 pompiers et blessant grièvement un troisième.

La seconde équipe n'a pas eu le temps de réagir. En effet, une simulation FDS faite par le NIST de la scène d'incendie a montré que la concentration d'oxygène contenu au rez-de-chaussée n'était pas suffisante pour permettre la combustion et donc aucun indice visuel laissait présager un tel phénomène à travers l'épaisse fumée noire. Il ne s'agissait en fait que de gaz très chauds, se déplaçant rapidement à travers la pièce. Cet incident réel démontre que les incendies ne sont pas que flammes et fumée et que le danger dû à la température (chaleur) intense est bien réel.

La représentation de la chaleur est donc partie intégrante de la visualisation d'une simulation d'incendie. L'objectif est de concevoir une méthode de visualisation simple qui, de manière similaire à la densité de fumée du chapitre précédent, construit un pont entre les méthodes de visualisations scientifiques traditionnelles et approximatives dans le domaine de la distorsion visuelle causée par la chaleur. La température est la valeur calculée par le simulateur FDS qui sera utilisée pour développer une méthode de visualisation simple qui s'intègre bien à un rendu à base de polygones et à la technique de visualisation de la fumée élaborée au chapitre 4.

Plusieurs méthodes de visualisations des effets de la chaleur ont été abordées au chapitre 2. Les méthodes de Nguyen (2004) et de Zhao et al. (2003) sont les plus significatives pour la présente recherche. En effet, la technique approximative de Nguyen (2004), à base de textures de bruit, est suffisamment simple et flexible pour y attacher une composante physique plus importante. Quant à la technique développée par Zhao et al. (2003), elle se rapproche du travail réalisé par FDS : une simulation physique du transfert de chaleur, une relation établie entre la température et l'effet sur la lumière ainsi que la production et l'utilisation d'un volume de données.

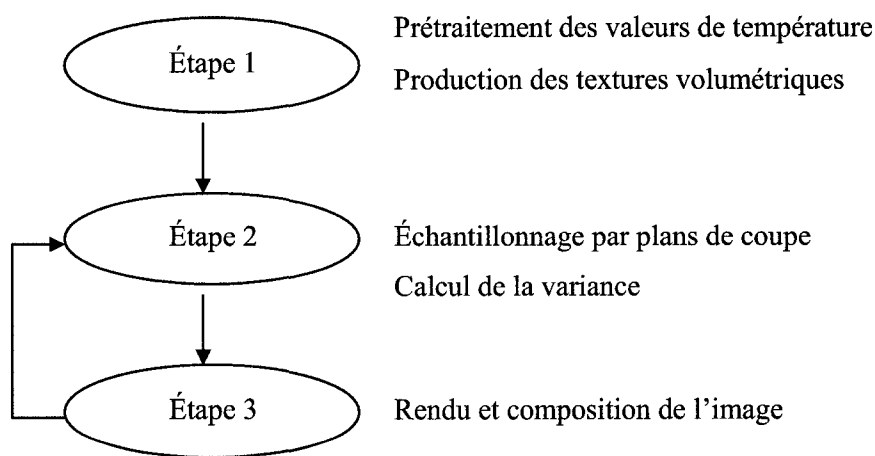


Figure 5.1 Étapes de la génération de l'effet de chaleur

La figure 5.1 résume les étapes de la génération de l'effet de chaleur. Pour réaliser l'effet de distorsion, une relation empirique est établie entre les valeurs de température produites par FDS et l'indice de réfraction de l'air. À partir de cette relation, il est possible de calculer les volumes de données contenant pour chaque point de grille les variations d'indice de réfraction par rapport à l'air ambiant. Ces volumes sont emmagasinés dans des textures volumétriques et sont repris lors de l'échantillonnage par plans de coupe. La moyenne des valeurs échantillonnées ainsi que la variance sont calculées dans la direction de la caméra sur le processeur graphique. Finalement, la variance est utilisée de pair avec une texture de bruit gaussien pour réaliser l'image de la distorsion à l'intérieur d'un nuanceur de fragments.

5.2 Hypothèses simplificatrices

Quatre hypothèses simplificatrices sont nécessaires avant d'aller plus loin. La première est que l'incendie est d'abord considéré comme une source de chaleur dans l'air ambiant. Ainsi, les différents produits issus de la combustion sont négligés. Ces produits dépendent de ce qui est consumé lors de l'incendie et leur composition est très difficile à établir à tout moment de la simulation.

La seconde hypothèse est que la pression de la simulation d'incendie demeure constante. Seuls les scénarios où il y a un minimum d'échange d'air sont considérés. Ceci permet d'exploiter la grande force du simulateur d'incendie FDS, soit les incendies à grande surface bien aérée.

La troisième hypothèse est que l'indice de réfraction de l'air absolu demeure constant pour toutes les longueurs d'ondes du spectre visible. L'indice de réfraction est en réalité fonction de la longueur d'onde. C'est ce qui produit la décomposition des couleurs (dispersion de la lumière) lorsque la lumière traverse un prisme ou une goutte d'eau : chaque longueur d'onde a un indice de réfraction différent et voyage à des vitesses différentes. La longueur d'onde qui sera utilisée est celle produite par la raie D du sodium, fréquemment utilisée comme seuil de référence.

Finalement, le phénomène de biréfringence dû à la présence de fumée n'est pas pris en compte. Le milieu d'incendie est donc, à toute fin pratique, isotrope.

5.3 Quantification de l'impact sur la lumière

L'indice de réfraction n est une grandeur qui caractérise la vitesse de propagation de la lumière à l'intérieur d'un milieu donné. L'indice de réfraction équivaut au rapport de vitesse v_A et v_B entre deux milieux A et B. Lorsque le milieu B est considéré vide, on parle alors d'indice de réfraction absolu. Cet indice est fonction de la température, de la pression, du milieu, de la direction de propagation et de la longueur d'onde. La loi de Snell-Descartes, qui décrit le comportement de la lumière à l'interface de deux milieux, peut être utilisée pour calculer la déviation des rayons de lumière au passage des interfaces.

Pour de l'air ambiant contenant 0.03% de dioxyde de carbone par volume, la dépendance de l'indice de réfraction sur la température et la pression peut être empiriquement décrite par l'équation 5.1 (Lide 2003) :

$$n = (c_1 \times 10^{-8} \times Pa \times \left(\frac{1.0 + Pa \times (61.3 - T) \times 10^{-10}}{96095.4 \times (1.0 + c_2 \times T)} \right)) + 1.0 \quad (5.1)$$

où $c_1 = 27712$ à $\lambda = 589.29$ nm et $c_2 = 0.003661$. La pression ambiante est en pascals et doit demeurer constante alors que la température est variable et décrite en degré celsius.

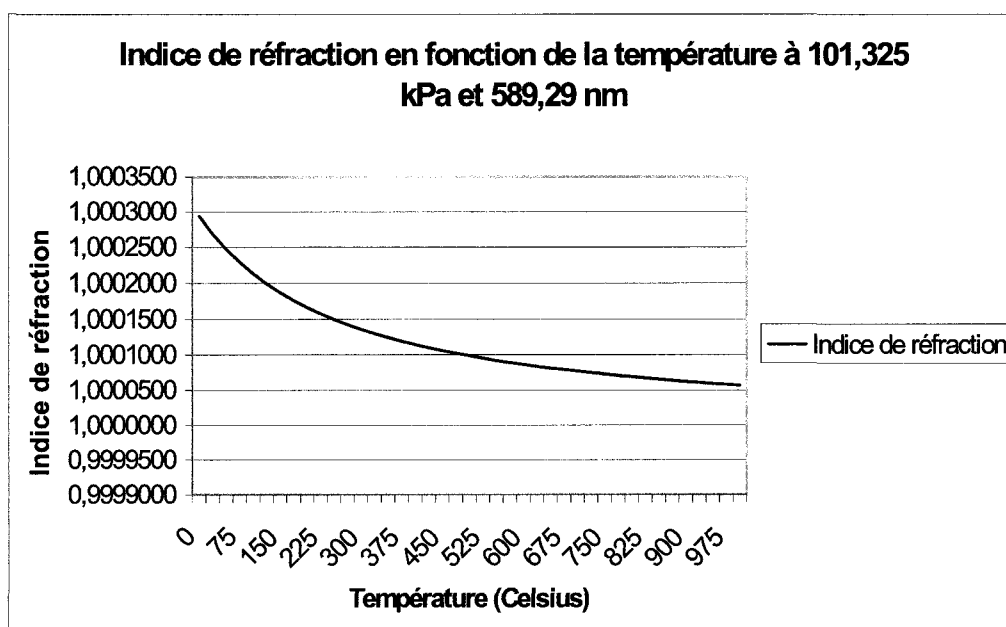


Figure 5.2 Indice de réfraction en fonction de la température et de la pression à partir de la relation de Lide (2003)

La figure 5.2 illustre la relation de Lide en fonction de la température. L'indice de réfraction tend vers 1 en fonction de la température, qui est l'indice de réfraction du vide. Ainsi, plus l'air devient chaud, plus l'indice de réfraction diminue et plus grande est la différence par rapport à l'air ambiant. La variation de l'indice est par ailleurs très faible : elle se situe en deçà de trois dix millièmes.

Tableau 5.1 Valeurs d'indice de réfraction pour différents matériaux pour $\lambda = 589.29$ nm, $T = 0^\circ\text{C}$ et $P = 101,325$ kPa

Matériaux	Indices de réfraction
Air	1,000293
Dioxyde de carbone	1,00049
Eau	1,333
Verre	1,412

Le tableau 5.1 fournit pour fin de comparaison la valeur des indices de réfraction d'autres éléments communs.

La formule de Lide permet de calculer l'indice de réfraction en fonction de la température et la loi de Snell-Descartes peut être utilisée pour avoir une idée de la déviation des rayons de lumière. Un modèle de rendu à base de lancer de rayons peut ainsi calculer, pour un point de vue donné, les déviations qui résultent de la variation de l'indice de réfraction, comme montré dans Zhao et al. (2003). Toutefois, la technique du lancer de rayons est lente et difficile à intégrer avec l'effet utilisé pour la fumée. De plus, la résolution de la loi de Snell-Descartes en nuances de fragments taxe grandement les performances. L'idée est donc d'exploiter une mesure qui décrit qualitativement le degré de variabilité des déviations. Cette mesure est la variance des différences d'indice entre l'air ambiant et le point considéré dans la direction de la caméra.

La variance est une mesure qui sert à caractériser la dispersion d'un échantillon. Ainsi, il est possible de qualifier le trajet d'un rayon droit allant de l'œil à l'objet

considéré. Une variance nulle indique que les différences d'indice sont identiques et qu'un rayon partant de l'œil demeure droit, celui-ci étant considéré comme voyageant dans un milieu isotrope. Dans le cas inverse, une forte variance signifie une grande variation dans les valeurs d'indices et la présence de plusieurs couches ou interfaces. La variance est calculée avec l'équation 5.2 :

$$V(\Delta E_N) = \frac{\sum_{i=1}^n (\Delta E_{N_i}^2) - \frac{[\sum_{i=1}^n (\Delta E_{N_i})]^2}{n}}{n} \quad (5.2)$$

où ΔE_N est la différence de l'indice de réfraction par rapport à l'air ambiant et $V(X)$ la variance calculée des échantillons. La variance est calculée en temps réel sur le processeur graphique à partir de l'échantillonnage par plans de coupe.

5.3 Algorithme et implémentation CPU

5.3.1 Prétraitement des valeurs de températures

La température est la variable source utilisée pour mesurer la force de la distorsion de chaleur appliquée. Le processus de traitement est identique au traitement pour la fumée illustré à la figure 4.3, la température provenant des mêmes données sources dans le même format. Les valeurs de température sont extraites des fichiers PLOT3D et transposées dans le format RAW, comme indiqué à la figure 4.4. Le volume de température est converti en volume de valeurs d'indice de réfraction par l'équation 5.2 . La relation permet d'obtenir les valeurs n_{ijk} pour toutes les données de points de grille i,j,k du volume de données.

L'indice de réfraction de l'air est borné fermé [1.0 , 1.000293]. En effet, une valeur supérieure indique que l'air ambiant ne serait plus de l'air. Une valeur inférieure à 1.0 indiquerait que la lumière se déplace plus vite dans le milieu que la vitesse de la lumière, ce qui est encore à ce jour impossible.

$$\Delta E_N = (1.000293 - n_{ijk}) / (0.000293) * bpc \quad (5.3)$$

ΔE_N : Variation de l'indice de réfraction par rapport à l'air ambiant

n_{ijk} : Indice de réfraction au point de grille i,j,k

bpc : Bits par canal supportés par la texture volumétrique

L'équation 5.3 permet de calculer l'information qui est entreposée à l'intérieur des textures volumétriques, à raison d'une texture par pas de temps t , pour permettre l'utilisation de l'information dans les nuanceurs. Cette équation calcule la différence de l'indice de réfraction par rapport à celui de l'air ambiant pour chaque point de grille i,j,k . Les valeurs de ΔE_N sont dans l'intervalle $[0,1]$ et sont multipliées par bpc , allant d'une variation nulle (air) à la plus grande variation possible (vacuum). L'information peut être enregistrée sur des textures volumétriques de 8, 16, 24 ou 32 bits dépendant des capacités matérielles disponibles et de la précision voulue.

5.3.2 Échantillonnage du volume de données

Pour procéder dans le rendu de l'effet de chaleur, les volumes de données contenus dans les textures volumétriques doivent être échantillonnés. L'échantillonnage est réalisé grâce à des plans de coupe parallèles, les mêmes que ceux construits pour la densité de fumée de la section 4.2.2. Les plans de coupe réfèrent, grâce aux coordonnées de textures, aux volumes de données et sont orientés dans la direction de la caméra. Les plans permettent de récupérer des valeurs discrètes dans le volume de données des valeurs de ΔE_N dans l'espace. La variance de l'équation 5.2 peut alors être calculée dans un nuanceur de fragments.

La distance plan-plan s est tout aussi importante, car celle-ci a un impact sur le nombre d'échantillons, la précision de l'échantillonnage et conséquemment les performances.

5.4 Rendu, composition et animations

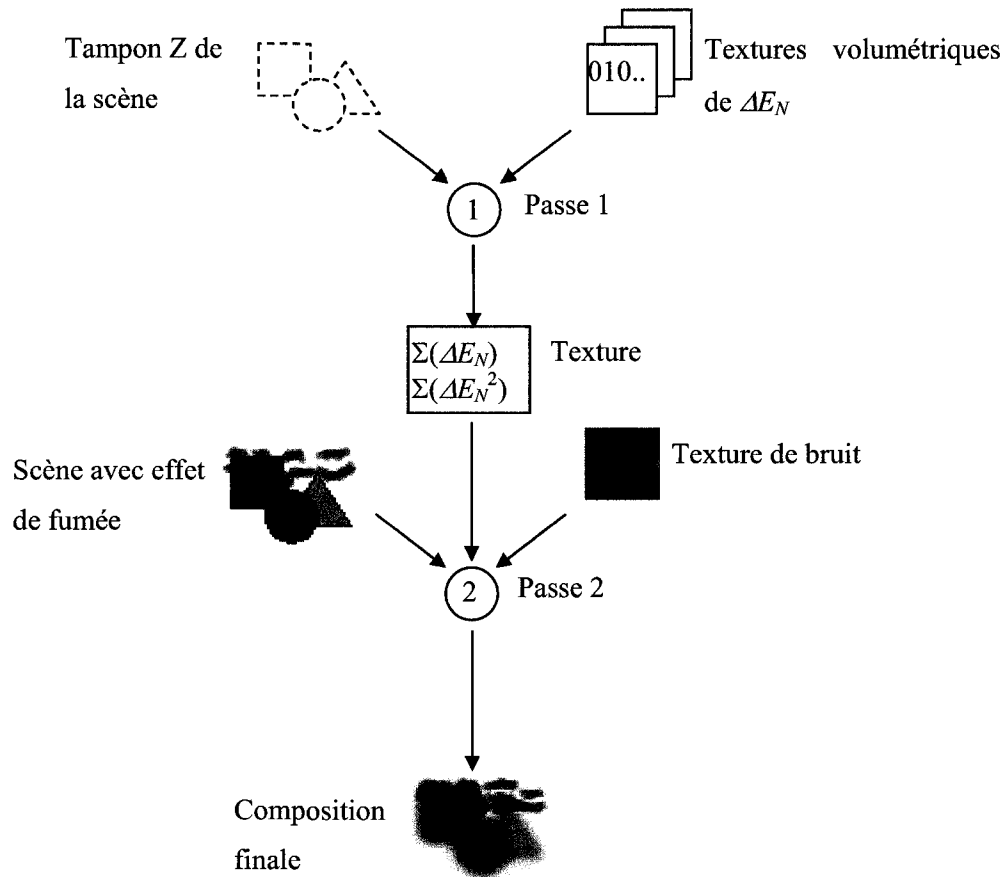


Figure 5.3 Rendu de l'effet de distorsion de chaleur

Le rendu de l'effet de chaleur, illustré à la figure 5.3, s'effectue en plusieurs passes et nécessite plusieurs ressources disponibles au préalable. La première passe effectue le calcul sur le processeur graphique des sommations nécessaires au calcul de la variance de l'équation 5.2. Cette passe requiert en entrée le tampon Z de la scène ainsi que les textures volumétriques contenant les valeurs de ΔE_N . Elle produit en sortie une texture qui sera utilisée lors de la seconde passe. La seconde passe reprend la scène avec

le rendu de fumée, la sortie de la passe précédente ainsi qu'une texture de bruit gaussien pour accomplir la composition de l'image finale. Les prochaines sections décrivent en détails chacune des passes.

5.4.1 Première passe : calculs de $\Sigma(\Delta E_N)$ et $\Sigma(\Delta E_N)^2$

La variance des valeurs de ΔE_N est la mesure qualitative nécessaire à l'effet de chaleur. L'équation 5.2 demande l'évaluation de deux moyennes et ceci nécessite le calcul de deux sommations, soit $\Sigma(\Delta E_N)$ et $\Sigma(\Delta E_N)^2$. La première passe exploite donc le matériel graphique pour calculer, par fragments, ces sommations. Le tableau 5.2 montre les nuanceurs de sommets et de fragments utilisés.

Tableau 5.2 Nuanceurs HLSL de sommets et de fragments utilisés pour la première passe de l'effet de chaleur

```
VS_OUTPUT VeffectHeatVS( float4 inPosition      : POSITION,
                        Float3 inTexCoord      : TEXCOORD0 )
{
    VS_OUTPUT Output;
    Output.Position = mul( inPosition, gsWorldViewProjection );
    Output.Texcoord = inTexCoord.xyz;
    return Output;
}

PS_OUTPUT VeffectHeatPS( VS_OUTPUT In )
{
    PS_OUTPUT Output;
    float refractionIndexA = tex3D(RefractionVolumeTextureSamplerA,In.Texcoord).a;
    float refractionIndexB = tex3D(RefractionVolumeTextureSamplerB,In.Texcoord).a;
    float refractionFraction = lerp(refractionIndexA,refractionIndexB,gLerpValue);
    Output.RGBAColor=float4(0.00390625f,refractionFraction*refractionFraction,refractionFraction,1.0);
    return Output;
}
```

Les sommets des plans de coupe ainsi que les coordonnées de texture correspondante dans le volume de données sont envoyés au nuanceur de sommet. Celui-ci ne sert qu'à transposer les sommets dans l'espace de projection et à passer les sommets modifiés et les coordonnées de texture au nuanceur de fragments.

Le nuanceur de fragments utilisent deux échantillonneurs sur les deux textures volumétriques bornant l'intervalle de temps t . Les deux échantillons `refractionIndexA` et `refractionIndexB` servent à calculer la valeur interpolée linéaire `refractionFraction`. Cette dernière variable représente la valeur de ΔE_N au temps t pour un fragment sur un plan de coupe visible.

Une manière de calculer les sommations demandées par l'équation 5.2 est d'exploiter l'opération de mélange alpha du matériel de l'équation 5.4 . Pour ce faire, il suffit de spécifier que les couleurs des fragments de destination $C_{Destination}$ (tampon arrière) et source C_{Source} (fragments du plan de coupe i) doivent être toutes deux considérées avec des paramètres de mélange B_{Source} et $B_{Destination}$ de 1.0. Cette opération est appelée pour tous les fragments valides, c'est à dire tous les fragments qui passent le test de profondeur.

$$C_{Finale} = C_{Source} * B_{Source} + C_{Destination} * B_{Destination} \quad (5.4)$$

Il devient alors facile de calculer les sommations nécessaires. Ainsi, chaque fragment dessiné donne en sortie le quadruplet (1 / bpc , `refractionFraction*refractionFraction`, `refractionFraction`, 1). La première passe produit en sortie une texture où la composition des pixels est la suivante : le canal rouge contient le nombre de plans dessinés pour chaque fragment, le canal vert $\Sigma(\Delta E_N)^2$ et le canal bleu

$\Sigma(\Delta E_N)$. Le tableau 5.2 donne le code HLSL pour une texture de sortie où chaque canal a une précision de 32 bits.

Finalement, pour que les calculs de sommations soient justes, le tampon Z de la scène 3D qui sera composée lors de la deuxième passe doit préalablement être disponible ou calculé. En effet, le tampon de profondeur permet de déterminer si un fragment est visible ou non. Un fragment non visible, par exemple un plan de coupe à travers un objet 3D, ne doit pas compter lors du calcul des sommations car seuls les rayons de lumière droits en direction de la caméra sont considérés. Le tampon Z est aisément obtenu en effectuant un rendu de la scène 3D désiré sur le tampon arrière en réinitialisant les canaux de couleurs à la couleur noire (0.0) .

5.4.2 Deuxième passe : bruitage et composition

La deuxième passe utilise la texture des sommations produites lors de la première passe pour composer l'image finale. Cette passe requiert la scène 3D composée avec en option l'effet de fumée, la texture des sommations calculées ainsi qu'une texture de bruit gaussien. Cette passe a pour objectif le calcul de la variance $V(\Delta E_N)$ et le rendu de la distorsion due à la chaleur.

La variance peut être aisément calculée au nuanceur de fragments, donné au tableau 5.3. Un échantillonneur récupère les valeurs de la texture contenant les sommations décrites précédemment et affecte les valeurs à la variable *heatData*. Par la suite, il est possible de calculer chaque moyenne de l'équation 5.2 en utilisant le nombre de plans contenus dans le canal rouge et les sommations contenues dans les

canaux vert et bleu. Finalement, la variance par fragments est calculée à partir de l'équation 5.2 et la valeur est entreposée dans la variable *pixelVariance*.

Tableau 5.3 Nuanceurs HLSL de sommets et de fragments utilisés pour la seconde passe de l'effet de chaleur

```
VS_OUT HeatDistortionVS( float4 inPosition : POSITION,
                        float2 inTexCoord : TEXCOORD0 )
{
    VS_OUT Output;
    Output.Position = inPosition;
    Output.TexCoord0 = inTexCoord;
    return Output;
}

float4 HeatDistortionPS(VS_OUT In) : COLOR
{
    float4 heatData = tex2D(RenderedHeatTextureSampler, In.TexCoord0);
    float4 noisePixel = tex2D(NoiseTextureSampler, In.TexCoord0);

    float nbPlane = heatData.r*256.0f;
    float squareDiff = heatData.g;
    float diff = heatData.b;
    float diffMean = heatData.b / nbPlane;
    float pixelVariance = (squareDiff - diff*diffMean) / (nbPlane);
    noisePixel *= pixelVariance;
    float4 modulatedNoisePixel = tex2D(RenderedSceneTextureSampler, In.TexCoord0 +
noisePixel.xy);

    return modulatedNoisePixel;
}
```

Il ne reste qu'à réaliser l'effet de distorsion. La distorsion est un effet visuel qui embrouille la vision. Les objets perdent leur rectitude et parfois les éléments se courbent, comme lors de mirage. Une manière de réaliser cet effet sur nuanceur est d'exploiter une texture de bruit gaussien et de l'utiliser pour appliquer une distorsion sur une image donnée, comme illustrée à la figure 5.4



Figure 5.4 Application d'un bruit gaussien sur une image

Un bruit gaussien appliquée à la totalité de l'image n'a toutefois pas d'intérêt, car seulement des régions spécifiques de la scène doivent être brouillées. Le bruit gaussien est donc modulé avec un masque. Ce masque est la variance $V(\Delta E_N)$ calculée par pixel.

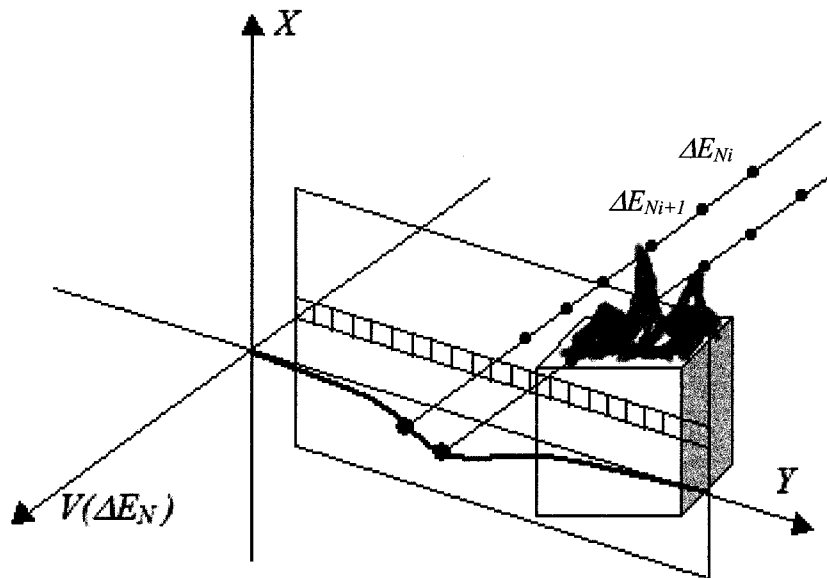


Figure 5.5 Variance par pixel des ΔE_{Ni}

La figure 5.5 illustre pourquoi la variance peut être utilisée comme masque au bruit gaussien. Soit X, Y la résolution en pixel de l'image source et $V(\Delta E_N)$ la variance des échantillons ΔE_{Ni} . La variance décrit le niveau de variabilité d'un groupe d'échantillons. Ainsi, des rayons droits traversant des zones de différents indices de réfraction, donc de chaleur, vont induire une variance non nulle. Il s'agit de zones qu'il

faut brouiller grâce à la texture gaussienne. Des rayons traversant des zones où les indices varient peu impliquent que la variance du groupe d'échantillon sera faible (ou nulle) et indiquent qu'il n'est pas nécessaire de brouiller l'image, les rayons ayant toutes les chances de demeurer droits.

L'application du masque et de la texture de bruit gaussien est un traitement fait sur les pixels d'une image et non sur les fragments de la géométrie. Ceci implique l'utilisation d'un rendu s'effectuant dans l'espace écran. Un traitement dans l'espace écran ne tient compte que de la résolution du rendu et chaque fragment correspond exactement à un pixel. Ceci permet de calculer une variance par pixel et de faire un traitement par pixel de l'image source, soit la scène 3D avec l'effet de fumée. Pour permettre ce type de rendu, un rectangle non transformé aux coordonnées $(-0.5, -0.5, X - 0.5, Y - 0.5)$ est fourni au nuanceur de sommets. Un décalage de 0.5 pixel est nécessaire en rendu 2D pour s'assurer que chaque texel correspond correctement à un pixel et ainsi éviter une distorsion de la texture de l'image source.

Le rectangle envoyé au nuanceur de sommets du tableau 5.3 n'est pas transformé, car il est déjà dans le système de coordonnées de l'écran. L'effet de distorsion peut maintenant être calculé dans le nuanceur de fragments du tableau 5.3. La variance calculée par pixel *pixelVariance* est modulée avec l'échantillon de la texture de bruits *noisePixel*. La valeur modulée *modulatedNoisePixel* est utilisée pour décaler les coordonnées de textures de l'image source, ce qui a pour effet de brouiller (ou non) les pixels. Ce décalage indique à l'échantillonneur de l'image source d'aller chercher un pixel avoisinant.

5.4.3 Animations

Parallèlement au rendu de la densité de fumée, un rendu image par image ne donnera pas une visualisation agréable si le pas de temps n'est pas suffisamment petit. Pour adoucir la visualisation en temps réel, les deux textures des ΔE_N bornant l'intervalle du temps t courant sont échantillonnées et les échantillons sont interpolés linéairement entre les bornes. Le calcul de l'interpolation linéaire est fait dans le nuanceur de fragment du tableau 5.2. Il est alors possible d'obtenir un rendu moins brisé lors de la visualisation en temps réel de la simulation.

CHAPITRE 6 - RÉSULTATS

6.1 Moteur de rendu PHX

Le moteur PHX est un moteur de rendu sur nuanceurs à base de polygones qui a été développé dans le cadre du présent travail pour supporter les effets visuels de la fumée et de la distorsion de chaleur. Celui-ci répond à l'objectif intégrateur énoncé lors de l'introduction. Il intègre les algorithmes de traitements des fichiers de simulations ainsi que les algorithmes de visualisation décrit aux chapitres 4,5 et en annexes. Pour produire les images de la simulation, le moteur requiert deux éléments importants : une scène 3D et les fichiers d'une simulation FDS.

La scène 3D est produite dans le logiciel de modélisation 3dStudioMax 8. Celle-ci est une version augmentée de l'environnement simulé avec le simulateur d'incendie FDS. Un plugiciel permet la spécification des matériaux, des textures et des nuanceurs associés à la géométrie. L'exécution d'un script exporte les données de la scène en format binaire et/ou XML pour notre moteur PHX. Les fichiers de simulation sont produits par FDS et ceux-ci sont prétraités directement dans le moteur en fonction de l'effet (voir les sections se rapportant au prétraitement pour chacun des effets des chapitres précédents).

Le moteur réalise la visualisation grâce à la librairie DirectX 9.0c. Un module basé sur les interfaces d'effets DirectX permet la visualisation de la scène et des effets visuels liés à la fumée et à la distorsion entièrement sur GPU et de manière interactive.

L'utilisateur se promène à l'intérieur de la scène et peut ainsi obtenir des résultats selon différents points de vue.

Ce moteur permet donc à son utilisateur d'analyser et d'étudier les résultats de la simulation d'incendie de manière interactive. Cet outil a été utilisé pour produire les résultats qui sont présentés dans les prochaines sections.

6.2 Plateforme et simulations

Tableau 6.1 Plateforme de visualisation

Spécification	
Système d'exploitation	Windows XP Édition Familiale
Processeur	Intel Pentium M cadencé à 1.7 GHz
Mémoire	1,5 Go cadencé à 667 MHz
Carte Vidéo	Mobility Radeon ATI X600
Mémoire Vidéo	64 Mo
Version DirectX	9.0c SDK version décembre 2005

Le tableau 6.1 donne les spécifications matérielles et logicielles avec lesquelles les résultats ont été produits. Le tableau 6.2 décrit les paramètres des simulations FDS utilisées pour les deux scénarios d'incendie testés.

Tableau 6.2 Scénarios simulés sur FDS

	Scénario C107	Scénario Cuisine
Taille de la grille (Lxlxh)	120x107x33	64x128x64
Dimension (Lxlxh)	18m x 16.05m x 4.95m	5.15m x 9.9m x 3.15m
Durée de la simulation	479 secondes	300 secondes
Variables simulées	Densité de fumée Température	Densité de fumée Température
Valeurs aux points de grille	Oui	Oui
Taux de dégagement de chaleur du foyer d'incendie	50000 HRRPUA	500 HRRPUA

Deux scénarios d'incendie ont été simulés sur le simulateur FDS. Le scénario C107 décrit un incendie se déroulant dans une classe type de l'École polytechnique de Montréal. Les flammes prennent naissance sous le bureau du professeur et les bouches d'aération demeurent ouvertes assurant un apport d'oxygène constant. Le scénario Cuisine décrit un incendie ayant lieu dans une cuisine. La cuisinière est l'élément déclencheur de l'incendie et les flammes se propagent le long des armoires.

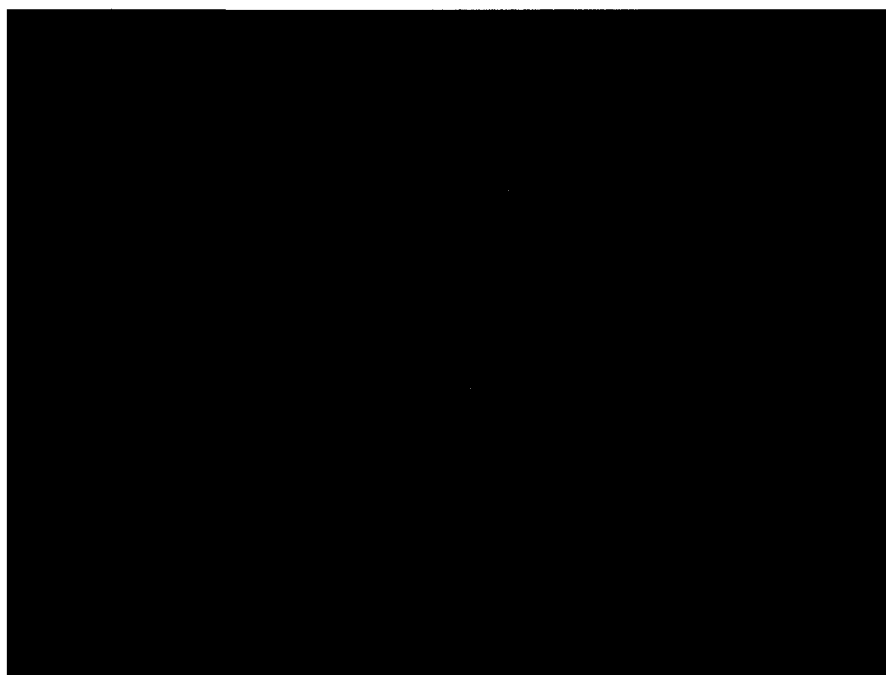


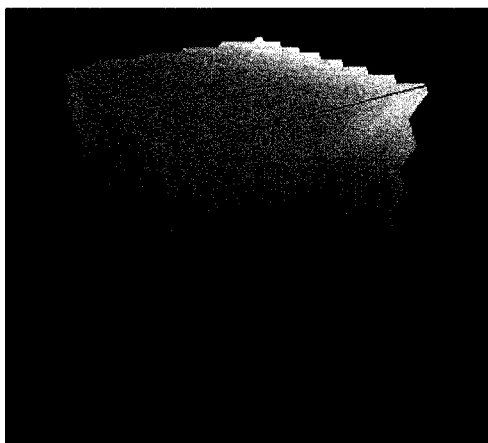
Figure 6.1 Rendu C107



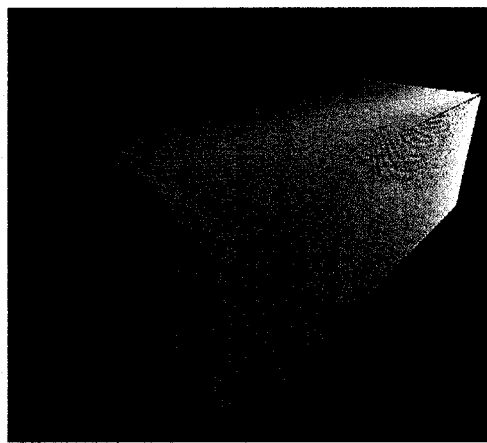
Figure 6.2 Rendu Cuisine

Les figures 6.1 et 6.2 montrent un rendu dans le logiciel 3DSMAX des deux scènes simulées. La scène C107 de la figure 6.1 est visualisée à murs ouverts mais simulée à murs fermés. La scène Cuisine, montrée à la figure 6.2, est une petite cuisine et une salle à dîner séparée. Les modélisations géométriques de ces deux scénarios ainsi que les simulations ont été réalisées par Eddy Petro du département de génie mécanique à l'École polytechnique de Montréal.

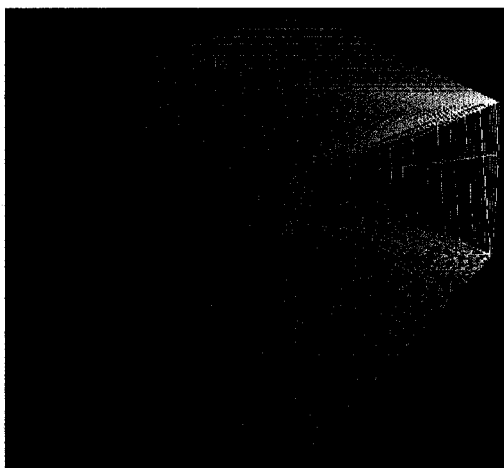
6.3 Plans de coupe



(a)



(b)



(c)

Figure 6.3 Plans de coupe (a) Volume 5m x 5m x 5m $s_f = 1.0$ (b) Volume C107 $s_f = 1.0$
(c) Volume C107 $s_f = 0.25$

Les plans de coupe échantillonnent les volumes de données en utilisant les textures volumétriques appropriées pour les effets de fumée et de chaleur des chapitres 4 et 5. La figure 6.3 illustre plusieurs séries de plans de coupe pour différents volumes. La figure 6.3 (a) montre le volume témoin cubique de dimension $5m \times 5m \times 5m$ et une fréquence d'échantillonnage s_f de 1.0 sur un volume de $5 \times 5 \times 5$. La couleur des plans de l'image (a) est fonction des coordonnées de textures attribuées aux sommets des plans. La couleur varie de $(0.0, 0.0, 0.0)$ à $(1.0, 1.0, 1.0)$ en fonction des coordonnées de texture (s, t, q) .

La figure 6.3 (b) et (c) illustre deux séries de plans de coupe faits sur le volume de la simulation C107 à deux fréquences d'échantillonnage différentes à deux points de vue différents. Les plans sont dessinés en mode fil de fer et font toujours face à la caméra. Tous les voxels traversés par la visée de la caméra à travers le volume sont échantillonnés au moins une fois dans l'image (b) alors que seulement environ le quart des voxels sont échantillonnés sur l'image (c).

6.4 Effets de fumée

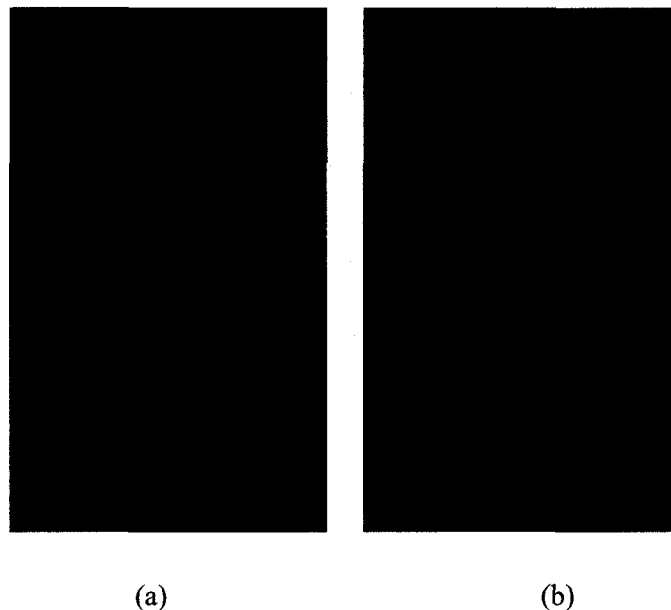


Figure 6.4 Textures alpha 8 bits de densité de fumée simulation C107 128x128x32 pixels au temps 200.01s (a) tranche 8 (b) tranche 9

L'effet de fumée utilise des textures contenant la densité de fumée. La figure 6.3 montre plusieurs tranches (plan x-y) d'une texture de densité de fumée de la simulation C107 à 200.01s. Un format alpha 8 bits a été choisi pour son faible coût en espace mémoire et les textures volumétriques ont été redimensionnées à la puissance de 2 la plus proche. La figure 6.4 (a) et (b) illustre bien le foyer d'incendie de la simulation C107, soit les zones où la densité de fumée est la plus élevée.

Des résultats similaires ont été obtenus pour le second scénario. La taille des textures volumétriques s'adapte à la taille du volume de la simulation cuisine. Les figures 6.5 (a) et (b) montrent respectivement les tranches 29 et 31 de la texture de densité de fumée générée au temps 130.0s. Les deux cercles présentés sur ces deux images indiquent une densité de fumée très forte et ceci correspond à la position de la cuisinière, soit le foyer d'incendie. La dernière image montre comment la densité est répartie dans les zones supérieures du volume. Les contours de la cuisine sont visibles sur l'image (c) si on regarde attentivement les régions qui n'y sont pas complètement noires.

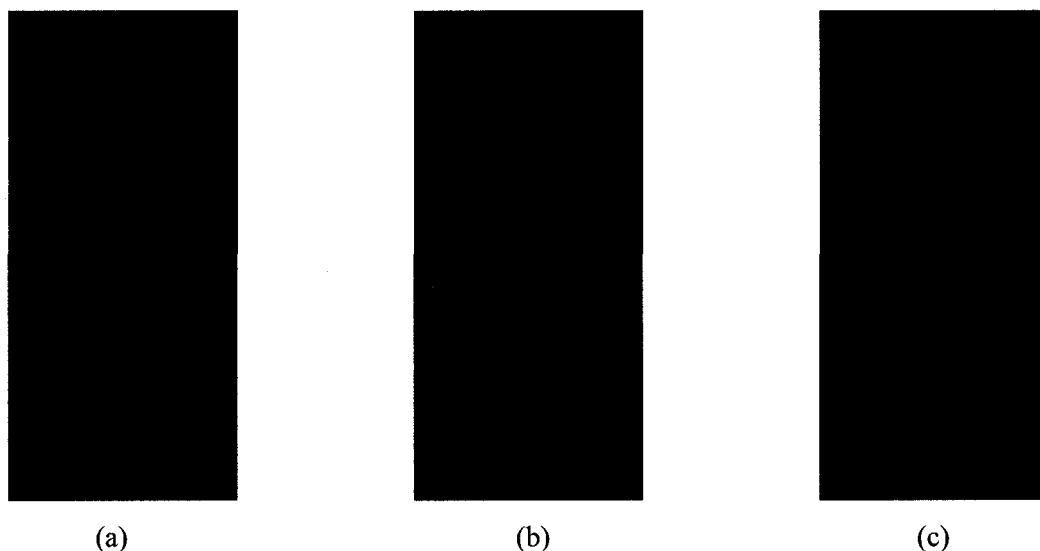


Figure 6.5 Textures alpha 8 bits de densité de fumée simulation Cuisine 64x128x64 pixels au temps 130.0s (a) tranche 29 (b) tranche 31 (c) tranche 54

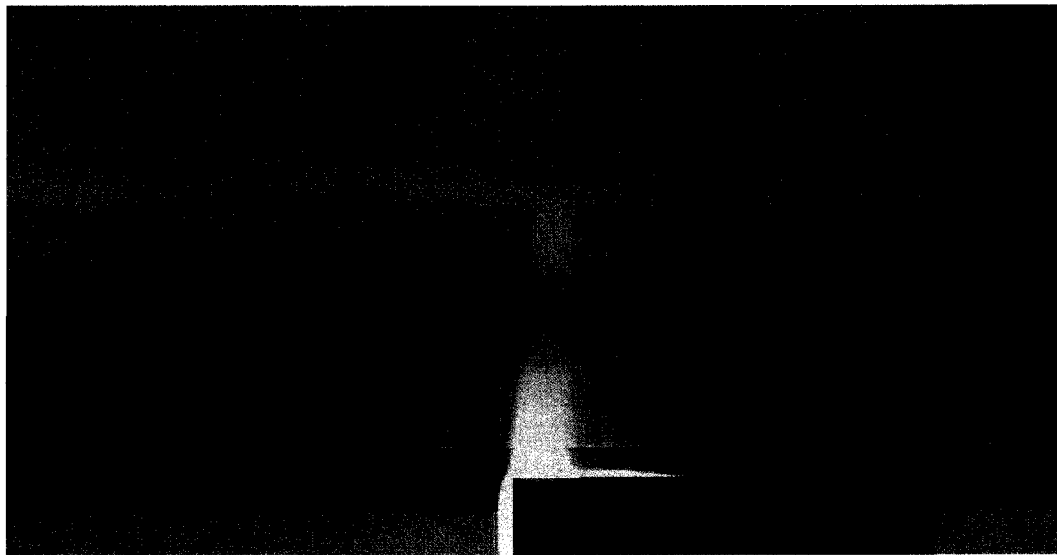


Figure 6.6 Échelle de couleurs de l'effet de fumée

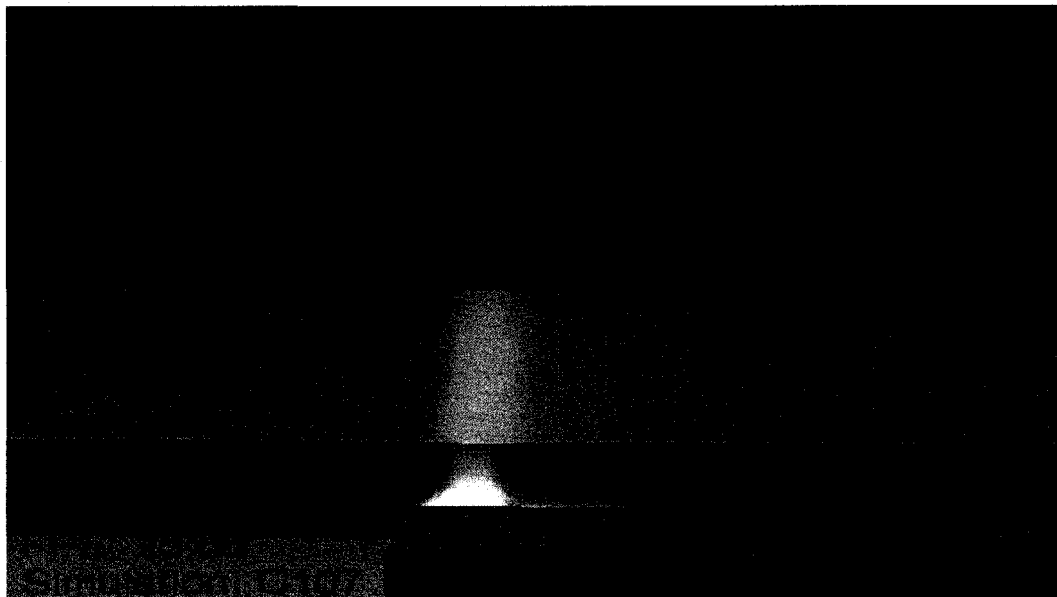


Figure 6.7 Effet de fumée simulation C107 textures 8 bits alpha 128x128x32 avec une résolution 640x480 vue externe sans scène $sf = 2.0$

Les figures 6.7 et 6.8 présentent les résultats obtenus pour l'effet de fumée appliqué à la simulations C107 pour des fréquences d'échantillonnage de 2.0, 1.0 et 0.25. La figure 6.7 montrent une vue externe sans scène avec volume suréchantillonné qui permet de bien voir les dimensions du volume de fumée avec l'échelle de couleur de la figure 6.6. Les figures 6.8 a,b et c montrent pour différentes fréquences d'échantillonnage l'intégration de l'effet de fumée avec les scènes 3D de la simulation C107. La limite de densité de fumée a été fixée à $2000 \text{ mg} / \text{m}^3$.



(a)



(b)

Figure 6.8 Effet de fumée simulation C107 textures 8 bits alpha 128x128x32 avec une résolution 640x480 (a) Vue interne $s_f = 2.0$ (b) Vue interne $s_f = 1.0$ (c) Vue interne $s_f = 0.25$

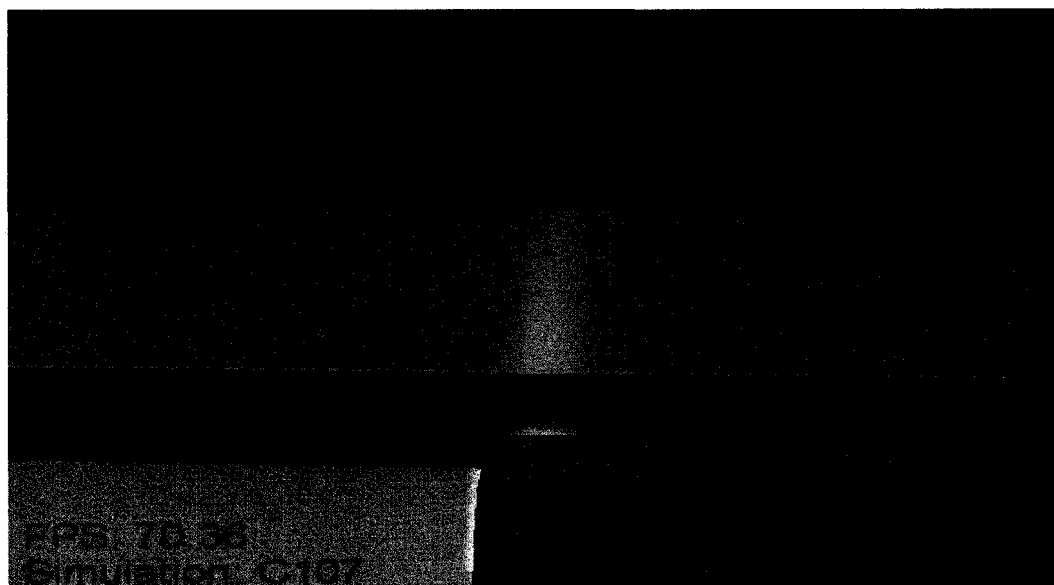


Figure 6.8c

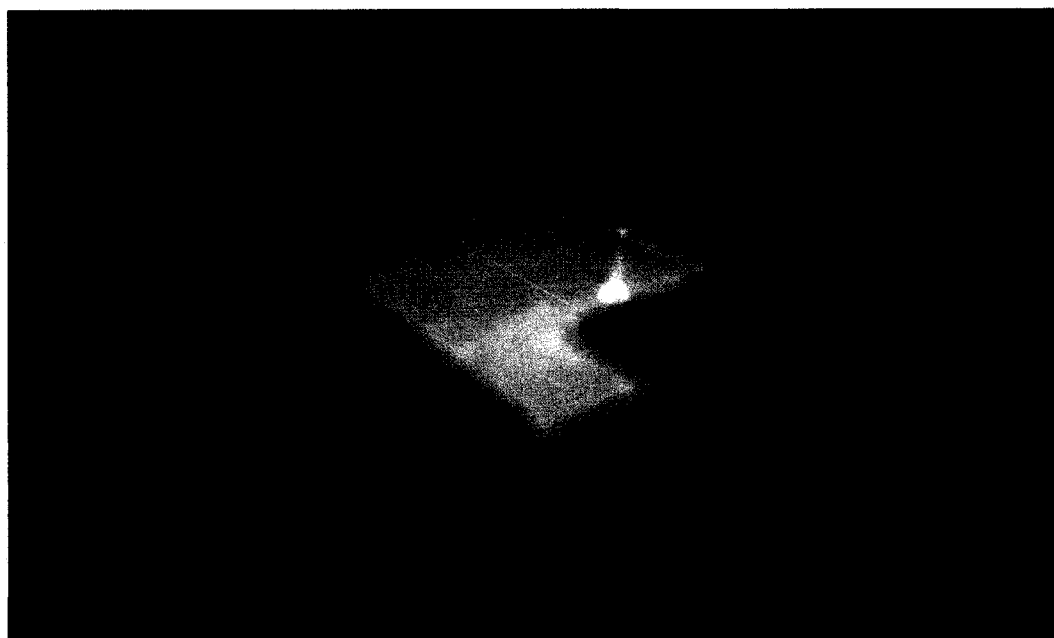


Figure 6.9 Effet de fumée simulation Cuisine textures 8 bits alpha 64x128x64 avec une résolution de 640x480 vue externe sans scène $s_f = 2.0$

L'image de la figure 6.9 est un rendu externe de la simulation Cuisine avec l'effet de fumée. La fumée s'est propagée aux deux pièces de la cuisine et est concentrée vers le plafond. Le foyer d'incendie (cuisinière) est clairement visible sur l'image. Les images de la figure 6.10 présentent la scène 3D avec l'effet de fumée sous différents échantillonnages.



(a)

Figure 6.10 Effet de fumée simulation Cuisine textures 8 bits alpha 64x128x64 avec une résolution de 640x480 (a) Vue interne $s_f = 2.0$ (b) Vue interne $s_f = 1.0$ (c) Vue interne $s_f = 0.25$



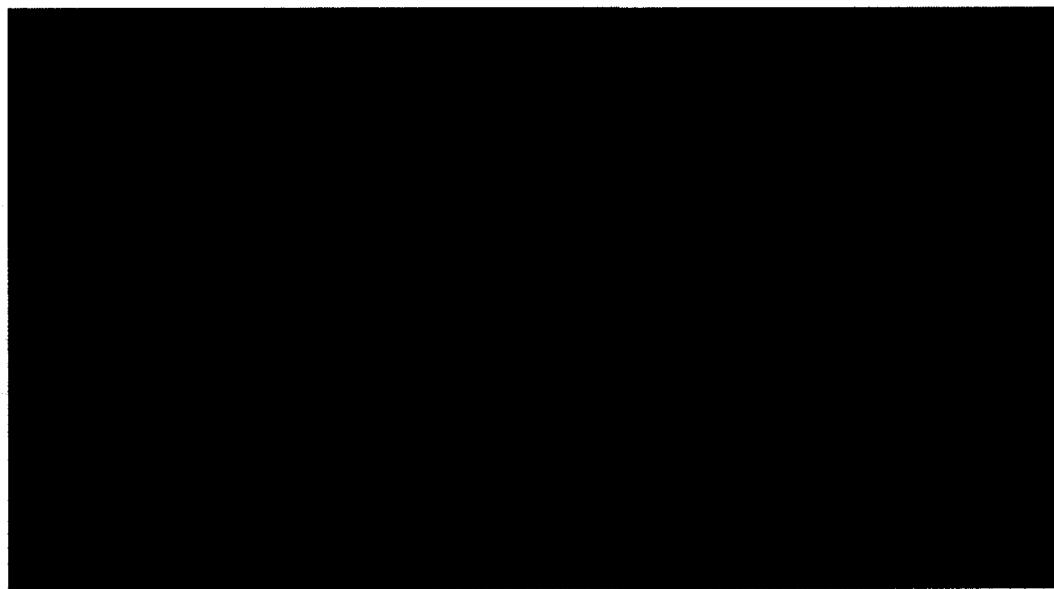
Figure 6.10b



Figure 6.10c

6.5 Effets de distorsion de chaleur

La variance par pixel est un résultat intermédiaire important car celle-ci sert de masque pour la texture de bruit lors de la composition de l'effet de chaleur. Elle permet aussi de voir rapidement la zone de l'image qui sera brouillée. La figure 6.11 illustre la variance pour un point de vue donné pour les deux scénarios simulés dans le canal rouge de l'image. Dans les deux cas la variance est particulièrement forte autour des foyers d'incendie soit sous le pupitre et au-dessus de la cuisinière (figure 6.11a et 6.11b). La figure 6.11c illustre le foyer d'incendie de la simulation C107 d'un point de vue plus éloigné que la figure 6.11a. Cette figure montre une grande diminution des valeurs de variance dans la région du foyer d'incendie (région bleue) puisque la grande majorité des plans de coupe échantillonnent des zones froides du volume de données.



(a)

Figure 6.11 Variance par pixels résolution 640x480 sur canal rouge (a) C107 proche (b) Cuisine (c) C107 éloigné



Figure 6.11b



Figure 6.11c



Figure 6.12 Texture de bruit gaussien 128 x 128 8 bits luminance

Les séries d'images des figures 6.13 et 6.14 montrent les résultats obtenus pour l'effet de chaleur avec et sans composition de l'effet de fumée. La texture de bruit gaussien de la figure 6.12 a été utilisée pour l'effet de chaleur. Les figures 6.13b et 6.14b illustrent la distorsion appliquée aux scènes 3D des deux simulations tests. Finalement, les figures 6.13c et 6.14c montrent l'effet final obtenu suite à la composition avec l'effet de fumée.



(a)



(b)

Figure 6.13 Effet de chaleur simulation C107 résolution 640x480 textures alpha 8 bits $s_f = 1.0$ (a) Scène sans effet (b) avec effet de chaleur (c) avec effet de chaleur et fumée

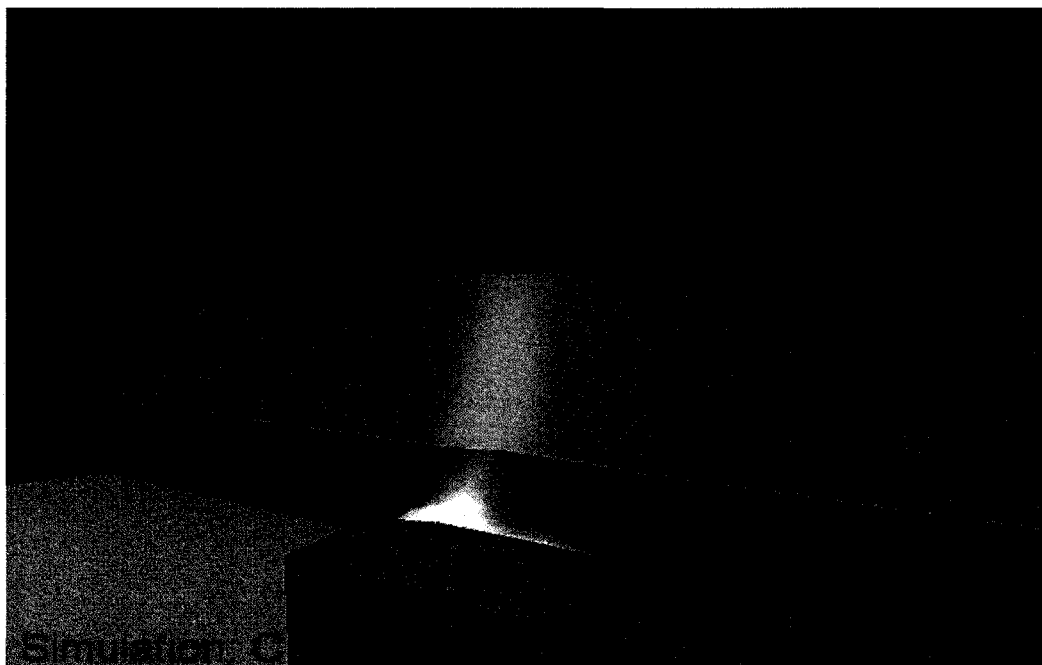
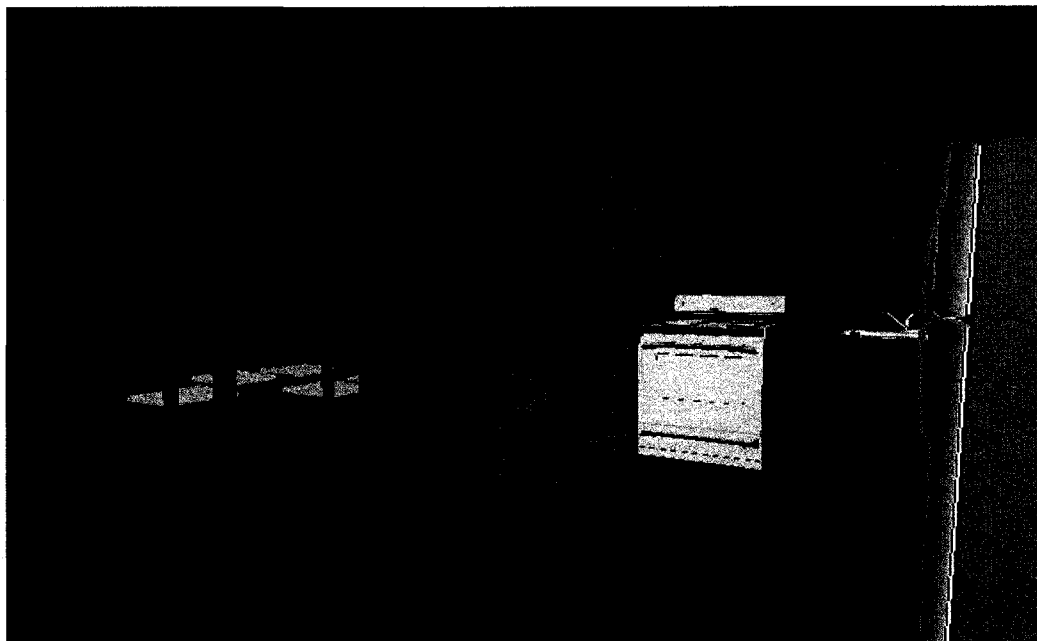
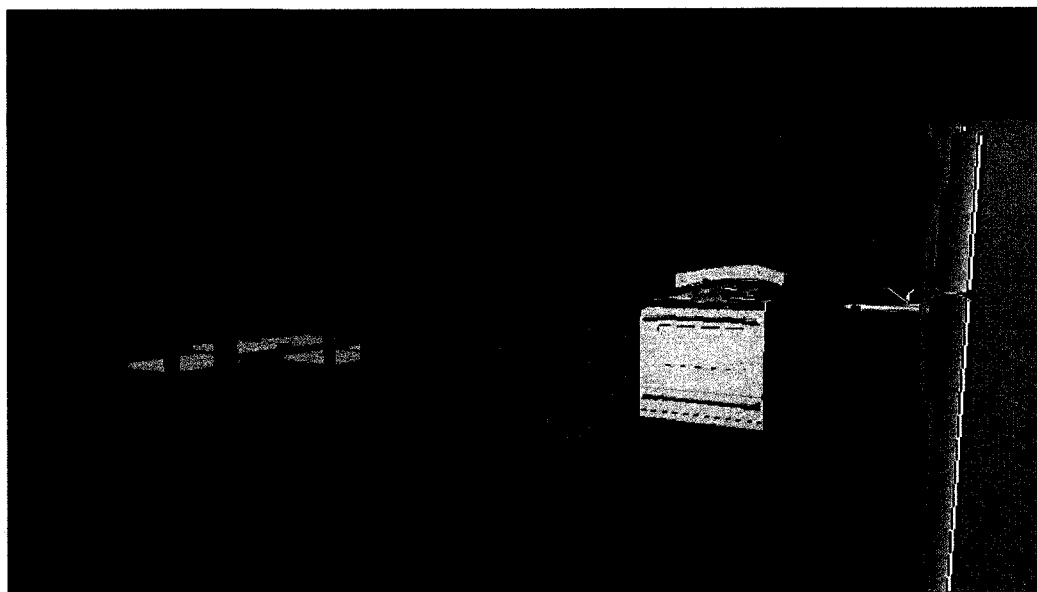


Figure 6.13c



(a)



(b)

Figure 6.14 Effet de chaleur simulation Cuisine résolution 640x480 textures alpha 8 bits
 $s_f = 1.0$ (a) Scène sans effet (b) avec effet de chaleur (c) avec effet de chaleur et fumée



Figure 6.14c

CHAPITRE 7 - DISCUSSION

7.1 Résultats

Les résultats visuels obtenus et présentés au chapitre 6 sont satisfaisants. La fumée est présente dans les régions attendues soit proche du foyer d'incendie et graduellement dans tout le volume. L'échelle de couleurs utilisée permet de déceler aisément les zones plus denses où la visibilité sera nulle.

Le rendu de la chaleur est réussie. La technique basée sur la variance cible bien les régions où l'indice de réfraction varie beaucoup soit proche des foyers d'incendie. De plus, la variance est une mesure qui s'adapte bien car elle dépend aussi de la taille de la population échantillonnée. Ainsi, la quantité et la taille de la distorsion varient en fonction de la distance du point de visualisation à la source de chaleur observée comme dans la réalité.

Les effets proposés dans cette recherche ne sont toutefois pas parfaits. La fumée se propage très rapidement dans le volume et fait ainsi ressortir un artefact résultant de l'utilisation des plans de coupe. La figure 7.1 illustre cet artefact. Comme le volume de données est de dimension fini il y a présence de coupures aux extrémités et lors d'intersections avec les surfaces de la scène 3D. L'artefact peut être diminué en augmentant la fréquence d'échantillonnage mais celle-ci a un effet direct sur la performance et l'aspect visuel (section 7.2). Cet artefact existe aussi pour l'effet de chaleur mais dans une moindre mesure étant donné que celui-ci est beaucoup plus localisée et peu présent aux extrémités des volumes.



(a)



(b)

Figure 7.1 Artefact de la technique par plans de coupe (a) vue externe (b) intersection avec des surfaces

L'effet de chaleur est particulièrement sensible aux calculs de la variance. Le problème de l'algorithme utilisé pour le calcul de la variance par pixel est que celui-ci implique le calcul de sommes de valeurs au carré. Ceci peut rapidement mener vers une instabilité numérique et des dépassements numériques dans les variables. Le problème de dépassement est un problème considérable en nuanceur. Par exemple, un tampon arrière standard de 8 bits par canal n'a que 8 bits pour représenter la somme des x_i et des x_i^2 avec la technique présentée au chapitre 5. Le nombre de plans de coupe utilisés a une influence directe sur le nombre d'échantillons et doit aussi être considéré. De plus, plus les valeurs des échantillons sont grandes plus on atteint la capacité de la variable rapidement. Les simulations présentées ont utilisé un tampon de 8 bits par canal mais le nombre de plans de coupe a été contrôlé pour éviter les dépassements. Ce contrôle a toutefois une influence sur la qualité de la visualisation de l'effet de chaleur car une trop grande diminution de l'échantillonnage augmente la probabilité de manquer des détails intéressants. Une bonne visualisation de la chaleur implique donc le choix d'une fréquence d'échantillonnage et d'une texture de bruit adéquates.

7.2 Paramétrisation des effets

Plusieurs paramètres de rendu ont été mentionnés dans les chapitres précédents. Ces paramètres ont une influence sur la qualité et le temps de rendu de la simulation.

La précision d'enregistrement des volumes de données est l'un de ces paramètres. Encodées sur 32 bits, ces données sont transformées et enregistrées dans les textures volumétriques et définissent leurs formats. Plus la précision des données est grande, plus la taille des textures volumétriques augmente et plus la visualisation sera précise. Une précision plus grande requiert l'utilisation de plusieurs canaux de textures voir plusieurs textures volumétriques ce qui peut rendre plus compliqué le travail de reconstitution des valeurs en nuances. De plus, le matériel graphique cible ne supporte peut-être pas le ou les formats de texture requis. L'expérience a toutefois montré (chapitre 6) qu'une représentation sur 8 bits est suffisante dans la plupart des situations.

La taille des textures volumétriques a une influence directe sur l'espace occupé en mémoire morte, vive et vidéo. La mémoire morte n'est généralement pas un problème mais la mémoire vive et la mémoire vidéo sont fortement limitées en taille par rapport à celle-ci. De plus, les performances du rendu peuvent être affectées. Pour que le matériel graphique puisse utiliser les textures dans les nuances, il faut d'abord y transférer l'information en provenance de la mémoire vive. Or, ce transfert d'informations requiert un temps non négligeable. Il est possible de contourner le problème en transférant en tout ou en partie les textures volumétriques au moment opportun, soit avant la visualisation ou pendant la simulation entre le traitement de chaque image. Ceci permet d'éviter un trafic mémoire important. Le tout est un équilibre entre la qualité, la taille et les performances recherchées.

La fonction de transfert est un autre paramètre affectant le rendu. La fonction de la figure 4.10 est un exemple typique de fonction de transfert unidimensionnelle que l'on peut utiliser pour représenter les valeurs de densité. Il est possible de la modifier pour étudier les valeurs de densité par échelon où l'on différencie ces derniers avec des couleurs différentes. Le logiciel de simulation FDS ne permet pas de déterminer la couleur de la fumée résultant de la combustion. Celle-ci est dépendante des espèces chimiques impliquées dans la combustion qui eux sont connus lors de l'élaboration du scénario d'incendie. La fonction de transfert peut être adaptée au besoin pour obtenir une couleur de fumée différente pour tenir compte des espèces chimiques et ce même en fonction des valeurs de densité.

La fréquence d'échantillonnage s_f est un paramètre ayant une influence directe sur la qualité du rendu et indirectement sur les performances. Lorsque la distance s entre les plans de coupe vaut s_0 , on s'assure que chaque voxel du volume de données sera échantillonné par au moins un plan de coupe. La fréquence est donc un paramètre de contrôle direct sur la qualité du rendu. Un sous échantillonnage, obtenu en augmentant la distance s entre chaque plan de coupe, approxime de moins en moins la simulation jusqu'à devenir une représentation sommaire de celle-ci. Indirectement, les performances du rendu s'améliorent car le nombre de fragments à traiter diminue considérablement. Un sur-échantillonnage a l'effet inverse : le nombre de fragments à traiter augmente mais la qualité de la visualisation du volume de fumée s'améliore et ce particulièrement aux frontières du volume où l'artefact illustré à la figure 7.1 diminue. La fréquence d'échantillonnage est particulièrement importante pour l'effet de chaleur car celle-ci influe sur plusieurs paramètres simultanément : la qualité du rendu, la performance et les dépassements de capacité liés au calcul de la variance par pixel.

L'effet de chaleur étant de nature qualitative il importe que la texture de bruits utilisée donne un résultat agréable. Comme l'effet de chaleur travaille dans l'espace écran, il est très facile de créer des images avec une trop grande distorsion avec une

mauvaise texture de bruit. L'effet inverse est aussi possible : une texture de bruit avec des valeurs trop faible fera peu ou pas de modifications à l'image. La détermination de la texture de bruit adéquate se fait par essais et erreurs et varie selon les simulations.

Finalement, la résolution a un impact important sur la vitesse d'exécution. En effet, plus la résolution est grande plus il y a de pixels à traiter. Ceci est particulièrement notable avec l'effet de distorsion : chaque pixel supplémentaire demande le calcul d'une variance supplémentaire. La résolution doit donc être adaptée en fonction des capacités graphiques disponibles.

7.3 Améliorations et perspective

La fumée a été représentée en utilisant une fonction de transfert à une dimension. Celle-ci permet d'attribuer des propriétés optiques comme la couleur et l'opacité aux données contenues dans les volumes. Essentiellement, cette fonction de transfert permet aux données de devenir visibles. Toutefois, il arrive qu'un volume de données contienne de l'information qui est difficile à visualiser avec une fonction de transfert à une dimension. Levoy (1988) et Kniss et al. (2002) proposent dans leurs travaux de recherche respectifs l'utilisation de fonctions de transfert à plusieurs dimensions. Levoy (1998) utilise des fonctions de transfert où la seconde dimension est le gradient au point (x,y,z) (taux local de changement) dans le volume de données. Kniss et al. (2002) montrent des exemples d'application de fonction de transfert à deux et trois dimensions sur des volumes de données scalaires et à plusieurs variables. Dans le domaine médical, il arrive fréquemment d'observer deux couches de tissus qui possèdent la même densité. Il devient alors impossible de les distinguer en n'utilisant qu'une fonction de transfert à une dimension. Dans le cas d'incendies, la fumée est composée de plusieurs produits de réactions. Il serait donc intéressant d'exploiter les fonctions de transfert multidimensionnelles pour observer ces différents composés. Il est aussi possible de les utiliser à d'autres fins. Par exemple, faire ressortir le côté tridimensionnel de la fumée en utilisant le taux local de changement dans la composition de la couleur. Une autre idée serait d'utiliser les autres variables d'une simulation FDS comme la fraction de mélange Z pour ajouter une teinte orangée à la fumée lors de la présence de feu et ainsi simuler le milieu participant.

La performance et la qualité de la représentation reposent principalement sur la technique de plans de coupe utilisée. Un problème difficile à résoudre est la spécification de la fréquence d'échantillonnage adéquate pour conserver un rapport performance / qualité visuelle acceptable. Plusieurs recherches ont abouti sur diverses améliorations possibles dont Engel et al. (2001). Ces travaux portent sur la réduction significative du nombre de plans de coupe requis pour échantillonner un volume de données scalaires tout en maintenant une bonne qualité de rendu. La méthode est basée sur une technique mathématique avec pré-intégration et s'exécute sur GPU. Une amélioration apportée aux plans de coupe permettrait d'avoir des simulations de plus grandes tailles qui sont visualisées à un taux d'images par seconde acceptable pour la visualisation interactive.

Finalement, la suite logique de ce projet de recherche est l'intégration des flammes. Les pistes de recherche sont très nombreuses. Une avenue possible est l'utilisation de fonctions de transfert multidimensionnelles en utilisant la technique de plans de coupe actuelle. Des variables comme le HRR (taux de dégagement de chaleur) et / ou la fraction de mélange Z peuvent servir de base lors de l'élaboration de ces fonctions. D'autres solutions sont possibles : concevoir une solution structurelle comme les splines de Lamorlette et Foster (2002), utiliser des particules comme Nguyen (2004), etc. Il n'y a, après tout, jamais de fumée sans feu!

CONCLUSION

Les méthodes hybrides de visualisation présentées dans cette recherche ont permis la visualisation de résultats numériques d'une simulation d'incendie basé sur le simulateur FDS. Des effets de fumée et de chaleur sont combinés à la scène d'incendie pour produire une simulation riche et interactive tout en maintenant un bon compromis sur la qualité de l'information physique. La possibilité de se déplacer à la première personne et d'avoir des résultats animés contribuent à la qualité des résultats observés ainsi qu'au sentiment d'immersion.

Les capacités des unités de traitement graphique (GPU) ont permis de réaliser le traitement et la visualisation des simulations d'incendie. L'effet de chaleur a particulièrement bénéficié de ces capacités lors du calcul de la variance par pixel. Malgré la présence d'artefacts graphiques indésirables dus à l'utilisation d'une technique de visualisation volumétrique à base de plans de coupe, le GPU aura permis la visualisation interactive d'une simulation sur du matériel relativement âgé.

Un pipeline de visualisation de simulation a été élaboré et permet la transformation des données volumétriques jusqu'à la réalisation des effets. Ce pipeline inclus un moteur de rendu, un plugiciel pour l'export de la scène ainsi que des effets volumétriques.

Les objectifs de ce projet de recherche ont été atteints. Plusieurs pistes pour faire évoluer ces travaux de recherche ont été mentionnées. La suite logique de ce projet est sans aucun doute l'élaboration d'une technique de visualisation du feu. C'est uniquement avec cette dernière technique que l'on pourra véritablement aider les gens impliqués dans le combat contre les incendies.

BIBLIOGRAPHIE

ABANTO J., BARRERO D., REGGIO M. et OZELL B. (2004) Airflow modelling in a computer room. *Building and Environment*, 39(12), 1393-1402.

ABANTO J., REGGIO M., BARRERO D. et PETRO E. (2006) Prediction of fire and smoke propagation in an underwater tunnel. *Tunneling and Underground Space Technology*, 22(1), 90-95

BARRERO D., OZELL B. et REGGIO, M. (2003) On CFD and Graphic Animation for Fire Simulation. *CFD2003*. Vancouver

BERGER M., TROUT T. et LEVIT N. (1990). Ray tracing mirages. *IEEE Computer Graphics and Applications*, 10(3), 36-41

Building and Fire Research Laboratory du National Institute of Technologies and Standards. (2006). FDS User's Guide. États-Unis. Consulté le 4 mars 2007, tiré de <http://www.fire.nist.gov/fds/docs>

Building and Fire Research Laboratory du National Institute of Technologies and Standards. (2006). FDS Technical Reference Guide. États-unis. Consulté le 4 mars 2007, tiré de <http://www.fire.nist.gov/fds/docs>

CABRAL B., CAM N. et FORAN J. (1994). Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proceedings ACM Symposium on Volume Visualization 94*. 91-98. New York : IEEE

COOK R.L., PORTER T. et CARPENTER L. (1984). Distributed ray tracing. *Computer Graphics SIGGRAPH 84 Conference Proceedings*, 18(3), 147-154

CHANDRASEKHAR S. (1960). Radiative transfer. Dover, New York

CHIBA N., MURAOKA K., TAKAHASHI H. et MIURA M. (1994) Two-dimensional visual simulation of flames, smoke and the spread of fire. *The Journal of Visualization and Computer Animation*. 5, 37-53

ENGEL K., KRAUS M. et ERTL T. (2001) Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. In *Siggraph/Eurographics Workshop on Graphics hardware 2001*.

FATTAL R. et LISCHINSKI D. (2004) Target-Driven smoke animation. *ACM Transactions on Graphics*, 23(3), 441-448, États-Unis : ACM

FEDKIW R., STAM J. et JENSEN H., Visual Simulation of Smoke. In *Computer Graphics (SIGGRAPH 2001)*, 15-22, New York : ACM

FOSTER N. et METAXAS D. (1996) Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5), 471-483

IKITS M., KNISS J., LEFOHN A. et HANSEN C. (2004) Volume rendering techniques. In R. Fernando, *GPU GEMS*, 667-692. Boston : Addison-Wesley

KNISS J., KINDLMANN G. et HANSEN C. (2002) Multi-Dimensional Transfer Functions for Interactive Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3), pp. 270-285

KRÜGER J. et WESTERMANN R. (2005) GPU simulation and rendering of volumetric effects for computer games and virtual environments. *European Association for Computer Graphics 26th Annual Conference*. 24(3). 685-693

LEE H., KIM L., MEYER M. et DESBRUN M. (2001) Meshes on fire. *EG Workshop on Computer Animation and Simulation*. 75-84

LIDE D. (2003) Handbook of Chemistry and Physics. CRC Press LLC, 84^{ième} édition.

JENSEN H. W. et CHRISTENSEN P. H. (1998). Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH 98 Conference Proceedings, Annual Conference Series*, 311-320

KAJIYA J.T. et HERZEN von B. P. (1984). Ray tracing volume densities. *Computer Graphics SIGGRAPH 84 Conference Proceedings*. 18(3) 165-174

LAMORLETTE A. et FOSTER N. (2002). Structural Modeling of Flames for a Production Environment. *ACM Transactions on Graphics*. 21(3) 729-735 : San Antonio, USA : ACM

LEVOY, M. (1998). Display of Surfaces from Volume Data. *IEEE Computer Graphics & Applications*. 8(2), 29-37

MUSGRAVE, F. (1990). A note on ray tracing mirages. *IEEE Computer Graphics and Applications*. 10(6), 10-12

NGUYEN, H. (2004). Fire in the « Vulcan » Demo. In R. Fernando, *GPU GEMS*, 87-105. Boston : Addison-Wesley

PERRY C.H. et PICARD R.W. (1994) Synthesizing flames and their spreading. *Proceedings of the Fifth Eurographics Workshop on Animation and Simulation*. 1-14

PHARR M. et HUMPHREYS G. (2004) Volume Scattering. *Physically based rendering*. (12, pp. 569-595) Morgan Kaufmann : Chine

REEVES W.T. (1983). Particle systems – a technique for modelling a class of fuzzy objects. *Computer Graphics SIGGRAPH 83 Conference Proceedings*, 17(3), 358-376

RUSHMEIER H., HAMINS A. et MUN YOUNG C. Volume rendering of pool fire data, *IEEE Computer Graphics and Applications*. 15(4), 62-67.

SERON F., GUTIERREZ D., GUTIERREZ G. et CEREZO E. (2004). Visualizing sunsets through inhomogeneous atmospheres. *Proceeding of Computer Graphics International*, 349-356

STAM J. et FIUME E. (1993). Turbulent Wind Fields for Gaseous Phenomena. In *Proceedings of ACM SIGGRAPH 1993 Computer Graphics Proceedings, Annual Conference Series*. 369-373. New York : ACM.

STAM J. et FIUME E. (1995). Depiction of fire and other gaseous phenomena using diffusion processes. *Proceedings of SIGGRAPH*. 129-136

STAM J. (1999) Stable Fluids. In *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*. 121-128. New York : ACM.

TREUILLE A., MCNAMARA A., POPOVIC Z. et STAM J. (2003). Keyframe control of smoke simulations. *ACM Transactions on Graphics (Proceedings of ACM Siggraph 2003)*. 21(3), 716-723

WEI X. , WEI L. , MUELLER K. et KAUFMAN A. (2002). Simulating Fire with Texture Splats. *IEEE Visualization 2002. Proceedings*. 227-234. Piscataway, NJ : IEEE

ZHAO Y., WEI X., FAN Z., KAUFMAN A. et QIN H. (2003). Voxels on Fire. *IEEE Visualization 2003*. 271-278. Piscataway, NJ : IEEE

WESTERMANN R. (1998). Efficiently using graphics hardware in volume rendering applications. In *Computer Graphics. Proceedings SIGGRAPH 98 Conference Proceedings*. 169-177. New York : ACM.

WHITTED T. (1980) An improved illumination model for shaded display. *Comm. ACM*, 26(6), 342-9

ZHAO Y., HAN Y., FAN Z., QIU F., KUO Y., KAUFMANN A. et MUELLER K. (2007). Visual Simulation of heat shimmering and mirage. *IEEE Transactions on Visualization and Computer Graphics*, 13(1), 179-189

ANNEXES

A.1 Intersection Droite-Plan

Soit la droite D et deux points P_1, P_2 sur celle-ci.

L'équation de la droite D peut être exprimée de manière paramétrique par :

$$P = P_1 + u(P_2 - P_1) \quad \text{Éq. (1)}$$

Soit P_3 le point d'intersection et P un point sur le plan de normale N. L'équation d'un plan peut donc s'écrire :

$$N \bullet (P - P_3) = 0 \quad \text{Éq. (2)}$$

En substituant P de l'équation (1) dans l'équation (2), on obtient :

$$N \bullet (P_1 + u(P_2 - P_1)) = N \bullet P_3 \quad \text{Éq. (3)}$$

Il ne reste plus qu'à résoudre pour u :

$$u = \frac{N \bullet (P_3 - P_1)}{N \bullet (P_2 - P_1)} \quad \text{Éq. (4)}$$

Pour tester si la droite D intersecte le plan que sur la plage P_1, P_2 , il suffit de vérifier que u se retrouve entre 0 et 1.

Lorsque le dénominateur est nul, la droite est soit parallèle au plan (aucune solution) ou bien sur le plan (infinité de solution).

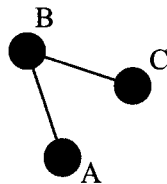
A.2 Ordonnancement d'un groupe de sommets autour d'un sommet d'ancrage dans le sens anti-horaire ou horaire

L'orientation de trois sommets dans un plan peut être déterminé grâce à la valeur du signe du déterminant ci-dessous, qui correspond au double de l'aire du triangle en valeur absolue composé des sommets a,b,c :

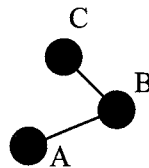
$$\begin{vmatrix} x_a & y_a & 1 \\ x_b & y_b & 1 \\ x_c & y_c & 1 \end{vmatrix}$$

Ceci revient donc à calculer :

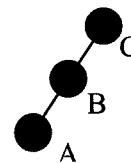
$$(x_a \times (y_b - y_c)) - (y_a \times (x_b - x_c)) + (x_b \times y_c - y_b \times x_c)$$



Sens horaire



Sens anti-horaire



Sommets colinéaires

Il est fort utile de représenter ce test en terme d'implication :

$$b > c \Leftrightarrow \text{Orientation}(a,b,c) = \text{Anti-horaire}$$

$$b < c \Leftrightarrow \text{Orientation}(a,b,c) = \text{horaire}$$

$$b = c \Leftrightarrow \text{Orientation}(a,b,c) = \text{colinéaire}$$

On peut maintenant caractériser les sommets l'un par rapport à l'autre. Ceci nous permet d'utiliser avec aisance un algorithme de tri plus conventionnel comme le tri à bulle, qui requiert un test de comparaison entre les éléments à trier. En appliquant un tri à bulle à une série de sommets, on peut obtenir les sommets triés dans le sens désiré à partir d'un point d'ancrage donné a .