

Université de Montréal

Simulation de flammes interactives en temps réel

par

Sammy Fatnassi

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la faculté des études supérieures et postdoctorales

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

Mai 2010

© Sammy Fatnassi, 2010

Université de Montréal
Faculté des études supérieures et postdoctorales

Ce mémoire de maîtrise intitulé

Simulation de flammes interactives en temps réel

présenté par
Sammy Fatnassi

a été évalué par un jury composé des personnes suivantes :

Neil Stewart
président-rapporteur

Pierre Poulin
directeur de recherche

François Major
membre du jury

Sommaire

La synthèse d'une flamme animée dans un environnement 3D virtuel, reste à ce jour une tâche ardue, exigeant de judicieusement balancer réalisme et coût de calcul. Dans ce mémoire, nous présentons un ensemble de techniques pour sa simulation en temps réel tout en modélisant une interaction à des forces externes. Nous désirons minimiser son coût de calcul tout en préservant une apparence convaincante dans l'optique d'une intégration au sein de systèmes existants, n'affectant pas indûment leurs performances. Un champ de vélocité est extrait d'une simulation de ressorts et mis à profit dans le déplacement de chaînes de particules modélisant la forme de la flamme par l'entremise de la paramétrisation d'une surface NURBS. Considérant l'importance qu'ils ont sur notre perception de la combustion, nous prenons également soin de reproduire l'illumination, les ombres, et l'effet d'éblouissement qu'elle engendre.

Mots clefs :

Flamme, feu, simulation, bougie, temps réel, vent, éclairage, ombres

Abstract

The synthesis of an open flame in a virtual 3D environment, remains to this day an arduous task, requiring a wise balance between realism and processing cost. In this M. Sc. thesis, we present a set of techniques for its simulation in real time while also modeling the interaction with external forces. Our goal is to minimize the cost while preserving a convincing appearance, thus facilitating integration of the techniques into existing systems without unduly affecting their performance. A velocity field is extracted from a spring-mass simulation which contributes to moving chains of particules that are used in modeling the flame shape through the configuration of a NURBS surface. In light of the importance they have on our perception of combustion, we also take care to duplicate the lighting, shadows and bloom the flame gives rise to.

Keywords :

Flame, fire, simulation, candle, real time, wind, lighting, shadows

Table des matières

1	Introduction	1
1.1	Présentation	1
1.2	Objectifs	2
1.3	Définitions	3
2	Anatomie d'une flamme	5
2.1	Oxydation	5
2.2	Chaleur	7
2.3	Illumination	7
2.4	Forme	8
3	Travaux antérieurs	9
3.1	Modèle physique	9
3.2	Modèle empirique	13
3.3	Modèle capture	19
4	Implémentation	21
4.1	Modélisation	22
4.1.1	Système de particules	22
4.1.2	Surface NURBS	23
4.1.3	Flammes complexes	27
4.2	Simulation	27
4.2.1	Grille de noeuds	28
4.2.2	Système de ressorts	29
4.2.3	Forces externes	30
4.2.4	Champ de vitesse	30

4.3	Apparence	31
4.3.1	Placage de texture	31
4.3.2	Effet d'éblouissement	33
4.3.3	Transparence	34
4.4	Impact sur l'environnement	35
4.4.1	Illumination	36
4.4.2	Ombres	39
5	Résultats	45
5.1	Paramétrisation NURBS	47
5.2	Simulation de ressorts	49
5.3	Nombre de flammes	49
5.4	Performances avec les effets	50
6	Conclusion	57
6.1	Comparaison	58
6.2	Perspectives	59
	Bibliographie	61

Table des figures

2.1	Représentation du triangle de feu. Source : Wikipedia par Gustavb (original) et Vandal (traduction).	6
2.2	Zones d'une flamme laminaire. Source : [Boy06].	6
2.3	Flamme avec apport externe de comburant (oxygène) versus flamme de prémélange. Source : [Boy06].	8
3.1	Résultats de Stam et Fiume issus d'amalgames représentant des gaz soumis aux forces d'advection et du vent. Source : [SF95].	10
3.2	Résultats de Nguyen et al. issues d'une simulation de fluide Navier-Stokes du combustible et des gaz secondaires. Source : [NFJ02].	11
3.3	Résultats de Hong et al. reprenant les travaux de Nguyen et al. tout en leur ajoutant un motif cellulaire, restituant mieux le front des flammes turbulentes. Source : [HSF07].	12
3.4	Résultats de Melek et Keyser intégrant la méthode de fluide stable développée par Stam, aux travaux de Nguyen et al. Source : [MK02].	12
3.5	Résultats d'un système de particules conçu par Reeves et intégré à la production cinématographique ' <i>Star Trek II : The Wrath of Khan</i> '. Source : [Ree83].	13
3.6	Résultats de Wei et al. combinant une simulation de fluide à une technique de <i>texture splats</i> pour les détails de hautes fréquences. Source : [WLMK02].	14
3.7	Représentation 2D d'une chaîne de particules et son utilisation dans la modélisation d'une flamme. Chaque particule S_i possède une position relative aux axes X/Y et est chronologiquement ordonnée selon son indice i . Source : [BPP01]. . .	15
3.8	Résultats de Beaudoin et al. issus d'isosurfaces modélisées à l'aide d'une chaîne de particules. Source : [BPP01].	15

3.9	Résultats de Lamorlette et Foster reprenant le concept de la chaîne de particules de Beaudoin et al. augmenté d'une méthode stochastique pour les détails de hautes fréquences et utilisant un rendu volumétrique, menant à la création de séquences pour la production cinématographique ' <i>Shrek</i> '. Source : [LF02].	16
3.10	Résultats de Horvath et Geiger issus d'un système de particules pour la modélisation du flot des flammes et de plusieurs simulations de fluide 2D orientées écran pour les détails de hautes fréquences. Source : [HG09].	17
3.11	Technique de rendu volumétrique présentée par Fuller et al. Source : [FKM ⁺ 07].	18
3.12	Résultats de Fuller et al. issus d'un rendu volumétrique utilisant la déformation d'un volume englobant pour la forme générale, et d'un <i>bruit de Perlin amélioré</i> pour les détails de hautes fréquences. Source : [FKM ⁺ 07].	18
3.13	Résultats de Balci et Foroosh utilisant une simulation de tissu simulant l'apparence d'une flamme de soie. Source : [BF06].	19
3.14	Résultats de Bridault issus de chaînes de particules paramétrant une surface NURBS texturée. Source : [Bri07].	19
3.15	Résultats de Stich et Magnor produits à l'aide d'une analyse statistique de l'apparence d'une flamme. Source : [SM05].	20
4.1	Mèche et émetteurs de particules de l'implémentation de Bridault, pour la modélisation d'une flamme. Source : [Bri07].	23
4.2	Modélisation de notre flamme laminaire où quatre émetteurs (triangles) génèrent des chaînes de particules (cercles blancs) spécifiant la surface extérieure du volume de la flamme. La pointe est générée par la position de la plus ancienne particule émise par la chaîne centrale (cercle <i>e</i>) lors de la paramétrisation de la surface NURBS.	24
4.3	Exemple d'une surface NURBS (vert) avec ses points de contrôle (rouge). Source : Wikipedia par l'utilisateur Chrschn.	26
4.4	Deux configurations de mèches suggérées par Bridault, pour la modélisation de flammes complexes. Source : [Bri07].	27
4.5	Vue latérale d'une configuration de noeuds possible (sans les ressorts). Les flèches représentent les vecteurs de vitesse extraits du système et l'ensemble de leurs directions permettent aux particules de se déplacer en suivant la forme générale d'une flamme de bougie.	28

4.6	Configuration des 14 ressorts liant chaque noeud à ses voisins. Figure (a) présente les liens directs tandis que la figure (b) présente les liens diagonaux.	29
4.7	Vue latérale de la propagation par ressorts, d'une force reçue sur un noeud à ses voisins immédiats.	29
4.8	Étapes de la création d'une texture de flamme, que l'on peut appliquer sur notre surface NURBS de type cylindrique.	32
4.9	Extraction de la coordonnée U à partir de l'angle horizontal d'un émetteur à l'axe des X . Les quatre émetteurs présents sur cette image, se trouvent sur le plan formé par les axes X et Y	33
4.10	Flamme (a) avec translation de texture et (b) sans.	34
4.11	Flamme (a) avec effet d'éblouissement et (b) sans.	34
4.12	Chevauchement de plans semi-transparents (a) avec et (b) sans gestion de la transparence.	36
4.13	Impacts visuels des ombres et de l'illumination sur le réalisme d'un environnement de synthèse.	38
4.14	Illumination anisotropique d'une scène.	39
4.15	Angle entre l'inclinaison de la flamme et la différence de position du fragment.	40
4.16	Scène (a) avec et (b) sans volume photométrique appliqué lors de l'illumination.	42
4.17	Principe de la technique de <i>shadowmap</i> . Le point rouge est dans l'ombre puisque la texture <i>shadowmap</i> générée à partir du point de vue de la lumière contient un point plus près (blanc). A l'opposé, le point vert reçoit des rayons de la source lumineuse puisqu'aucun élément obstruant le point n'est détecté, la texture contenant les informations du même point vert.	43
4.18	Ombres créées à partir (a) d'un <i>shadowmap</i> standard et (b) d'un <i>variance shadowmap</i> . La seconde méthode présente des contours flous restituant une apparence plus naturelle que la méthode de base exhibant des artefacts de discrétisation ainsi que des démarcations découpées.	43
4.19	Vue verticale du rendu de quatre des six faces d'un <i>cubemap</i> . La source lumineuse omnidirectionnelle est représentée par le cercle jaune, tandis que les rectangles verts représentent les faces du cube recevant chacun des rendus alignés sur les axes X^+ , X^- , Z^+ , Z^-	44

5.1	Scène de base de 4653 triangles, utilisée pour les tests avec une flamme de 175 particules. Performances : 137 images/s (avec éblouissement, flamme et ombres), 143 images/s (avec éblouissement et ombres) et 294 images/s (avec éblouissement et flamme).	46
5.2	Scène Sponza de 193990 triangles avec une flamme de 175 particules. Performances : 55 images/s (avec éblouissement, flamme et ombres), 55 images/s (avec éblouissement et ombres) et 108 images/s (avec éblouissement et flamme).	47
5.3	Scène Sibenik de 238909 triangles avec une flamme de 175 particules. Performances : 45 images/s (avec éblouissement, flamme et ombres), 45 images/s (avec éblouissement et ombres) et 87 images/s (avec éblouissement et flamme).	48
5.4	Maillage selon différents algorithmes de discrétisation de surface NURBS.	49
5.5	Coût du rendu d'une flamme, par configuration de surface NURBS.	50
5.6	Coût CPU du rendu d'une flamme, par nombre de particules. Note : Les configurations PL40 et PE40 se chevauchent.	51
5.7	Coût GPU du rendu d'une flamme, par nombre de particules.	52
5.8	Performances du rendu, par nombre de particules. Note : Les configurations PL40 et PE40 se chevauchent.	52
5.9	Comparatif du rendu versus la mise-à-jour d'une flamme, par nombre de particules.	53
5.10	Coût de la mise-à-jour de la flamme, par nombre de noeuds de la simulation de ressorts. Valeurs pour une grille de vitesse de tailles 5, 6, 7, 8, 9, 10 et 15.	53
5.11	Comparatif du rendu versus la mise-à-jour, par nombre de flammes.	54
5.12	Coût des ombres et de l'effet d'éblouissement, par nombre de flammes.	54
5.13	Performances des ombres et de l'effet d'éblouissement, par nombre de flammes.	55

Liste des tableaux

4.1	Exemple de la paramétrisation d'une surface NURBS pour une configuration de quatre chaînes périphériques de huit particules (voir figure 4.2).	26
-----	--	----

Liste des algorithmes

1	Mise-à-jour du système de particules	25
2	Gestion de la transparence	37

Remerciements

En premier lieu, je désire chaleureusement remercier Heather, elle a su me supporter dans ce périple décidé sur un coup de tête et parsemé d'embûches. Sa compréhension des diverses difficultés inhérentes à la recherche académique et sa présence m'ont permis de ne jamais perdre de vue la lumière au bout du tunnel. Une mention également à Pierre Baillargeon dont les nombreuses suggestions musicales ont su m'accompagner tout au long de ce trajet.

Merci au CRSNG, mon retour aux études s'étant concrétisée grâce à leur soutien financier par l'entremise de la subvention de recherche de mon directeur de maîtrise. Merci également aux membres du jury dont les suggestions ont permis d'améliorer la lecture de ce mémoire.

Et finalement, merci à Pierre Poulin, mon directeur de recherche. Sa rencontre sut me convaincre de faire le grand saut vers une période de deux ans emplies de stress, de travail ardu et d'un budget frugal! Ses nombreuses suggestions et corrections ont grandement contribué à faire de ce mémoire un accomplissement dont je peux être fier.

Droits d'auteur

Les diverses images issues de sources externes sont utilisées suivant les règles de droits d'auteur de leur source respective, soit :

ACM ACM COPYRIGHT NOTICE. Copyright © 2010 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept., ACM, Inc., fax +1 (212) 869-0481, or permissions@acm.org

IEEE ©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Graphics Interface Copyright ©2001 par l'Association canadienne de l'informatique. Il est permis de citer de courts extraits et de reproduire des données ou tableaux du présent compte rendu, à condition d'en identifier clairement la source.

Wikipedia Ce fichier est publié sous la licence Creative Commons Attribution ShareAlike 3.0. En résumé : vous êtes libres de redistribuer et faire des corrections de cette œuvre sous réserve de mentionner clairement son auteur, et de la publier sous une licence compatible avec celle-ci.

Chapitre 1

Introduction

[...]
Il suffira d'une étincelle
D'un rien, d'un geste
Il suffira d'une étincelle,
Et d'un mot d'amour
Pour allumer le feu
Allumer le feu
Et faire danser les diables et les dieux
-JOHNY HALLYDAY (ALLUMER LE FEU)

1.1 Présentation

Malgré les millénaires nous séparant de la première domestication du feu, sa nature demeure évasive, rébarbative à l'analyse et la modélisation. La relation ambiguë que l'homme lui entretient depuis toujours n'a su être complètement dissipée par une meilleure compréhension scientifique. Telle une déité, il est à la fois vénéré pour ses nombreuses vertus et craint pour son énorme potentiel destructeur. Par nuits froides d'hiver, nous sommes redevables à sa chaleur autour d'un repas ainsi qu'à la lueur d'une chandelle perçant la noirceur de nos soirées empreintes de mystères. Au travers de la conception d'outils progressivement plus complexes, il a guidé les premiers pas l'humanité vers le sol lunaire et mené à une meilleure compréhension de notre environnement. Nous redoutons malgré tout les catastrophes qu'occasionne sa perte de contrôle, menant à la destruction de maisons, villes et forêts.

“[...] il est parfaitement légitime de considérer que, par les perspectives nouvelles qu'il ouvrit à l'humanité naissante, par la maîtrise de nouvelles forces, mais aussi par tous les problèmes jusque-là inconnus qu'il obligea à résoudre, le feu fut l'un des

facteurs les plus importants dans le processus même d’homínisation”[Per85]

Un environnement familier acquiert une toute nouvelle dimension lorsque confiné aux fluctuations envoûtantes d’une chandelle et la danse de ses ombrages. Sa nature chaotique donne naissance à une pléiade de phénomènes fascinants difficilement synthétisables, tels une forme en constante évolution, un impact sur la luminosité très localisée variant rapidement et l’émission de fumée en présence d’un excédant de carburant.

Jusqu’au début du vingtième siècle, la combustion de divers matériaux demeura la principale méthode à notre portée pour dissiper la noirceur à la venue du crépuscule. Toute récréation virtuelle d’univers prédatant cette période se doit ainsi d’être en mesure de simuler adéquatement ce phénomène si elle espère proposer une authenticité convaincante. Une bonne maîtrise des nombreux aspects d’une flamme, telles son apparence, l’illumination et les ombres résultantes, introduit un gain considérable de réalisme au rendu d’un environnement. Dans cette optique, nous proposons un système en mesure de reproduire les aspects visuels les plus importants d’une flamme en constante évolution, sa réaction à l’environnement ainsi que son impact sur l’illumination et les ombres d’une scène virtuelle.

1.2 Objectifs

Nous désirons reproduire la flamme laminaire d’une bougie en temps réel et trahissant minimalement sa nature synthétique. Ceci implique une apparence convaincante, une dynamique de déplacement réaliste, une réaction à des forces externes telles que le vent et aucune boucle d’animation répétitive. Par l’entremise d’éclairage et d’ombres, notre simulation doit être en mesure d’avoir un impact plausible sur son environnement. Un changement de perspective sur la flamme ne devrait pas diminuer son réalisme, comme il peut être le cas avec certaines techniques existantes, tel le *billboarding*¹. Le tout doit pouvoir être réalisé en un temps de calcul raisonnable afin de faciliter son intégration dans un environnement temps réel où elle doit ne représenter qu’une petite partie du tout.

Nous proposons d’accomplir ces objectifs en combinant plusieurs solutions précédemment avancées dans le domaine. Une surface NURBS (*Non-Uniform Rational Basis Splines*), est utilisée pour générer la forme de la flamme, lui conférant une apparence tridimensionnelle convaincante. Un système de particules paramétrise les points de contrôle de la surface de la flamme, permettant d’aisément modifier sa forme. Ces particules voient leurs positions affectées par un

1. Technique d’infographie simulant divers éléments, tel de la végétation, en insérant des plans 2D texturés dans la scène et orientés vers la caméra.

champ de vélocité extrait d'une simulation de ressorts réagissant aux forces externes telles que la convection des gaz, le déplacement et le vent, conférant au modèle une dynamique en constante évolution, temporellement cohérente, et à un coût de calcul raisonnable.

Ce mémoire se divise en six chapitres dans lesquels nous élaborons sur la nature d'une flamme au chapitre 2, puis survolons les différentes contributions au domaine de recherche au chapitre 3. Ceci nous mène à la présentation de notre implémentation au chapitre 4 et ses résultats au chapitre 5. Nous concluons sur une discussion des améliorations futures potentielles au chapitre 6.

1.3 Définitions

Plusieurs termes couramment utilisés dans ce mémoire conviennent d'être bien définis dès le début.

Flamme : Une réaction chimique dont l'émission de lumière et de chaleur laisse entrevoir une forme en constante évolution.

Flamme de diffusion : Flamme issue d'un carburant vaporisé, s'embrasant au contact d'un comburant et d'une chaleur suffisante. Sa caractéristique principale est la dissociation du carburant et comburant jusqu'à leur réaction une fois en contact, tels l'huile d'une lampe et l'oxygène ambiant.

Flamme laminaire : Contrairement à d'autres types de flammes, cette flamme possède une forme allongée bien définie et moins turbulente, sa forme provenant du mouvement laminaire des gaz en présence de gravité. Les flammes de diffusion en font partie.

Feu : Combustion d'une plus grande envergure, pouvant se concevoir comme un ensemble de flammes.

Trame : Durée d'exécution pour la synthèse d'une image d'une séquence.

CPU : *Central processing unit*. Processeur principal d'un ordinateur s'occupant de coordonner toutes les tâches du système.

GPU : *Graphic processor unit*. Processeur dédié aux graphiques 3D et grandement optimisé pour cette tâche. Le GPU réside habituellement sur une carte d'extension, indépendante du CPU qui lui envoie les commandes à exécuter.

Pixel : Plus petit élément de couleur d'une image.

Fragment : Pixel en cours d'évaluation par le système 3D et pouvant être retenu ou non dans son intégration à l'image finale.

Chapitre 2

Anatomie d'une flamme

[...]
Joli feu de bois
Feu qui chante
Dans le vent qui passe
Je te vois
Et je chante
Joli feu de bois
Et je chante,
je chante avec toi
-CHARLES TRENET (JOLI FEU DE BOIS)

Bien qu'il nous a accompagné au travers les âges depuis sa domestication il y a plus de 600,000 ans, la mécanique du feu n'a été explorée que plus récemment. Pendant longtemps, il fut considéré comme un des éléments de base régissant l'univers et ne nécessitant donc pas une étude approfondie, aux côtés de l'eau, l'air et la terre. Cette perception changea durant le Siècle des Lumières, période où l'Académie des sciences soumit le sujet "De la nature et de la propagation du feu" pour son concours de 1738. Bien que ce mémoire ne s'intéresse pas à la fidèle simulation des phénomènes physiques et chimiques sous-jacents à la combustion, une bonne compréhension mène tout de même vers une meilleure maîtrise des différents aspects visuels de la flamme, pouvant ainsi donner naissance à des pistes de solutions intéressantes.

2.1 Oxydation

En présence de conditions spécifiques, l'oxydation est une réaction chimique fortement exothermique donnant lieu à la lumière et la chaleur. Communément appelées le triangle de feu, ces circonstances sont constituées de trois éléments : un carburant, un comburant et de la chaleur



FIGURE 2.1 – Représentation du triangle de feu. Source : Wikipedia par Gustavb (original) et Vandal (traduction).



FIGURE 2.2 – Zones d'une flamme laminaire. Source : [Boy06].

(voir figure 2.1). Le carburant peut être issu d'un large éventail de matériaux tels que de l'huile ou du bois, et le rôle du comburant est usuellement exercé par l'oxygène. Cette réaction chimique s'explique par des molécules excitées par la chaleur, tentant de retrouver un état stable en se recombinant entre elles. Le phénomène s'auto-entretient jusqu'à épuisement d'un des trois éléments nécessaires.

Une flamme de diffusion est constituée de trois zones distinctes, soient le noyau bleuté où les gaz frais sont parfaitement brûlés, une zone riche en combustible au centre, et une zone pauvre sur le front de la flamme, mais riche en comburant (voir figure 2.2). Ce type de flamme exige qu'un carburant vaporisé entre en contact avec l'oxygène, et s'éloigner de sa source implique une présence proportionnellement accrue de gaz secondaires issus de l'oxydation. En présence d'un excès de carburant à haute température dans la zone riche, des suies sont formées selon un processus qui n'est toujours pas pleinement élucidé à ce jour. Une chandelle est un exemple de ce type de flamme : la chaleur permet à la paraffine liquéfiée de remonter la mèche par effet de capillarité, de se vaporiser puis de s'embraser au contact de l'oxygène.

Une flamme de prémélange est constituée des deux gaz nécessaires à la combustion déjà combinée, et requiert uniquement une source de chaleur suffisante pour s'embraser. Elle ne

nécessite aucun contact avec l'oxygène ambiant, produisant une combustion parfaite, expliquant son apparence uniforme et sans suies. La flamme d'un poêle à gaz en est un parfait spécimen et la couleur bleutée uniforme découle de l'absence de suie (voir figure 2.3b).

Le modèle atomique de Bohr introduit en 1913 est une amélioration au modèle de Rutherford inspiré de l'observation des planètes de notre système gravitant autour du soleil. Une molécule stable possède des atomes constitués d'un noyau et d'électrons à orbite fixe. La distance d'un électron à son noyau représente une énergie potentielle proportionnelle. Ce modèle permet d'expliquer la chaleur et lumière émanant d'une flamme, ce que nous accomplirons dans les sections suivantes.

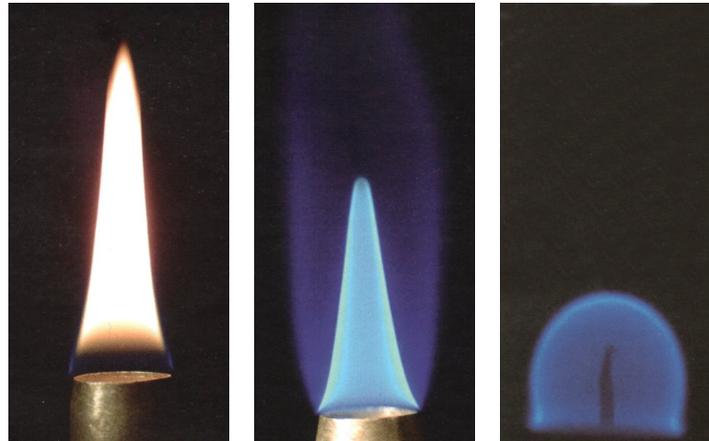
2.2 Chaleur

En présence d'une température suffisamment élevée, un seuil critique d'intercollisions de molécules est introduit, menant vers un état instable. Elles tentent alors de rétablir une stabilité en se recombinaut entre elles, créant des nouveaux types de molécules stables et des molécules instables à électrons libres, dénommés radicaux. Ces activités entraînent le changement d'orbite de certains électrons, libérant une énergie potentielle qui est captée par les radicaux puis convertie en énergie cinétique. Ces mouvements de radicaux occasionnent un très grand nombre de collisions, causant une augmentation de la température au sein du système.

2.3 Illumination

Deux causes distinctes sont à la base de l'illumination observable lors de la combustion. La première se trouve lors de la quête d'un état stable d'un électron. Son changement d'orbite occasionne sporadiquement l'émission d'un photon d'une longueur d'onde variant selon les molécules impliquées dans la réaction chimique. La plage bleutée du spectre de lumière de plusieurs flammes courantes en est issue et le phénomène est désigné sous le terme luminescence. Son existence n'implique pas la génération de chaleur, comme nous pouvons constater en présence de lucioles qui en sont également source.

La seconde cause provient de la radiation de corps noir. Ce type de particule possède la propriété de capturer les photons ambiants et d'ainsi engendrer un accroissement de leur température interne. L'énergie est retransmise sous forme de rayonnement thermique imperceptible par l'oeil humain, et de l'émission de nouveaux photons dont la longueur d'onde (et donc la couleur perçue) est tributaire à la température de la particule. La présence d'un excès de carbu-



(a) Flamme régulière : l'oxygène entre en contact avec le carburant sur le front de la flamme, causant leur oxydation.

(b) Flamme de prémélange : oxygène est mêlé au carburant en amont de la réaction, produisant une combustion parfaite du carburant.

(c) Flamme en micro-gravité : forme sphérique et combustion parfaite.

FIGURE 2.3 – Flamme avec apport externe de carburant (oxygène) versus flamme de prémélange. Source : [Boy06].

rant à température élevée (avoisinant 1500°C), occasionne une combustion imparfaite, catalysée à la création de fines particules de carbone présentant les caractéristiques d'un corps noir. Ces suies, conjuguées aux différentes plages de températures à la surface de la flamme, expliquent les zones variées du spectre de lumière émise. La fumée noire que l'on peut parfois entrevoir est constituée des particules n'ayant pas une chaleur suffisante pour retransmettre les photons dans le spectre visuel perceptible.

2.4 Forme

Le principe d'Archimède est à la base de la forme de la flamme de diffusion. Les gaz carburants s'élèvent dans l'atmosphère selon la théorie de la dynamique des fluides. Une augmentation de la température de la flamme introduit une accélération du déplacement vertical des gaz, expliquant sa forme allongée et de la hauteur conséquente à sa température. En présence d'un environnement de micro-gravité dans l'espace, une flamme prend plutôt une forme sphérique due au manque de poussée vers le haut. Cette forme est aussi tributaire à la dispersion uniforme du combustible et de l'emplacement de l'oxygène. Elle présente les mêmes caractéristiques qu'une flamme de prémélange, puisque le déplacement du carburant par diffusion est suffisamment lent pour obtenir une oxydation complète du carburant (voir figure 2.3c).

Chapitre 3

Travaux antérieurs

[...]
She took my father from my life
Took my sister and brothers oh
I watched her torturing my child
Feeble I was then but now I'm grown
Fire on Babylon
-SINEAD O'CONNOR (FIRE ON BABYLON)

Un domaine de recherche prolifique visant la restitution virtuelle des flammes s'est développé autour du phénomène. Ceci peut s'expliquer par l'antique fascination que nous lui portons et de son ubiquité dans les divers environnements synthétiques que nous façonnons. Malgré son apparence longuement observée, une flamme reste difficilement synthétisable, son état subordonné à un grand nombre de variables corrélées et fluctuant énormément entre deux moments avoisinants. Lors de la simulation d'une flamme en infographie, plusieurs approches se présentent à nous, réparties essentiellement sous trois catégories. Nous les désignons ici comme les modèles physiques, les modèles empiriques et les modèles de capture. Les sections qui suivent décrivent plus exhaustivement les caractéristiques de chacun, ainsi que les travaux majeurs qu'ils ont engendrés.

3.1 Modèle physique

Les modèles physiques visent une fidèle synthèse des différents aspects du phénomène. Une attention particulière est ainsi accordée au déplacement des particules de carburant, à la température du système, à l'émission de suies, à l'emplacement du front de combustion et même occasionnellement à la mécanique de propagation des flammes sur différents matériaux. Cette

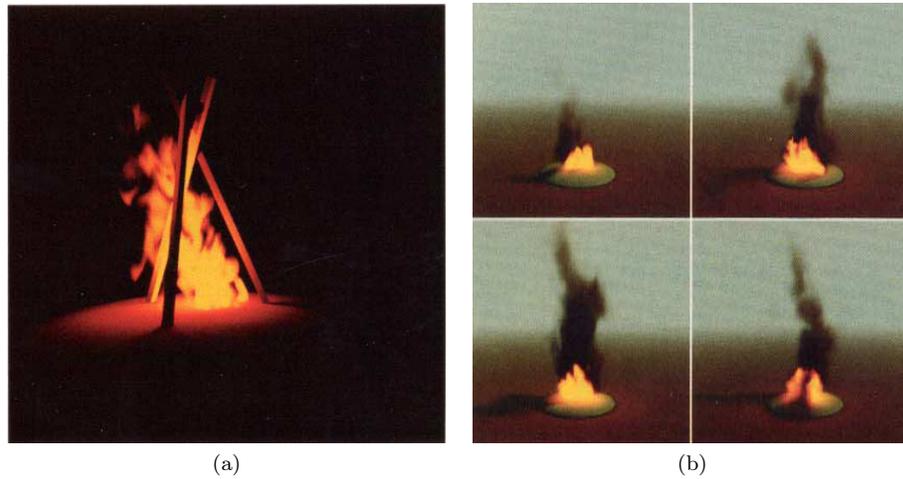


FIGURE 3.1 – Résultats de Stam et Fiume issus d’amalgames représentant des gaz soumis aux forces d’advection et du vent. Source : [SF95].

approche méticuleuse rend possible la création de visuels remarquables, mais à un coût d’évaluation considérable pouvant malheureusement nécessiter plusieurs heures de calcul par trame. Ce type de modèle s’avère de circonstances pour toute application demandant une apparence saisissante, sans les contraintes d’un rendu en temps réel, tel que dans l’industrie cinématographique. Il est également requis pour toute application où l’exactitude du comportement doit fidèlement suivre les principes physiques sous-jacents, tel que la simulation de la propagation d’incendie. Comme le système dépend d’un grand nombre de variables corrélées, la génération de résultats spécifiques est ardue, la nature chaotique des flammes se prêtant mal à une paramétrisation directe.

Dans leur article, Stam et Fiume [SF95] présentent une modélisation du comportement des gaz de diffusion soumis aux forces d’advection et du vent. Ils ont recours à des amalgames (*blobs*) aux formes altérées sous les forces reçues. Les contours non uniformes générés évitent de trahir trop rapidement leur nature synthétique, l’œil humain excellent à repérer des motifs réguliers. Une attention particulière est portée à rendre la technique propice à la simulation des flammes (voir figure 3.1).

Quelques années plus tard, Stam [Sta99] introduit une nouvelle méthode de résolution d’un système de fluide demeurant inconditionnellement stable, contrairement aux méthodes précédentes. Un système de simulation physique requiert l’évaluation d’équations décrivant un phénomène visé à intervalles réguliers. Cette discrétisation du temps implique que le système ne peut aspirer qu’à une approximation de la réalité, puisque les flammes exprimées sous forme de fluides existent dans un univers continu. Dans le cas d’une simulation instable, un pas de

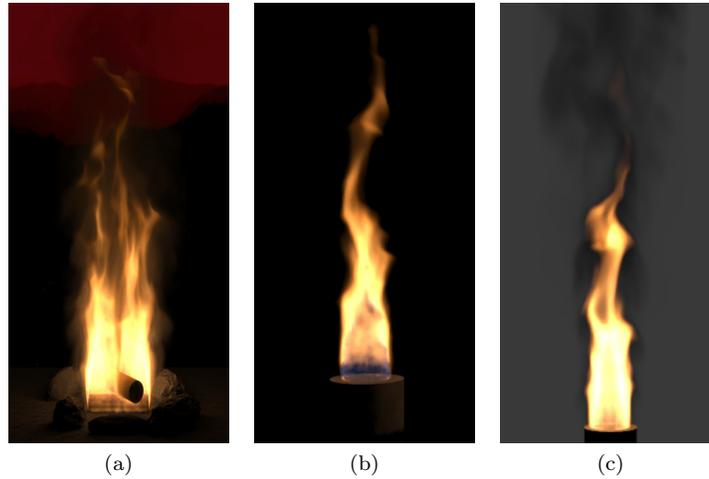


FIGURE 3.2 – Résultats de Nguyen et al. issues d’une simulation de fluide Navier-Stokes du combustible et des gaz secondaires. Source : [NFJ02].

discrétisation trop grossier génère un manque de précision menant rapidement à des résultats dégénérés, l’erreur s’accumulant à chaque itération. Afin de conserver une précision au-dessus d’un seuil raisonnable, ces systèmes nécessitent un grand nombre d’évaluations par seconde, introduisant un coût considérable. La méthode de simulation stable de Stam permet d’obtenir des résultats cohérents en présence d’un pas de temps plus élevé, et d’ainsi réduire sa fréquence d’évaluation. Bien que n’excellant pas en précision, elle demeure néanmoins raisonnable pour un modèle visuel ne requérant pas la précision d’une simulation scientifique.

Nguyen et al. [NFJ02] ont recours au système d’équations incompressible de Navier-Stokes pour générer une simulation physique de la combustion très fidèle à la réalité. Le combustible vaporisé ainsi que les gaz secondaires issus de l’oxydation sont modélisés indépendamment et déplacés sous les forces d’advection. Ceci rend possible la synthèse d’un large éventail de types de flammes, allant de laminaires à turbulentes. La luminescence issue de l’oxydation ainsi que de la radiation de corps noir des suies est recrée, restituant une apparence réaliste en exhibant une fumée en l’absence d’une température suffisante pour un rayonnement perceptible des suies. Le système permet une détection et réaction à des obstacles bloquant le flot des gaz et au final, présente un visuel fort impressionnant (voir figure 3.2).

Hong et al. [HSF07] poursuivent les travaux de Nguyen et al. [NFJ02] en améliorant l’apparence chaotique du front de flamme turbulente. En introduisant la méthode *level set* et de *jump condition* aux équations de Navier-Stokes, ils parviennent à générer un motif cellulaire et de re-plis au déplacement des gaz, une amélioration à l’apparence lisse initiale. Les travaux de Nguyen et al. déjà excellents voient la simulation des flammes tumultueuses, comme les explosions ou

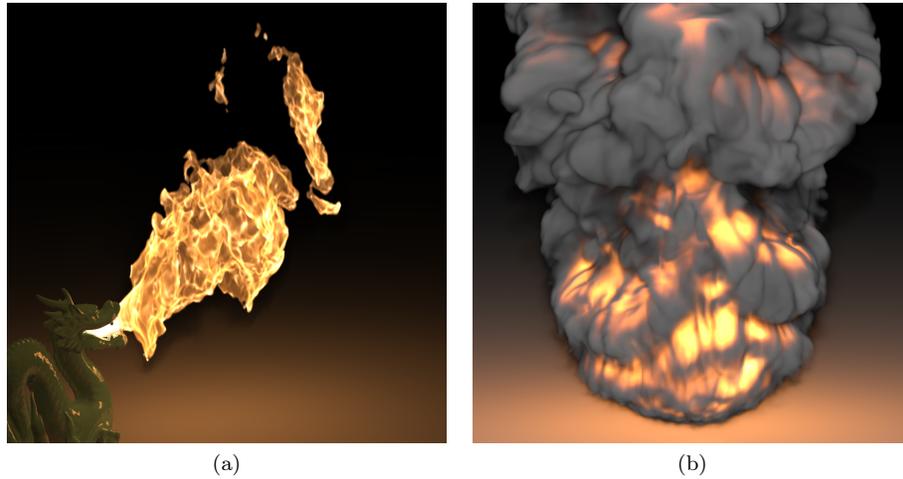


FIGURE 3.3 – Résultats de Hong et al. reprenant les travaux de Nguyen et al. tout en leur ajoutant un motif cellulaire, restituant mieux le front des flammes turbulentes. Source : [HSF07].

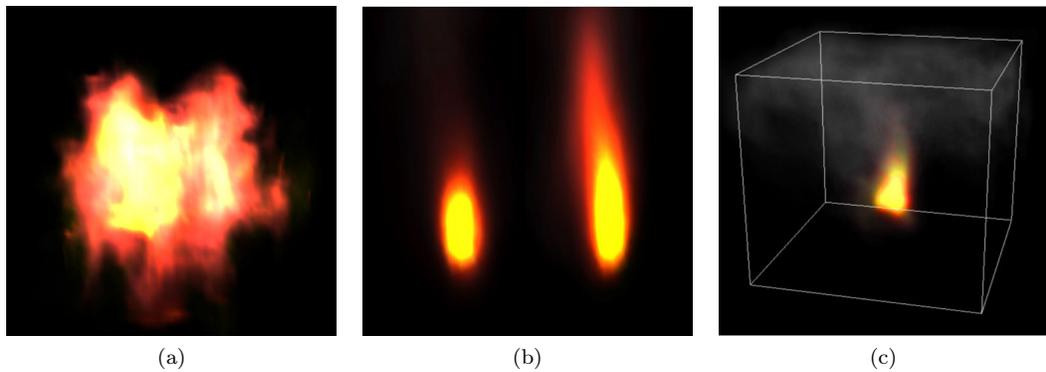


FIGURE 3.4 – Résultats de Melek et Keyser intégrant la méthode de fluide stable développée par Stam, aux travaux de Nguyen et al. Source : [MK02].

jets de gaz à haut débit, gagner grandement en réalisme (voir figure 3.3).

Melek et Keyser [MK02] introduisent un modèle s’inspirant également de la simulation des gaz de Nguyen et al. mais en ayant recours à la méthode de fluide stable développée par Stam [Sta99]. La simulation de fluide donne lieu au déplacement des gaz de carburant et de comburant dans une grille. Une oxydation se produit à leur contact mutuel, générant chaleur et gaz d’échappement qui évolueront également au sein du système. Les trois types de gaz impliqués dans la combustion sont ainsi considérés. Au final, ces informations sont exploitées pour la réalisation d’un rendu volumétrique s’attendant à la tâche de transformer ces informations tridimensionnelles en une représentation visuelle bidimensionnelle (voir figure 3.4).

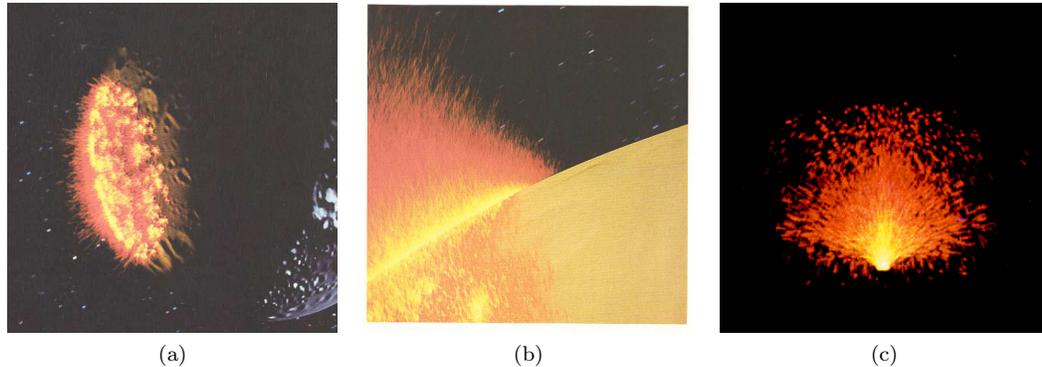


FIGURE 3.5 – Résultats d’un système de particules conçu par Reeves et intégré à la production cinématographique ‘*Star Trek II : The Wrath of Khan*’. Source : [Ree83].

3.2 Modèle empirique

Les modèles empiriques visent la synthèse du comportement et de l’aspect visuel d’une flamme, sans toutefois limiter l’implémentation par le maintien d’une authenticité des théories physiques et chimiques sous-jacentes. La rapidité et la simplification de la paramétrisation du système sont habituellement privilégiées face à l’exactitude. Les itérations rapides entre chaque ajustement s’en trouvent facilitées tout en prodiguant un contrôle plus intuitif et direct sur les résultats escomptés. Ils peuvent malgré tout s’inspirer des modèles physiques existants, mais en s’octroyant la latitude d’approximer les équations et simplifier plusieurs mécaniques. Certains systèmes offrent un rendu des flammes en temps réel où il est impossible de distinguer les trames individuelles de l’animation. D’autres parviennent à un rendu en temps interactif (entre 1 et 30 trames par seconde), où la suite d’images est plus saccadée, mais le temps de réponse est suffisamment court pour qu’une intervention sur le système affiche les résultats dans un laps de temps raisonnable. Plusieurs de ces modèles adoptent l’utilisation d’un système de particules, introduit par Reeves [Ree83].

Mis en oeuvre pour la première fois dans le film ‘*Star Trek II : The Wrath of Khan*’, Reeves [Ree83] introduit un système de particules pour la représentation de phénomènes diffus aux contours flous. Depuis, il est exploité dans la grande majorité des jeux vidéo, sa simplicité et ses performances étant tout indiquées pour simuler divers phénomènes sans forme et volume précis, tels que la fumée ou des étincelles (voir figure 3.5c). Nous pouvons visualiser un système de particules comme une fontaine possédant un émetteur de particules à sa base, chaque particule servant à représenter une ou plusieurs gouttelettes d’eau. Celles-ci possèdent des propriétés propres à l’application définissant leur comportement, telles qu’une durée de vie, une position, une vitesse et une fonction de trajectoire. Leur position et d’autres propriétés sont régulièrement



FIGURE 3.6 – Résultats de Wei et al. combinant une simulation de fluide à une technique de *texture splats* pour les détails de hautes fréquences. Source : [WLMK02].

mises à jour et utilisées lors du rendu de la scène, en affichant une petite gouttelette d'eau orientée caméra, leur apparence texturée diffuse se fondant bien les unes aux autres. L'incrément du pas de temps amène le retrait des particules périmées ainsi que l'insertion de nouvelles particules au système. La première application cinématographique fut la création d'un énorme front de flammes engloutissant une planète entière dans le film '*Star Trek II : The Wrath of Khan*' (voir figures 3.5a et 3.5b). Bien qu'un excellent point de départ pour la création de flammes par sa simplicité d'implémentation et sa rapidité d'exécution, ce type de système peine malgré tout à restituer le contour bien défini d'une flamme et son gradient de couleur distinctif. Les performances relativement faibles du matériel informatique conjugué au nombre élevé de particules requises pour l'obtention de visuels convenables rendaient l'atteinte du rendu temps réel inaccessible à cette période.

Wei et al. [WLMK02] introduisent le modèle lattice Boltzmann (LBM) dans la résolution d'une simulation de fluide. Sa nature facilite la parallélisation des calculs, rendant plus aisée la simulation de collisions en présence d'un environnement complexe. Ils en tirent profit en modélisant l'interaction d'une flamme à des forces externes telles que le vent et les obstacles, pour finalement synthétiser sa forme générale. Comme les détails de hautes fréquences sont manquants à ce stade, une technique de *texture splats* introduite par King et al. [KCR99] y remédie en appliquant une série d'images turbulentes affichées successivement, et dont l'opacité individuelle est pondérée selon une fonction de lissage. Les résultats finaux obtiennent un rendu en temps réel et représentent un grand progrès en terme de qualité visuelle. Malgré tout, ils demeurent insatisfaisants pour l'apparence d'une flamme laminaire, les contours plus diffus générés restituant mal la réalité (voir figure 3.6).

Beaudoin et al. [BPP01] s'attellent à la simplification de la paramétrisation du feu ainsi qu'à

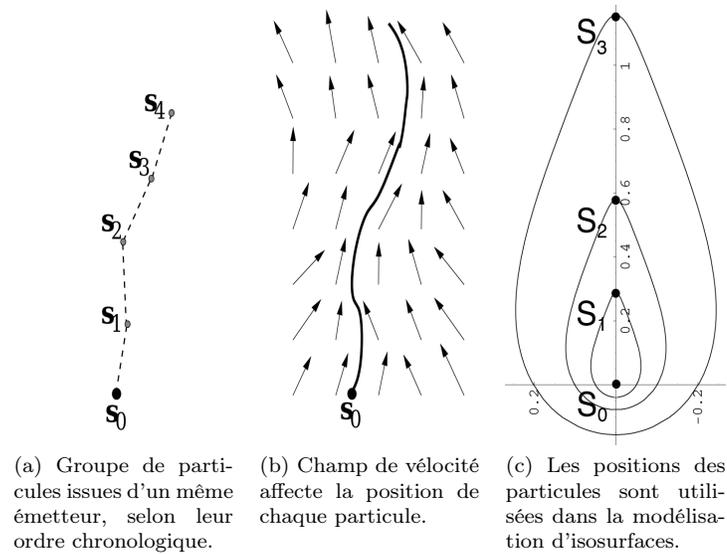


FIGURE 3.7 – Représentation 2D d'une chaîne de particules et son utilisation dans la modélisation d'une flamme. Chaque particule S_i possède une position relative aux axes X/Y et est chronologiquement ordonnée selon son indice i . Source : [BPP01].

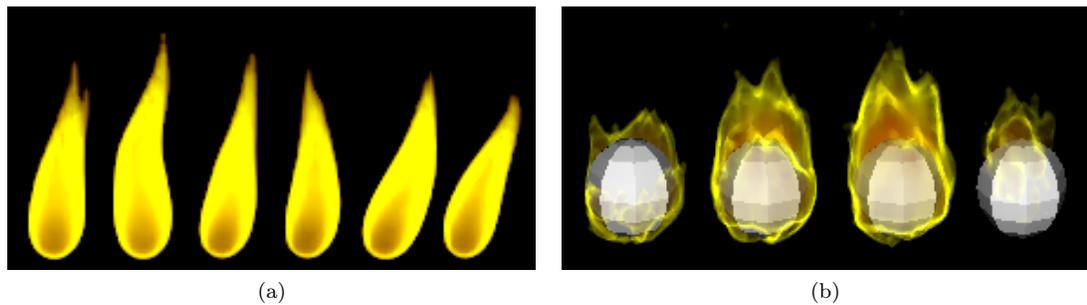


FIGURE 3.8 – Résultats de Beaudoin et al. issus d'isosurfaces modélisées à l'aide d'une chaîne de particules. Source : [BPP01].

la modélisation de sa propagation sur différentes surfaces. Ils introduisent le concept d'une chaîne de particules (voir figure 3.7a) définie comme l'ensemble des particules générées par un même émetteur, et mis à profit dans la représentation du squelette d'une flamme. Chaque segment de la chaîne définit une isosurface servant à un rendu volumétrique de la flamme (voir figure 3.7c). Les particules sont affectées par un champ de vitesse représentant les forces externes telles que le vent et la convection des gaz et un feu est constitué du regroupement de plusieurs flammes. L'affichage des résultats s'accomplit par une méthode de rendu volumétrique, se prêtant donc peu à des performances en temps réel (voir figure 3.8).

Lamorlette et Foster [LF02] émettent le constat que l'apparence chaotique d'une flamme peut être en grande partie modélisée stochastiquement. Ils reprennent le concept d'une chaîne de



FIGURE 3.9 – Résultats de Lamorlette et Foster reprenant le concept de la chaîne de particules de Beaudoin et al. augmenté d’une méthode stochastique pour les détails de hautes fréquences et utilisant un rendu volumétrique, menant à la création de séquences pour la production cinématographique ‘*Shrek*’. Source : [LF02].

particules définissant le squelette d’une flamme pour sa forme générale et permettant l’emphase sur un contrôle artistique des résultats. Ce squelette déforme un isocontour subissant ensuite une rotation autour de l’axe des Y pour obtenir le volume de la flamme et menant à une apparence visuelle impressionnante par l’entremise d’un rendu volumétrique. La méthode a été déployée dans la production cinématographique ‘*Shrek*’, pour la synthèse d’une grande variété de flammes (voir figure 3.9) mais à un taux de 0.37 image/s sur du matériel datant de 2002, les résultats sont loin d’être en temps réel.

Horvath et Geiger [HG09] amènent une approche réconciliant le temps de calcul réduit des méthodes empiriques à la précision et la qualité visuelle des simulations physiques. Ils y parviennent en dissociant le mouvement général d’une flamme de ses détails de hautes fréquences. Un système de particules modélise le flot des gaz tridimensionnellement, accordant un plein contrôle artistique sur la propagation des flammes. Les informations de masse, carburant et impulsion sont subséquentement propagées à un ensemble de simulations de fluide bidimensionnelles, divisé en plusieurs tranches orientées caméra. La complexité de la simulation étant réduite d’un ordre de grandeur et hautement parallélisable, chaque tranche de simulation se voit résolue sur le GPU par une simulation de Navier-Stokes pouvant avoir une résolution de grille supérieure à ce qu’un modèle tridimensionnel peut se permettre. Les informations de densité, carburant et température issues de chaque tranche sont recueillies, potentiellement altérées par un artiste pour plus de contrôle, puis converties en lumière et opacité à l’aide d’une fonction suivant les équations de radiation de corps noir. Le modèle est limité à un point de vue unique et n’atteint pas des performances interactives. Il permet malgré tout des itérations rapides et des résultats visuels



FIGURE 3.10 – Résultats de Horvath et Geiger issus d'un système de particules pour la modélisation du flot des flammes et de plusieurs simulations de fluide 2D orientées écran pour les détails de hautes fréquences. Source : [HG09].

remarquables, le rendant tout indiqué pour les productions cinématographiques pour lesquelles il est destiné (voir figure 3.10).

Fuller et al. [FKM⁺07] introduisent un rendu volumétrique temps réel aux résultats visuels intéressants et facilement paramétrables. Comme plusieurs auteurs précédents, ils y parviennent en dissociant la forme générale de la flamme de sa composante de hautes fréquences chaotique. La forme s'obtient en déformant le volume englobant d'une flamme (voir figure 3.11c) par l'entremise d'une simulation de particules ou toute autre alternative équivalente. Quant aux détails de la flamme, ils sont générés par un *bruit de Perlin amélioré* venant perturber l'accès à une texture de profil d'une flamme, menant à une apparence de turbulence. La création de cette texture de profil permet d'aisément diriger l'apparence du rendu volumétrique (voir figure 3.12).

Suite à l'observation d'une flamme de soie en action telle que présentée sur la figure 3.13a, Balci et Foroosh [BF06] décident d'aborder la problématique sous un tout nouvel angle. Ce type de simili flamme doit son existence au théâtre grec antique et peut toujours être observée dans certains faux foyers modernes, étant apte à restituer le phénomène adéquatement. Elle consiste en un tissu de soie ancré à sa base, illuminé par une lumière rougeâtre, et se mouvant au rythme d'un souffle ascendant. Balci et Foroosh proposent de dupliquer numériquement ce subterfuge, en synthétisant la flamme laminaire d'une bougie à l'aide d'une simulation de tissu, et la flamme plus complexe d'une torche à l'aide d'un ensemble de ces simulations. Ils parviennent à des résultats visuels intéressants tout en étant avares en coût de calcul, menant à des performances en temps réel. Malheureusement, le changement de perspective d'un observateur révèle rapidement l'artifice, le placage d'une texture sur tissu trahissant sa nature bidimensionnelle sans épaisseur (voir figure 3.13).

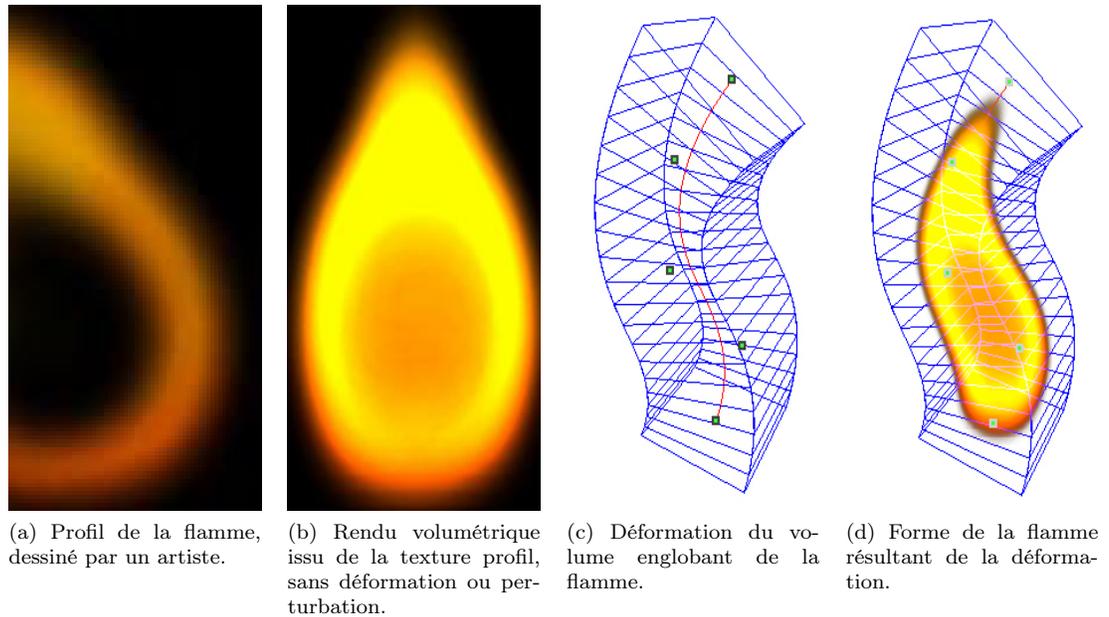


FIGURE 3.11 – Technique de rendu volumétrique présentée par Fuller et al. Source : [FKM⁺07].

Dans sa thèse de doctorat, Bridault [Bri07] reprend le concept de chaîne de particules qu’il met à profit dans la paramétrisation d’une surface NURBS. Il y conjugue un champ de vélocité issu d’une simulation de fluide et contribuant au déplacement des particules. Un nombre varié de formes de flammes laminaires peut être aisément synthétisé grâce à la flexibilité de la configuration rendue possible par les chaînes de particules. Pour les flammes plus complexes, telles qu’une torche ou un feu de camp, il suffit de combiner plusieurs systèmes. La simulation de fluide donne lieu à une dynamique de flamme vraisemblable et son approche mène à d’excellents résultats visuels (voir figure 3.14), tout en conservant des performances en temps réel.

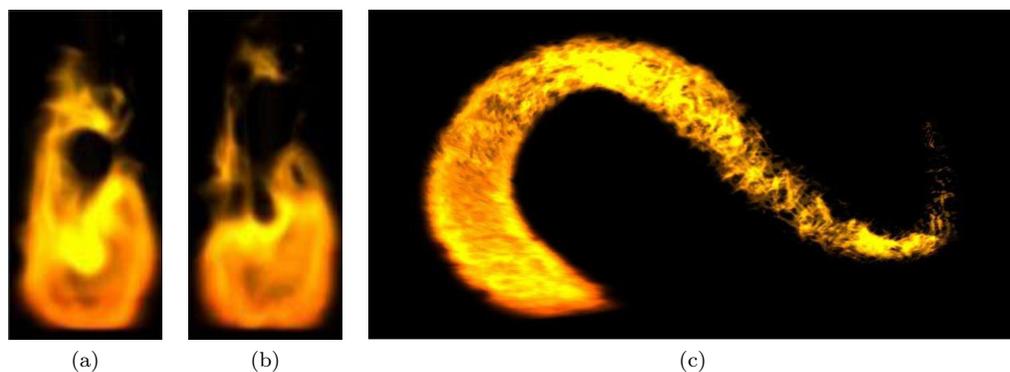


FIGURE 3.12 – Résultats de Fuller et al. issus d’un rendu volumétrique utilisant la déformation d’un volume englobant pour la forme générale, et d’un *bruit de Perlin amélioré* pour les détails de hautes fréquences. Source : [FKM⁺07].

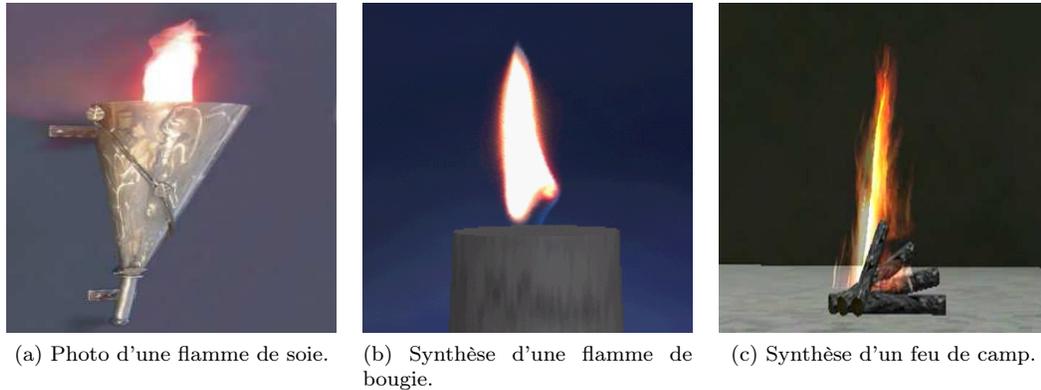


FIGURE 3.13 – Résultats de Balci et Foroosh utilisant une simulation de tissu simulant l'apparence d'une flamme de soie. Source : [BF06].

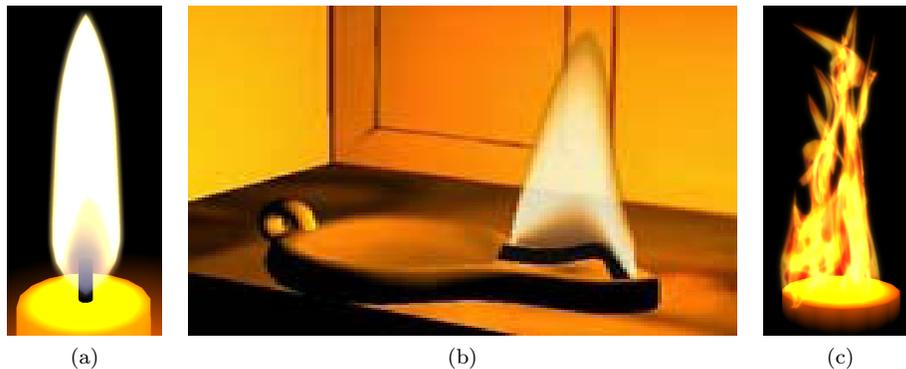


FIGURE 3.14 – Résultats de Bridault issus de chaînes de particules paramétrant une surface NURBS texturée. Source : [Bri07].

3.3 Modèle capture

Les modèles de capture se spécialisent dans la numérisation d'un phénomène pour être en mesure de le restituer virtuellement à volonté. Une de ses applications est couramment mise en oeuvre dans les jeux vidéo, lors de l'affichage d'une courte séquence vidéo de flamme cyclique, plaquée sur une texture orientée vers la caméra. Des modèles plus avancés entreprennent d'également analyser le contenu numérisé pour être en mesure de générer de nouvelles séquences. Ces systèmes sont principalement limités par leur nature non interactive et la quantité de stockage requise pour contrer la nature cyclique de courtes séquences.

Schödl et al. [SSSE00] présentent une technique de synthèse de vidéo acyclique. Ils analysent préalablement les trames d'une séquence vidéo dans le dessein d'en extraire l'intersimilitude. Ces résultats permettent d'afficher subséquentement une série infinie de courts intervalles vidéo en sautant aléatoirement d'un segment à un autre, tout en s'assurant de préserver la fluidité de

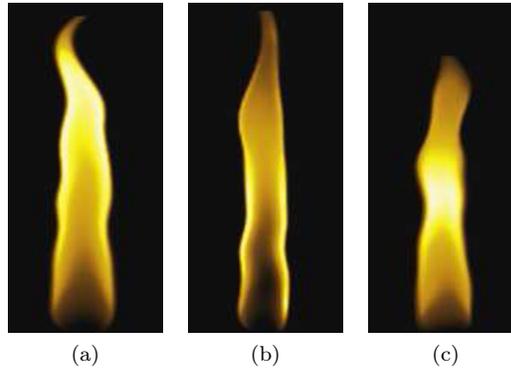


FIGURE 3.15 – Résultats de Stich et Magnor produits à l’aide d’une analyse statistique de l’apparence d’une flamme. Source : [SM05].

la séquence en évaluant la similitude de la région de départ et d’arrivée. Bien que la technique permette de restituer tout type de flamme observable, elle reste lourdement handicapée par le manque d’interaction avec son environnement et l’énorme quantité de données requise pouvant facilement se retrouver dans le domaine des méga-octets même compressées, tout dépendant de la longueur et de la résolution de la séquence vidéo.

Stich et Magnor [SM05] proposent une méthode issue du domaine du traitement d’images. Ils extraient des caractéristiques d’apparence pour chaque trame d’une séquence vidéo d’une flamme qui sont ensuite interprétées par un modèle d’apprentissage statistique. Ce modèle est ensuite utilisé dans la production de nouvelles séquences, respectant l’apparence de base de la flamme (voir figure 3.15).

Chapitre 4

Implémentation

*Man fears the darkness, and so he scrapes away
at the edges of it with fire.*
-REI AYANAMI (NEON GENESIS EVANGELION)

La prémisses de notre système est l'obtention de résultats minimisant le coût de calcul tout en offrant une apparence de flamme convaincante. Ceci nous a menés à une analyse des travaux déjà accomplis dans la catégorie des modèles empiriques. Nous inspirant de l'excellente apparence visuelle présente dans la thèse de Bridault [Bri07] combinée aux bonnes performances issues de la méthode de Balci et Foroosh [BF06], nous avons opté pour une approche hybride palliant leurs points faibles respectifs.

Bridault [Bri07] présente des flammes aux formes variées dont le dynamisme conserve une apparence réaliste en présence de forces externes, grâce à une simulation de fluide. Il y conjugue des chaînes de particules pour modéliser la forme à l'aide de surfaces NURBS et leurs différentes formes s'obtiennent en variant la configuration des chaînes de particules et en combinant plusieurs systèmes ensemble. Leur projet étant destiné à la restitution virtuelle des ruines d'un forum, tel qu'il pouvait être perçu au 2^e siècle apr. J.-C., il porte une attention particulière à l'illumination issue d'une chandelle et d'une lampe à l'huile afin de bien retransmettre l'ambiance des lieux en ces temps ancestraux. Bien que sa technique obtient une performance en temps réel, une simulation de fluide requiert néanmoins un temps de calcul substantiel se prêtant mal à un usage non exclusif dans un logiciel.

Balci et Foroosh [BF06] introduisent une synthèse de flamme extraite d'une simulation de tissu sur laquelle ils appliquent une texture. L'apparence demeure intéressante pour un point de vue fixe, et ils arrivent à simuler des flammes plus complexes en combinant plusieurs systèmes.

La simulation de tissu est plus avare de temps de calcul qu'une simulation de fluide de dimension équivalente, permettant d'aisément préserver des performances temps réel. Malheureusement, la nature bidimensionnelle du tissu révèle rapidement le subterfuge lors du changement du point de vue où l'on peut aisément percevoir les plans de flammes (voir figure 3.13c).

Nous proposons ainsi de reprendre la versatilité des chaînes de particules paramétrant une surface NURBS pour la modélisation d'une flamme, tel qu'avancé par Bridault[Bri07], mais d'extraire le champ de vélocité affectant les particules, d'un système de ressorts tridimensionnel plutôt que la simulation de fluide originale. Notre modèle permet ainsi de synthétiser une flamme laminaire d'apparence réaliste et à un faible coût de calcul tout en étant suffisamment flexible pour être élargi à d'autres types de flammes.

4.1 Modélisation

4.1.1 Système de particules

Nous inspirant de la thèse de Bridault [Bri07], nous avons opté de définir le volume de notre flamme à l'aide d'un ensemble de particules se déplaçant sous les forces d'un champ de vélocité et paramétrant une surface NURBS. Chaque particule est générée à partir d'émetteurs dont la position est déterminée selon son emplacement sur une mèche virtuelle et configurable par l'utilisateur. Son implémentation, telle que définie par Bridault [Bri07], peut-être observée à la figure 4.1. En nous inspirant de ces travaux, nous définissons une mèche virtuelle à l'aide de N chaînes de particules (ensemble des particules issues d'un même émetteur) selon la configuration observable à la figure 4.2. L'emplacement de chaque émetteur de particules est spécifié par l'utilisateur tout en demeurant restreint à un cube unitaire, ce qui permet d'ajuster la forme de la flamme à sa convenance lors du rendu, tout en utilisant une mise à l'échelle pour la taille désirée. Dans le cas d'une flamme laminaire dont la forme générale s'apparente à un cylindre, seules cinq chaînes de particules sont nécessaires (quatre périphériques et une centrale).

Lors de la mise-à-jour de la simulation, chaque émetteur génère une nouvelle particule se déplaçant à haute vitesse et possédant une espérance de vie limitée à quelques trames. Cette durée peut se concevoir comme le temps requis pour l'oxydation de tous les gaz émis à un instant précis, une longue période amenant une taille de flamme plus allongée et un plus grand nombre de particules à simuler. Lors de son existence, une particule voit sa position affectée à chaque itération par la valeur du champ de vélocité liée à son emplacement actuel. Cet emplacement étant restreint à l'intervalle d'un cube unitaire, l'indexage du champ de vélocité s'en trouve

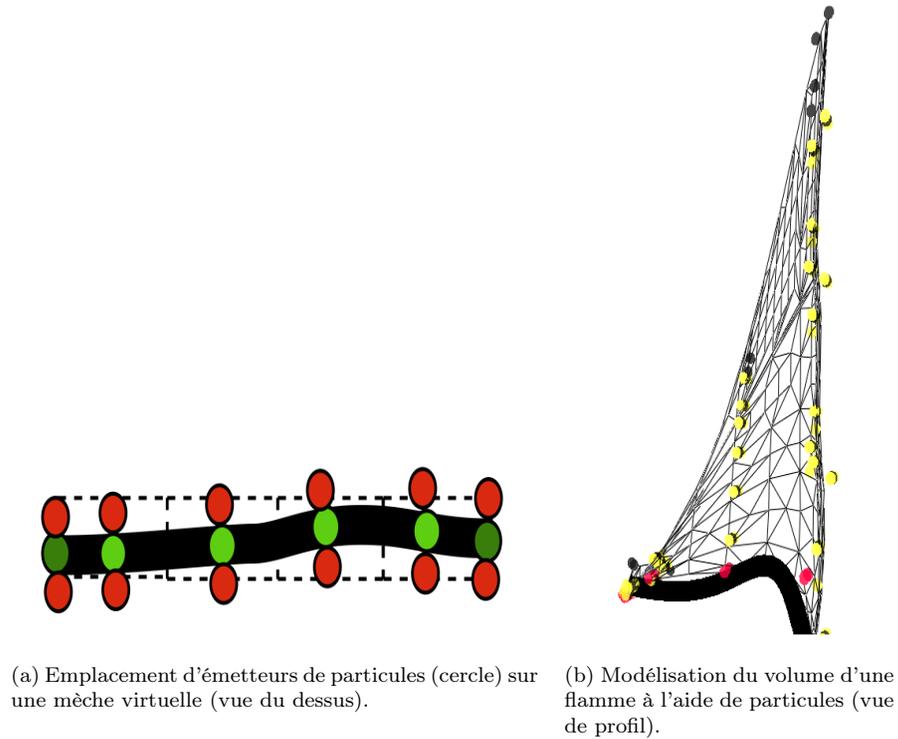


FIGURE 4.1 – Mèche et émetteurs de particules de l'implémentation de Bridault, pour la modélisation d'une flamme. Source : [Bri07].

facilité. Lorsqu'une particule se retrouve à l'extérieur de cette région, son déplacement est plutôt induit par sa vitesse précédente pondérée par un facteur d'atténuation. Lors de la mise-à-jour des particules, la valeur du champ de vitesse est modulée par une fonction de distribution de carburant associée à chaque émetteur et permettant un second contrôle sur la forme de la flamme. Considérant la forme cylindrique d'une flamme de bougie, un terme constant suffit pour les chaînes périphériques, tandis que la chaîne centrale est pondérée par un terme légèrement supérieur pour obtenir une pointe (voir procédure 1).

4.1.2 Surface NURBS

Une flamme présente une apparence aux formes courbes sans aspérités, qu'une modélisation aux angles trop réguliers restitue difficilement. Une surface NURBS est une représentation mathématique de géométrie couramment utilisée dans la modélisation de solide à surface continue. Elle a l'avantage d'être flexible, précise et d'avoir une influence localisée, ce qui permet de recréer une grande variété de géométries, allant d'une ligne droite aux courbes subtiles et élancées d'une voiture sport. Comme elle ne requiert qu'une liste de paramètres définissant sa surface, la quantité de données est réduite et ne nécessite aucune discrétisation du modèle lors de son

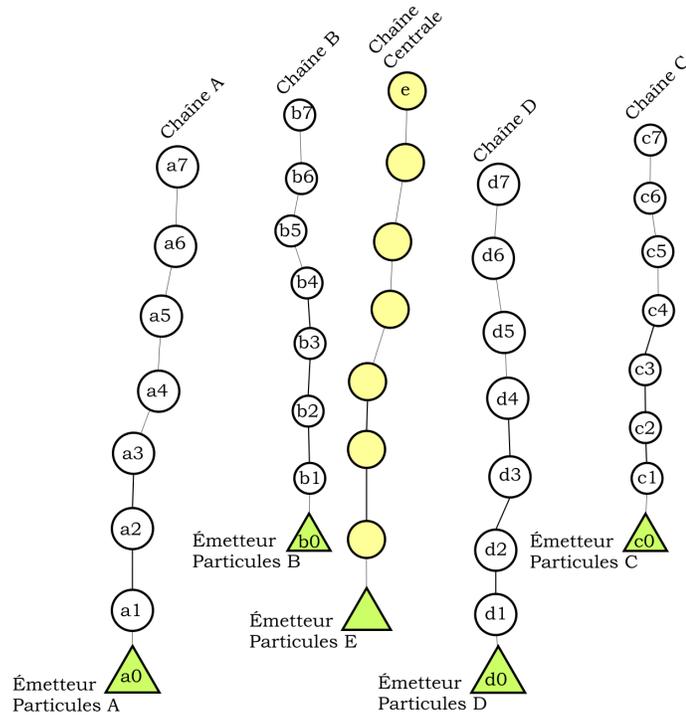


FIGURE 4.2 – Modélisation de notre flamme laminaire où quatre émetteurs (triangles) génèrent des chaînes de particules (cercles blancs) spécifiant la surface extérieure du volume de la flamme. La pointe est générée par la position de la plus ancienne particule émise par la chaîne centrale (cercle e) lors de la paramétrisation de la surface NURBS.

stockage, contrairement à une modélisation par triangles. L'évaluation d'une surface NURBS est incluse dans toute implémentation de OpenGL, ce qui permet une utilisation standardisée.

Une surface NURBS est définie par son ordre O , un tableau de points de contrôle de dimensions $I \times J$ et deux vecteurs de poids associés aux dimensions du tableau précédent. L'ordre d'une surface NURBS, également référé comme degré D , où $D = O - 1$, définit le nombre de points de contrôle contribuant à la position de tout emplacement de la surface. En pratique, une surface NURBS d'ordre 4 (degré 3) dite cubique est habituellement utilisée dans la représentation de surface pour sa flexibilité à un coût de calcul raisonnable tout en demeurant numériquement stable. Les points de contrôle définissent les positions vers lesquelles la surface doit tendre et leur influence sur chaque emplacement de la surface est fixée par des vecteurs de poids pondérant leur contribution. La figure 4.3 présente une surface NURBS (en vert) tendant vers ses points de contrôle (en rouge) selon deux vecteurs de poids non visibles.

Notre représentation de flamme laminaire utilise un tableau de points de contrôle de taille $I \times J$ où :

Procédure 1 Mise-à-jour du système de particules

```

Pour chaque chaîne:
{
  Ajouter particule à chaîne courrante
  Pour chaque particule dans chaîne courrante:
  {
    Si( position entre [0,0,0] et [1,1,1] ):
    {
      vitesse = ChampVitesse(position) * Carburant(chaîne)
    }
    Sinon:
    {
      vitesse = vitesse précédente * atténuation
    }
    position += vitesse

    vie -= delta temps
    Si( vie <= 0 ):
    {
      détruire particule
    }
  }
}

```

$$\begin{cases} I = P + 2 & P = \text{Nb. particules par chaîne} \\ J = N + D & N = \text{Nb. chaînes périphériques} \end{cases}$$

Ainsi, une ligne j du tableau contient la position de l'émetteur j , les P positions des particules de la $j^{\text{ième}}$ chaîne et la dernière position de la chaîne centrale (e). Dans le cas d'une flamme laminaire, nous désirons générer une forme lisse sans discontinuités visibles, lorsqu'observée latéralement. Pour y arriver, nous répétons les D premières lignes du tableau, à la fin. Pour aider à la compréhension, imaginons une feuille de papier roulée en un cylindre, elle nécessite un certain chevauchement du début et de la fin pour le maintien d'une base en forme de cercle. Cette restriction peut être relaxée verticalement puisque nous n'observons pas la flamme du dessous, ce qui permet de reproduire la forme conique plus aisément. Comme nous désirons une flamme attachée à sa mèche et se terminant bien à la pointe, nous configurons le vecteur de poids vertical de telle façon à ce que la surface soit forcée de passer par le premier et le dernier point de chaque chaîne en répétant la même valeur aux O premières et dernières colonnes du vecteur de poids vertical. Ceci force l'évaluation de la surface NURBS à pondérer uniquement le premier et le dernier point de contrôle en ces deux emplacements. Le vecteur de poids horizontal doit uniquement s'assurer que chaque emplacement de la surface est évalué par O points de contrôle

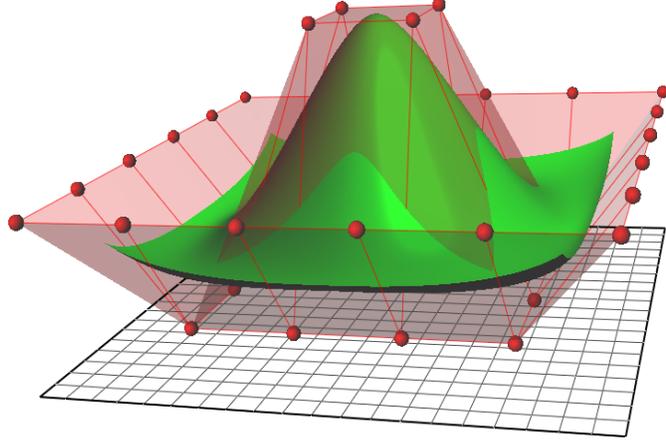


FIGURE 4.3 – Exemple d’une surface NURBS (vert) avec ses points de contrôle (rouge). Source : Wikipedia par l’utilisateur Chrschn.

Tableau points de contrôle :	<table style="border-collapse: collapse; text-align: center;"> <tr><td>a0</td><td>a1</td><td>a2</td><td>a3</td><td>a4</td><td>a5</td><td>a6</td><td>a7</td><td>e</td></tr> <tr><td>b0</td><td>b1</td><td>b2</td><td>b3</td><td>b4</td><td>b5</td><td>b6</td><td>b7</td><td>e</td></tr> <tr><td>c0</td><td>c1</td><td>c2</td><td>c3</td><td>c4</td><td>c5</td><td>c6</td><td>c7</td><td>e</td></tr> <tr><td>d0</td><td>d1</td><td>d2</td><td>d3</td><td>d4</td><td>d5</td><td>d6</td><td>d7</td><td>e</td></tr> <tr><td>a0</td><td>a1</td><td>a2</td><td>a3</td><td>a4</td><td>a5</td><td>a6</td><td>a7</td><td>e</td></tr> <tr><td>b0</td><td>b1</td><td>b2</td><td>b3</td><td>b4</td><td>b5</td><td>b6</td><td>b7</td><td>e</td></tr> <tr><td>c0</td><td>c1</td><td>c2</td><td>c3</td><td>c4</td><td>c5</td><td>c6</td><td>c7</td><td>e</td></tr> </table>	a0	a1	a2	a3	a4	a5	a6	a7	e	b0	b1	b2	b3	b4	b5	b6	b7	e	c0	c1	c2	c3	c4	c5	c6	c7	e	d0	d1	d2	d3	d4	d5	d6	d7	e	a0	a1	a2	a3	a4	a5	a6	a7	e	b0	b1	b2	b3	b4	b5	b6	b7	e	c0	c1	c2	c3	c4	c5	c6	c7	e
a0	a1	a2	a3	a4	a5	a6	a7	e																																																								
b0	b1	b2	b3	b4	b5	b6	b7	e																																																								
c0	c1	c2	c3	c4	c5	c6	c7	e																																																								
d0	d1	d2	d3	d4	d5	d6	d7	e																																																								
a0	a1	a2	a3	a4	a5	a6	a7	e																																																								
b0	b1	b2	b3	b4	b5	b6	b7	e																																																								
c0	c1	c2	c3	c4	c5	c6	c7	e																																																								
Vecteur Poids U :	<table style="border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> </table>	0	1	2	3	4	5	6	7	8	9	10	11																																																			
0	1	2	3	4	5	6	7	8	9	10	11																																																					
Vecteur Poids V :	<table style="border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>5</td><td>5</td><td>5</td></tr> </table>	0	0	0	0	1	2	3	4	5	5	5	5																																																			
0	0	0	0	1	2	3	4	5	5	5	5																																																					

TABLE 4.1 – Exemple de la paramétrisation d’une surface NURBS pour une configuration de quatre chaînes périphériques de huit particules (voir figure 4.2).

de chaînes différentes, ce qui s’obtient en assignant une valeur incrémentée uniformément à chaque indice. Les équations 4.1, 4.2 et 4.3 présentent la paramétrisation de la surface NURBS, tandis que le tableau 4.1 offre un exemple de son implémentation pour les valeurs de la figure 4.2.

$$\text{Vecteur Poids } U(j) = j - 1 \quad \forall j \in [1 : J + O] \quad (4.1)$$

$$\text{Vecteur Poids } V(i) = \begin{cases} 0 & 1 \leq i \leq O \\ i - O & O < i \leq I \\ I - O - 1 & I < i \leq I + O \end{cases} \quad (4.2)$$

$$\text{Tableau points } T(i, j) = \left[\begin{array}{cccccc} n_{1,0} & n_{1,1} & \dots & n_{1,P-1} & e & \\ \dots & \dots & \dots & \dots & \dots & \\ n_{N,0} & n_{N,1} & \dots & n_{N,P-1} & e & \\ n_{1,0} & n_{1,1} & \dots & n_{1,P-1} & e & \\ \dots & \dots & \dots & \dots & \dots & \\ n_{D,0} & n_{D,1} & \dots & n_{D,P-1} & e & \end{array} \right] \begin{array}{l} N \text{ chaînes} \\ \text{Répétition } D \text{ chaînes} \end{array} \quad (4.3)$$

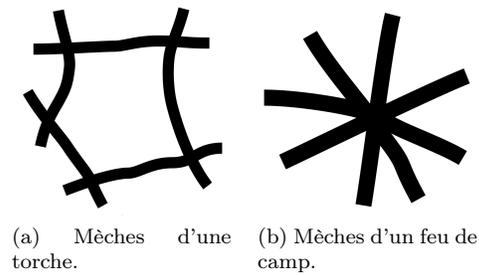


FIGURE 4.4 – Deux configurations de mèches suggérées par Bridault, pour la modélisation de flammes complexes. Source : [Bri07].

Comme nous avons opté pour détruire une particule uniquement à la fin de sa durée de vie et que chacune des chaînes émet une particule de même longévité à l'intérieur de la même trame, nous obtenons un nombre d'éléments identique dans chacune des chaînes. Dans le cas où nous aurions opté pour une stratégie de destruction ne respectant pas ce critère, tel que le retrait des particules à l'extérieur des limites du champ de vitesse, le système présenterait alors des chaînes de tailles potentiellement inégales. Puisque notre tableau de points de contrôle doit conserver une dimension constante en chaque ligne, deux alternatives se présentent alors à nous dans leur égalisation, le retrait d'éléments à un faible coût de calcul ou l'insertion de nouvelles valeurs interpolées pour une meilleure précision à un coût supérieur.

4.1.3 Flammes complexes

En présence de formes plus complexes que la flamme d'une bougie, telles qu'observées sur une torche ou un feu de camp, notre système ne parvient pas à bien restituer les formes aux angles accentués et avec chevauchements. Pour y arriver, nous devons plutôt recourir à plusieurs systèmes dont les mèches virtuelles peuvent se superposer. Bridault[Bri07] présente deux configurations s'appliquant au type de flammes évoquées précédemment (voir figure 4.4). Ceci introduit malheureusement une seconde problématique : le chevauchement de surfaces semi-transparentes. L'affichage adéquat de ces surfaces nécessite un rendu onéreux difficilement applicable aux surfaces NURBS. Nous proposons de contourner cet obstacle à l'aide d'une méthode que nous proposons à la section 4.3.3 sur la transparence.

4.2 Simulation

La forme d'une flamme est insaisissable, présentant une apparence en constante évolution et aux déplacements chaotiques tout en préservant une cohérence temporelle. Un système en mesure

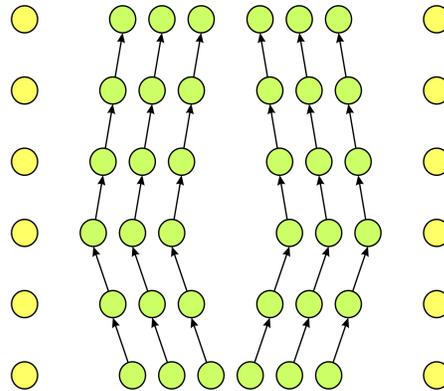


FIGURE 4.5 – Vue latérale d’une configuration de noeuds possible (sans les ressorts). Les flèches représentent les vecteurs de vitesse extraits du système et l’ensemble de leurs directions permettent aux particules de se déplacer en suivant la forme générale d’une flamme de bougie.

de reproduire ces mouvements s’avère nécessaire, et nous proposons d’y arriver à l’aide d’une grille de noeuds liés entre eux par des ressorts afin d’en extraire un champ de vitesse. Le système subit l’influence d’une force régulièrement obtenue aléatoirement à l’intérieur d’un intervalle de valeurs, et résulte en des déplacements pouvant grandement varier tout en conservant une cohésion dans le temps à l’aide de ressorts les mitigeant.

4.2.1 Grille de noeuds

Chaque indice d’un tableau tridimensionnel de taille $X \times Y \times Z$ conserve la référence à un noeud d’une simulation de ressorts. La position d’origine d’un noeud du système est attribuée selon la position du noeud précédent plus un déplacement positif variable, signifiant qu’il n’y a aucun entrecroisement entre chaque pair de noeuds lié en Y . L’emplacement des noeuds influence la forme générale de la flamme puisque le champ de vitesse est extrait des différences de positions entre les noeuds, et les particules se déplacent en conséquence. Pour notre flamme laminaire, nous avons opté pour une forme mouvant initialement les particules vers l’extérieur pour finalement les faire converger vers la pointe à partir du quart des indices du tableau en Y (voir figure 4.5). Une série de noeuds sont ajoutés en périphérie de la simulation de ressorts (jaune sur la figure 4.5) et non comptabilisés lors de l’extraction du champ de vitesse. Ils contribuent à maintenir la forme originale malgré la force verticale constamment appliquée. Pour ancrer la flamme à sa mèche, nous fixons la position des noeuds du premier niveau (indice $y = 0$). Ils deviennent donc exempts de déplacement lors de la mise-à-jour du système. Nous rendons mobiles tous les autres noeuds non périphériques (jaunes). Plusieurs autres configurations peuvent être envisagées afin d’influencer la forme de la flamme vers un aspect désiré.

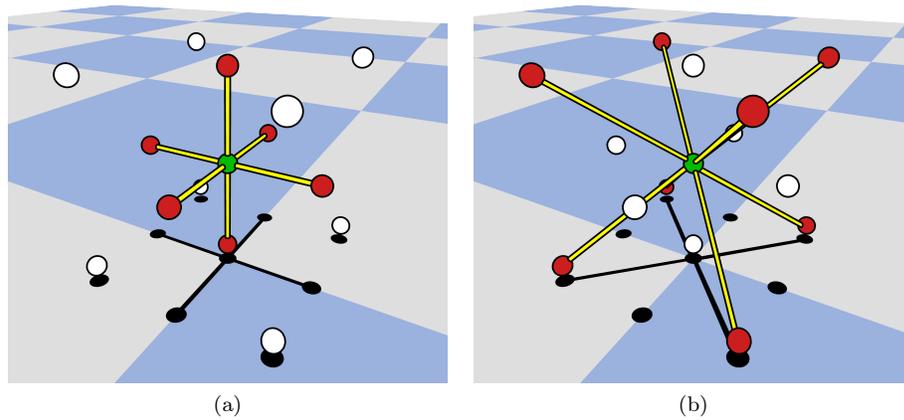


FIGURE 4.6 – Configuration des 14 ressorts liant chaque noeud à ses voisins. Figure (a) présente les liens directs tandis que la figure (b) présente les liens diagonaux.

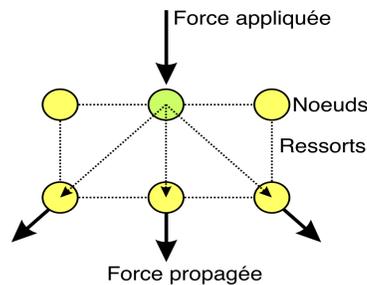


FIGURE 4.7 – Vue latérale de la propagation par ressorts, d'une force reçue sur un noeud à ses voisins immédiats.

Chaque noeud du système est lié par ressorts à ses six voisins immédiats et huit voisins diagonaux, ce qui permet de mitiger l'impact d'une force externe en la propageant à ses voisins et de retrouver sa position initiale en l'absence de cette force. Les liens directs (figure 4.6a) permettent au système de retrouver son état initial lorsqu'au repos, tandis que les liens en diagonale (figure 4.6b) permettent de propager une force reçue aux voisins perpendiculaires. Ainsi, une force de compression verticale appliquée à la simulation génère également un déplacement latéral des noeuds (voir figure 4.7).

4.2.2 Système de ressorts

Notre système est constitué d'un ensemble de noeuds localisés dans un environnement tridimensionnel et liés à leurs voisins immédiats par un ensemble de ressorts. Lorsqu'un noeud est

déplacé, le système de ressorts applique une force interne restaurant progressivement sa position initiale. Il est également influencé par un ensemble de forces externes modélisant la convection des gaz d'une flamme et sa réaction lors de son déplacement. À chaque incrément du pas de temps de la simulation, c'est-à-dire la discrétisation de la simulation par une évaluation à intervalle régulier, le système évalue les forces externes et internes appliquées sur les noeuds pour résoudre leurs positions à cet instant. Issue du domaine de la dynamique moléculaire, la méthode d'intégration de Verlet [HL97] permet d'évaluer l'évolution de ce système de façon stable et à moindre coût en requérant uniquement de conserver les positions courante et précédente d'un noeud plutôt que sa position et sa vitesse. Contrairement aux méthodes traditionnelles, les résultats demeurent stables sans imposer un pas de temps réduit, permettant une évaluation plus occasionnelle. Comme la vitesse est calculée implicitement, sa valeur est toujours conséquente aux déplacements des noeuds.

$$Pos_{t+1} = Pos_t + (Pos_t - Pos_{t-1}) \times atténuation + \overrightarrow{force} \times \Delta t^2 \quad (4.4)$$

4.2.3 Forces externes

Nous reproduisons trois forces externes pouvant influencer la forme de la flamme, soient la convection des gaz, le déplacement de la flamme et le souffle du vent. La convection des gaz est simulée à l'aide d'un vecteur de force appliqué à chaque noeud mobile du système. Ce vecteur de force est essentiellement orienté vers le haut avec sa direction et sa magnitude exactes aléatoirement attribuées à intervalle régulier à l'intérieur d'une plage de valeurs configurables. Ceci permet d'ajuster le niveau d'activité de la flamme et d'obtenir ce mouvement hypnotique acyclique qui lui est propre. Pour les forces issues du souffle du vent et du mouvement d'une flamme, nous calculons un vecteur de déplacement subséquent appliqué aux noeuds mobiles de la simulation. En supplément, comme un système de ressorts n'exhibe pas une réaction aux forces avec la même célérité que les gaz d'une flamme, nous appliquons ce même vecteur pondéré directement aux particules modélisant la flamme. La longueur du vecteur est conservée sous un seuil configurable afin de prévenir une flamme excessivement allongée en présence d'un déplacement trop rapide ou d'un vent trop intense.

4.2.4 Champ de vitesse

L'ultime objectif du système de ressorts est l'extraction d'un champ de vitesse contribuant au déplacement des particules modélisant une flamme. La direction principale du déplacement

des gaz d'une flamme de diffusion étant vers le haut, nous pouvons simplement établir les vecteurs du champ comme la différence de position de chaque noeud à son voisin d'indice supérieur (voir flèches sur figure 4.5). Notre champ de vélocité possède ainsi une dimension de $(X - 2) \times (Y - 1) \times (Z - 2)$ où $X \times Y \times Z$ est la dimension de notre tableau de noeuds (incluant les noeuds périphériques).

4.3 Apparence

La section précédente nous a introduit à une technique synthétisant la forme d'une flamme en constante évolution, mais simplement afficher une couleur unie ne suffit pas à concrétiser l'illusion de son authenticité. Pour y arriver, nous devons adresser deux de ses particularités, sa texture aux couleurs dégradées et son émission de lumière venant affecter notre perception.

4.3.1 Placage de texture

Cette technique fondamentale du domaine de l'infographie consiste à l'application d'images sur la géométrie rendue en se conformant à la perspective. Dans le cas de notre flamme laminaire, nous pouvons imaginer une forme cylindrique sur laquelle nous enroulons la capture photographique d'une flamme. Elle permet de simuler une complexité des surfaces plus détaillée qu'actuellement présente, puisqu'il serait excessivement onéreux de modéliser et rendre chaque petit détail. Ainsi, un artiste associe une texture de bois sur le dessus d'une table, plutôt que de péniblement modéliser le grain lui-même. La technique est considérablement optimisée sur toute carte graphique moderne et peut aisément être déployée par des artistes et des programmeurs.

Une flamme laminaire présente trois zones visuelles distinctes qu'il incombe de bien reproduire pour une apparence convaincante. Elles consistent en un noyau bleuté à sa base, une zone blanche de haute intensité lumineuse en son centre et la zone de contour et de pointe tendant vers le rouge (voir figure 2.2). Une image d'une flamme apposée naïvement à la modélisation n'est malheureusement pas en mesure d'adéquatement restituer son apparence tridimensionnelle puisqu'elle présente un point de vue unique et force la flamme synthétisée à revêtir le profil capturé. Nous proposons de générer une image exempte de ces défauts en modifiant une capture de flamme à l'aide d'un logiciel de traitement d'images, tel que présenté à la figure 4.8.

Le placage de textures sur géométrie requiert la capacité de spécifier quelle partie de l'image afficher à quel endroit, ce que les API graphiques offrent par l'entremise d'un système de coordonnées UV conjugué à une matrice de transformation. Ces coordonnées consistent en un couple

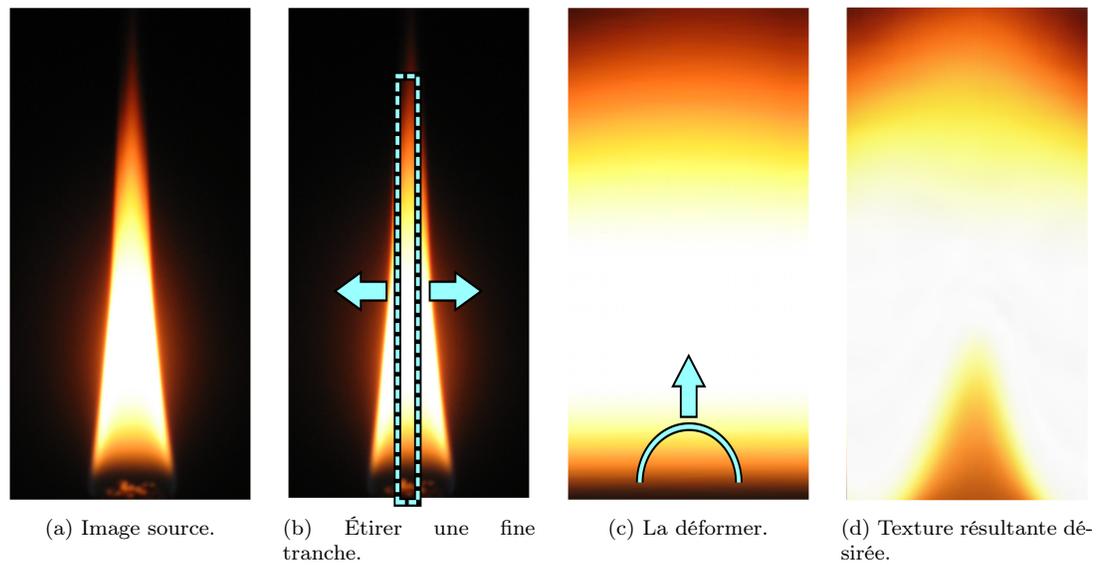


FIGURE 4.8 – Étapes de la création d'une texture de flamme, que l'on peut appliquer sur notre surface NURBS de type cylindrique.

de valeurs normalisées par les dimensions de la texture et permettent alors d'adresser les texels (*texture element*) sans préalablement connaître les dimensions de l'image. La valeur $[0.5, 0.5]$ spécifie ainsi le centre d'une image et plusieurs stratégies existent gérant des coordonnées à l'extérieur de la plage de valeurs $[0, 1]$. Appliquer une texture sur la géométrie consiste alors à assigner à un modèle, des coordonnées UV à chacun des sommets qui seront subséquemment modifiées par la matrice de transformation texture lors du rendu. Considérant que cette dernière possède habituellement les valeurs d'une matrice identité, la transformation est usuellement sans effet.

Une surface NURBS ne permet pas de spécifier directement les différentes propriétés de ses sommets (couleur, position, coordonnées UV) puisqu'ils sont générés dynamiquement. Nous devons plutôt les assigner aux points de contrôle de la surface. Pour les coordonnées V indexant la hauteur de la texture, nous utilisons le ratio $\frac{\text{Indice colonne}}{\text{Nombre colonnes}}$ du tableau de points de contrôle, où chaque colonne d'une ligne réfère aux particules d'une même chaîne. Pour les coordonnées U indexant la largeur de la texture, nous calculons l'angle normalisé entre la position de l'émetteur d'une particule, à l'axe des X (voir figure 4.9). Nous pouvons également concevoir cette valeur comme l'angle normalisé en coordonnées cylindriques de l'émetteur au centre de notre analogie de cylindre. Ceci fixe l'image de la flamme à la géométrie présente tout en occasionnant un désagrément visuel, les coordonnées n'étant pas spécifiées relatives à la position de l'observateur, la base bleutée en forme de cloche ne sera pas centrée sur la flamme en tout temps. Nous rectifions

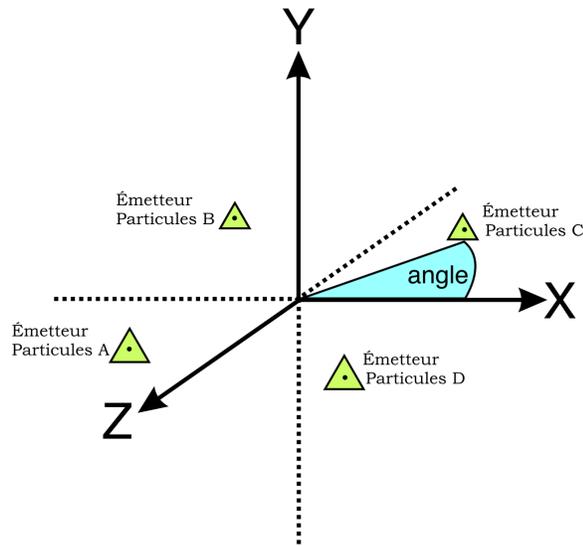


FIGURE 4.9 – Extraction de la coordonnée U à partir de l’angle horizontal d’un émetteur à l’axe des X . Les quatre émetteurs présents sur cette image, se trouvent sur le plan formé par les axes X et Y .

ce problème par l’ajout d’une translation horizontale à la matrice de transformation de texture, nous assurant que l’image demeure centrée au point de vue. Si nous reprenons l’analogie de notre flamme cylindrique, cette étape consiste glisser la photo de la flamme autour du cylindre, jusqu’à ce qu’elle soit centrée sur notre regard. Nous pouvons alternativement appliquer cette valeur directement aux coordonnées U de chaque point de contrôle.

4.3.2 Effet d’éblouissement

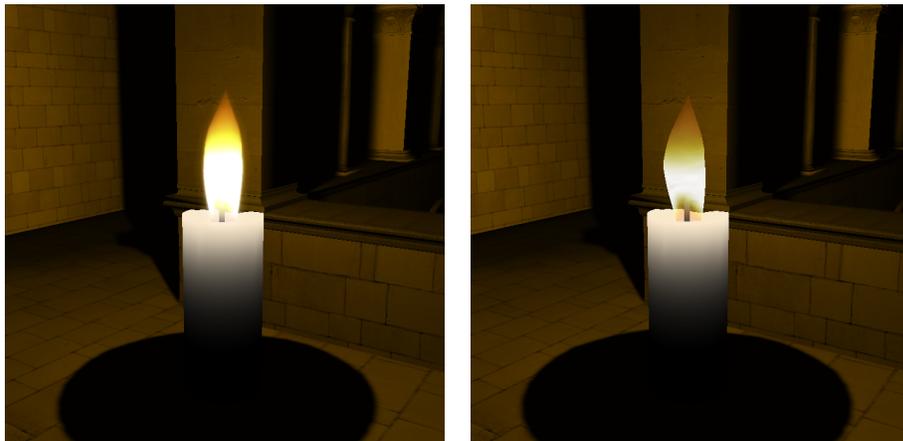
Un environnement présentant un contraste de luminosité élevé est plus difficilement observable dans tous ses détails par l’oeil humain ou une caméra. Cette occurrence s’observe à la frontière d’une pièce sombre et d’un environnement extérieur éclairé par un soleil radieux. Les récepteurs se saturent alors temporairement, expliquant l’éblouissement donnant à la scène une apparence d’amalgame blanc indistinct. Les technologies actuelles couramment utilisées n’étant pas en mesure de produire une intensité suffisante induisant ce phénomène, une technique de post-traitement dénommée effet d’éblouissement (ou *bloom*), a été développée à cet escient, vu son importance à notre perception du réalisme d’une image. Cette technique est couramment utilisée dans le domaine du jeu vidéo et Microsoft offre un exemple de sa mise en oeuvre¹.

1. <http://creators.xna.com/en-US/sample/bloom>



(a) Centre de la texture centrée sur la flamme avec le point de vue. (b) Centre de la texture non alignée sur le point de vue.

FIGURE 4.10 – Flamme (a) avec translation de texture et (b) sans.



(a)

(b)

FIGURE 4.11 – Flamme (a) avec effet d'éblouissement et (b) sans.

Nous pondérons la force de l'éblouissement selon l'intensité de la luminescence de notre flamme, dont le calcul est présenté à la section 4.4.1 sur l'illumination. La variation de l'éblouissement dans le temps, accorde à notre flamme une apparence plus organique tout en atténuant l'aspect synthétique de notre texture, tel que présenté sur la figure 4.11, bien que pour être pleinement appréciée, une séquence vidéo est requise.

4.3.3 Transparence

Le rendu de géométries semi-transparentes superposées nécessite un affichage ordonné (décroissant ou croissant) par leur distance à la caméra, permettant d'assurer que les couleurs seront adéquatement composées à l'arrière-plan. Cette étape suit habituellement le rendu stan-

dard de la géométrie opaque, simplifiant la gestion de la transparence et diminuant le coût de la surévaluation d'un pixel lorsqu'obstrué par un pixel opaque. En présence de plusieurs modèles semi-transparents sans chevauchement, un tri par objet est suffisant, ce qui est le cas d'une flamme laminaire convexe. Malheureusement, cette caractéristique est absente des flammes aux formes complexes, nécessitant une approche plus pointue adressant le problème introduit.

Dans les langages de programmation pour carte graphique ou *shading languages*, un fragment se définit comme un pixel en cours d'évaluation. Suite à la résolution de sa couleur, des tests déterminent s'il doit être conservé (profondeur, niveau de transparence) et l'image finale intégrera ce nouvel élément selon une fonction de composition (remplacer, ajouter, multiplier, mélanger, ...). Un tri de tous ces fragments ou même des triangles de la géométrie s'avère très onéreux, écartant la possibilité de résultats précis en temps réel. D'autres alternatives ont été proposées, telle que le *depth peeling*, mais les nombreuses passes de rendu exigées les rendent plutôt onéreuse pour une exécution en temps réel. Nous proposons de résoudre la problématique en s'inspirant de la technique de moyenne pondérée proposée par Bavoil et Myers[BM08]. Leur méthode permet d'estimer le résultat en seulement deux passes de rendu. Une première étape évalue la couleur RGB de chaque fragment, pondérée par sa valeur alpha, pour finalement conserver le résultat dans une destination temporaire. Une seconde étape trouve la valeur RGB moyenne à partir de la destination temporaire, en divisant la valeur RGB accumulée de chaque fragment par la somme de ses valeurs alpha pour finalement mélanger cette valeur normalement à l'image finale. Cette technique estime également la valeur alpha, finale comme la moyenne de toutes les valeurs alpha en les additionnant puis divisant cette somme par le nombre de fragments rendus à ce pixel. Dans le cas d'une flamme avec interchevauchements, nous avons remarqué que cette technique produit une démarcation prononcée à la frontière où le nombre de chevauchements varie. La valeur alpha variant trop abruptement, nous proposons plutôt de retenir la valeur alpha maximum d'un fragment, à chaque pixel. L'algorithme des étapes est présenté à la procédure 2. Bien que les résultats ne sont pas exacts, cette méthode permet d'obtenir une bonne apparence à moindre coût (voir figure 4.12).

4.4 Impact sur l'environnement

Un facteur important de la fascination induite par le feu s'explique par l'impact aux diverses nuances qu'il introduit à la perception de notre environnement. Une pièce sombre illuminée à la lueur d'une chandelle acquiert instantanément une ambiance chaleureuse, les aléas de la

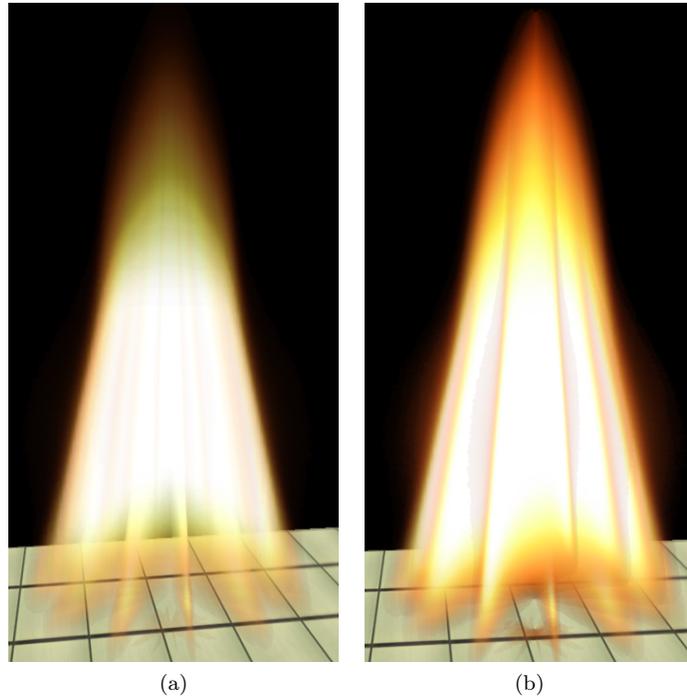


FIGURE 4.12 – Chevauchement de plans semi-transparents (a) avec et (b) sans gestion de la transparence.

flamme suscitant des mystérieuses ombres dansantes, et le cercle de lumière devenant un îlot de sécurité face à l'inconnu. L'illumination conjuguée à l'apparition d'ombres qu'elle induit sont deux impacts aisément observables et les négliger s'avèrerait une lacune cruciale à notre perception du réalisme de la scène (voir figure 4.13).

4.4.1 Illumination

Les caractéristiques principales de l'illumination de flammes sont sa nature anisotrope de portée limitée et fluctuant d'un instant à l'autre. La variation temporelle de l'intensité s'explique par la quantité irrégulière de gaz oxydé et se caractérisant par le volume de la flamme. Nous proposons de lier cette valeur à la tension moyenne de la simulation de ressorts. Elle est un bon indicateur puisqu'un système de haute tension signifie habituellement qu'il est en extension, accroissant ainsi la taille de la flamme par le déplacement accéléré des particules. Une intensité maximale configurable est spécifiée par une valeur RGB et modulée lors du rendu par le ratio de la tension moyenne courante sur la tension moyenne maximale détectée depuis le début de la simulation.

L'analyse photométrique d'une flamme laminaire de Bridault [Bri07], lui a permis d'observer l'anisotropie verticale d'une flamme en mesurant l'intensité de plusieurs points, tout en émettant

Procédure 2 Gestion de la transparence

1. Rendre les éléments opaques
 2. Initialiser la fonction de mélange des *render targets* à :
$$\begin{cases} RGB_{dst} = RGB_{src} + RGB_{dst} \\ \alpha_{dst} = \max(\alpha_{src}, \alpha_{dst}) \end{cases}$$
 3. Assigner *texture1* et *texture2* aux *render targets* respectifs
 4. Désactiver le test de profondeur (*depth test*)
 5. Rendre les éléments semi-transparents
 - $texture1.rgb = input.rgb \times input.alpha$ (somme des couleurs pondérée par son alpha)
 - $texture1.alpha = input.alpha$ (alpha maximum)
 - $texture2.r = input.alpha$ (somme des alphas)
 6. Initialiser la fonction de mélange des *render targets* à : $RGB_{dst} = RGB_{src} \times \alpha_{src} + RGB_{dst} \times (1 - \alpha_{src})$
 7. Assigner le tampon écran au *render target* principal
 8. Assigner *texture1* et *texture2* en entrée
 9. Rendre un quadrilatère plein écran avec les deux textures
 - $result.rgb = \frac{texture1.rgb}{texture2.alpha}$
 - $result.alpha = texture2.r$
-

le constat qu'il est raisonnable de supposer qu'elle demeure sensiblement constante horizontalement sur la flamme d'une bougie. Il s'ensuit qu'une source ponctuelle de lumière standard n'est pas apte à adéquatement reproduire ce comportement et Bridault propose donc d'utiliser une texture photométrique angulaire approximant les résultats. Ne disposant pas de l'équipement nécessaire pour un taux d'échantillonnage élevé, il pare à cette lacune en interpolant les valeurs à l'aide d'une méthode d'arc de spirale nécessitant quatre accès textures par fragment. En évaluant l'angle d'inclinaison générale de la flamme, il applique une rotation à la matrice de transformation texture afin de modifier la distribution d'intensité en fonction de son orientation (voir les figures 4.14a et 4.14b).

Nous proposons d'alléger le coût de calcul de l'évaluation de l'intensité en estimant sa valeur à partir du cosinus entre le vecteur d'inclinaison de la flamme et celui de la différence de positions (voir figure 4.15). Bien que différent de la réalité capturée par Bridault, cette méthode permet de bien retransmettre le dynamisme de l'éclairage à un coût raisonnable, puisque la valeur du cosinus s'obtient simplement par le produit scalaire des deux vecteurs normalisés (voir les figures 4.14c et 4.14d). Au final, l'illumination d'un fragment est déterminée par la valeur RGB de la source lumineuse, modulée par l'intensité de la flamme, l'atténuation due à la distance, le terme anisotropique et le coefficient lambertien de la surface éclairée. Le terme spéculaire a été omis puisqu'aucune de nos scènes ne le nécessitait, mais il peut aisément être intégré à l'équation 4.6. Le résultat est présenté à la figure 4.16 mais un visionnement de la flamme animée est nécessaire pour une pleine appréciation de la dynamique restituée. L'équation 4.5 décrit ce



(a) Rendu avec ombres et avec lumière.



(b) Rendu sans ombre et avec lumière.



(c) Rendu avec ombres et sans lumière.



(d) Rendu sans ombre et sans lumière.

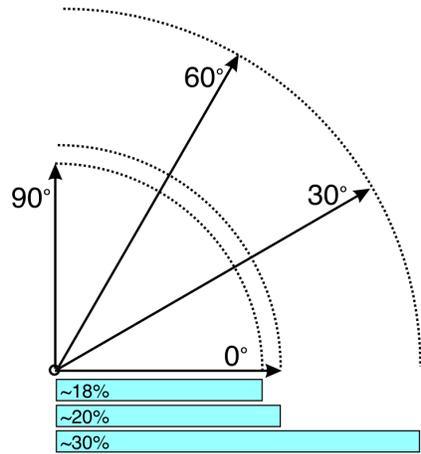
FIGURE 4.13 – Impacts visuels des ombres et de l'illumination sur le réalisme d'un environnement de synthèse.

calcul plus en détail et la figure 4.15 illustre les composantes du calcul de la valeur de l'angle θ .

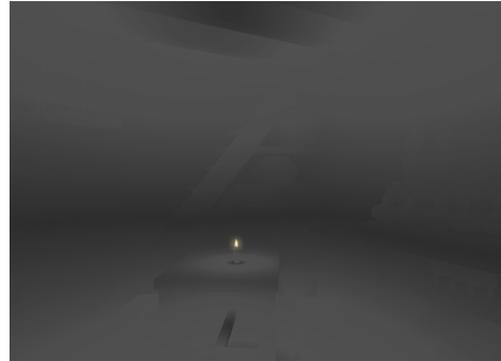
$$Aniso(\theta) = \begin{cases} I_{min} + (I_{max} - I_{min}) \times \left(\frac{|\cos \theta|}{\cos 45} \right) & |\cos \theta| \leq \cos 45 \\ I_{min} + (I_{max} - I_{min}) \times \left(\frac{1.0 - |\cos \theta|}{1.0 - \cos 45} \right) & |\cos \theta| > \cos 45 \end{cases} \quad (4.5)$$

$$\text{où } \begin{cases} I_{min} = \text{Intensité minimum anisotropie} & (\text{Fixé à } 0.15 \text{ dans notre implémentation}) \\ I_{max} = \text{Intensité maximum anisotropie} & (\text{Fixé à } 1 \text{ dans notre implémentation}) \end{cases}$$

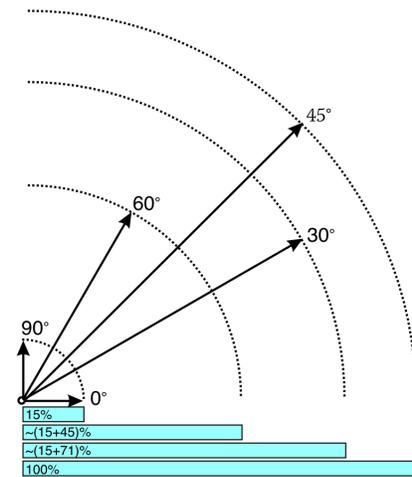
$$Lumière(\alpha, \theta, \sigma) = RGB_{Ambient} + RGB_{Diffus} \times Aniso(\theta) \times Lambert(\sigma) \times Atténue(\alpha) \quad (4.6)$$



(a) Valeurs des mesures spectrométriques d'une bougie selon les angles zénithaux, prise par Bridault [Bri07].



(b) Volume photométrique généré par Bridault. Source : [Bri07].



(c) Pourcentage de l'intensité lumineuse appliquée selon l'angle zénithal.



(d) Volume photométrique généré.

FIGURE 4.14 – Illumination anisotrope d'une scène.

$$\text{où } \begin{cases} \alpha = \text{Distance entre fragment et lumière} \\ \theta = \text{Angle entre inclinaison flamme et direction fragment} \\ \sigma = \text{Angle entre fragment et lumière} \end{cases}$$

4.4.2 Ombres

La génération d'ombres dans une scène représente un vaste domaine de recherche en soi et nécessite un pourcentage substantiel du temps de rendu de chaque trame. Notre source de lumière étant omnidirectionnelle, elle requiert une coûteuse évaluation à tous azimuts, contrairement aux

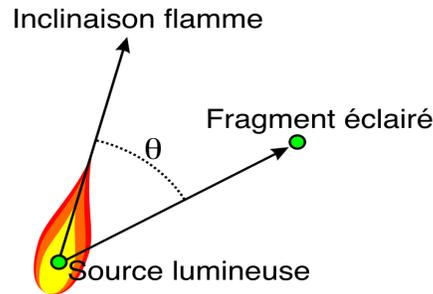


FIGURE 4.15 – Angle entre l’inclinaison de la flamme et la différence de position du fragment.

sources directionnelles usuellement utilisées dans les simulations temps réel. Nous avons opté de combiner une méthode de génération d’ombres dites *variance shadow mapping* (voir l’exemple d’implémentation par Donnelly et al. [DL06]) à une méthode d’accès texture de type *cubemap* pour l’aspect omnidirectionnel.

La partie *shadowmap* consiste à rendre préalablement la scène du point de vue de la source lumineuse, de conserver les profondeurs recueillies dans une texture, puis de déterminer lors du rendu normal si le fragment courant est bien l’élément le plus près de la lumière. Si la texture *shadowmap* contient une valeur de distance inférieure à la distance de la lumière au fragment courant, nous concluons que le rayon de lumière émanant de la source a été bloqué avant d’atteindre le fragment et que nous sommes donc dans l’ombre (voir figure 4.17). Sans l’ajout d’une passe de rendu additionnelle, la technique de *shadowmap* élémentaire ne permet pas de lissage de l’ombre, introduisant des artefacts de discrétisation que la méthode de *variance shadowmap* s’occupe de rectifier en modifiant la notion binaire d’être ou non à une distance inférieure, par un taux de probabilité. La figure 4.18 présente son impact sur la qualité du résultat.

La partie *cubemap* de la technique permet de synthétiser les ombres d’une source lumineuse omnidirectionnelle en rendant six scènes orthogonales alignées sur chacun des axes tridimensionnels. Nous pouvons imaginer une caméra en forme de cube, possédant six pellicules et capturant l’environnement visible sur chacune de ses faces. Lorsque l’on désire extraire la valeur d’une certaine direction, nous retenons la texture issue de la face du cube de sa direction prépondérante, tandis que les coordonnées *UV* proviennent des deux coordonnées normalisées restantes. Par exemple, la direction $(2, 4, 8)$ est normalisée à $(0.22, 0.44, 0.87)$, permettant de déterminer que nous voulons la valeur aux coordonnées $(0.22, 0.44)$ de la texture alignée sur l’axe Z^+ (voir figure 4.19).

Le sujet principal de ce mémoire n'étant pas le rendu des ombres, des travaux futurs bénéficieraient de s'y attarder davantage afin de réduire le coût de cette étape. Certaines pistes de solution peuvent s'orienter vers l'inclusion d'une technique de division de l'espace supprimant tout objet en dehors du champ de vision, et l'utilisation de niveaux de détails pour la géométrie.



(a)



(b)

FIGURE 4.16 – Scène (a) avec et (b) sans volume photométrique appliqué lors de l'illumination.

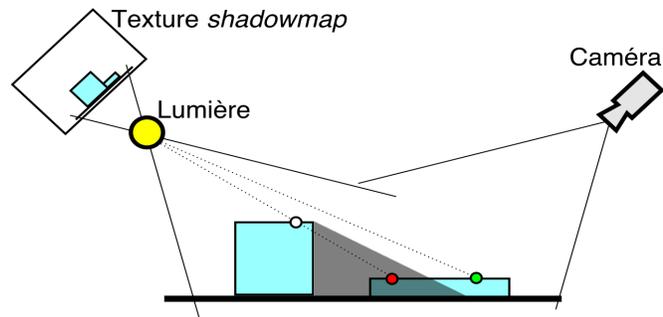


FIGURE 4.17 – Principe de la technique de *shadowmap*. Le point rouge est dans l'ombre puisque la texture *shadowmap* générée à partir du point de vue de la lumière contient un point plus près (blanc). A l'opposé, le point vert reçoit des rayons de la source lumineuse puisqu'aucun élément obstruant le point n'est détecté, la texture contenant les informations du même point vert.

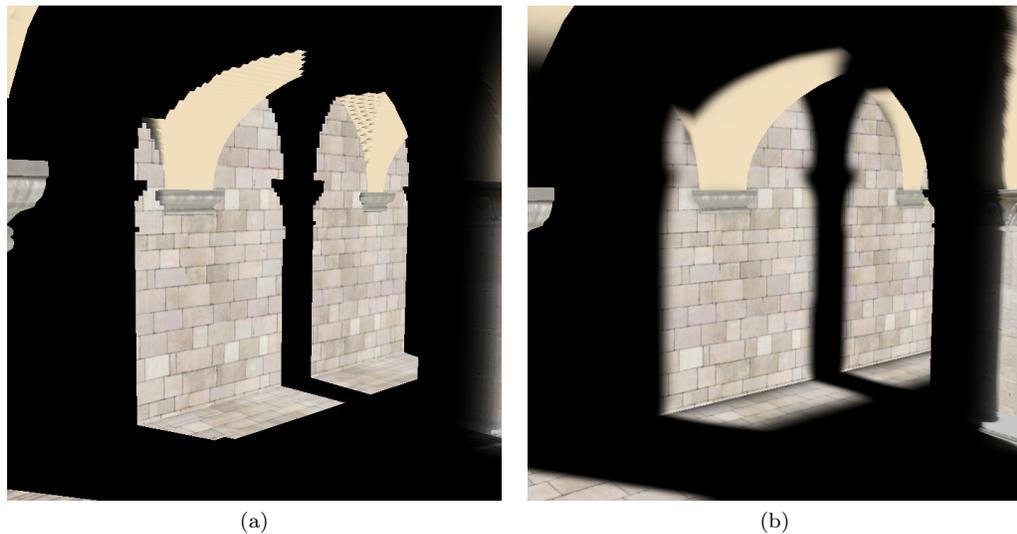


FIGURE 4.18 – Ombres créées à partir (a) d'un *shadowmap* standard et (b) d'un *variance shadowmap*. La seconde méthode présente des contours flous restituant une apparence plus naturelle que la méthode de base exhibant des artefacts de discrétisation ainsi que des démarcations découpées.

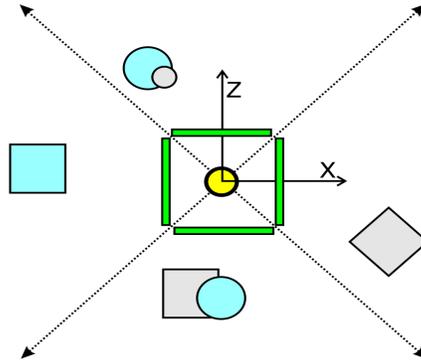


FIGURE 4.19 – Vue verticale du rendu de quatre des six faces d'un *cubemap*. La source lumineuse omnidirectionnelle est représentée par le cercle jaune, tandis que les rectangles verts représentent les faces du cube recevant chacun des rendus alignés sur les axes X^+ , X^- , Z^+ , Z^- .

Chapitre 5

Résultats

[...]
Hungry people don't stay hungry for long
They get hope from fire and smoke as the wheat grows strong
Hungry people don't stay hungry for long
They get hope from fire and smoke as they reach for the dawn
-RAGE AGAINST THE MACHINE (NEW MILLENIUM HOME)

La synthèse en temps réel d'une flamme laminaire réaliste à un coût minime, telle qu'énoncée comme objectif à la section 1.2, a été largement atteinte, comme en font foi les séquences vidéo disponibles en ligne¹ ainsi que les figures 5.1, 5.2 et 5.3. Nous constatons que le retrait de la flamme sur les scènes de Sponza² et Sibenik³, n'affecte pas leurs performances, puisque la majorité du coût se situe dans le rendu des ombres et de la scène non optimisée. Nous donnons lieu à une simulation ne nécessitant que 0.43ms (CPU) et 0.44ms (GPU) pour le rendu de la flamme, soit respectivement 2.58% et 2.64% d'une trame temps réel, à raison de 60 images/s. Plusieurs pondérâtes viennent influencer les performances du système complet, telle la non-exclusivité des ressources dans un système d'exploitation tel que OSX et Windows, mais au final, le rendu complet d'une scène minimaliste consistant en un plancher, une chandelle et une flamme, ne requiert que 0.94ms (1063 images/s) sans ombres et ni étapes de post traitement. La même scène avec ombres et post traitement résulte en un temps de rendu de 6.93ms (144 images/s). Le reste de cette section analysera les performances sous différentes configurations. Les mesures ont été acquises sur une flamme stable soumise à une force ascendante constante et moyennées

1. <http://www.iro.umontreal.ca/labs/infographie/theses/fatnasss/>

2. <http://hdri.cgtechniques.com/~sponza/files/>

3. <http://hdri.cgtechniques.com/~sibenik2/>

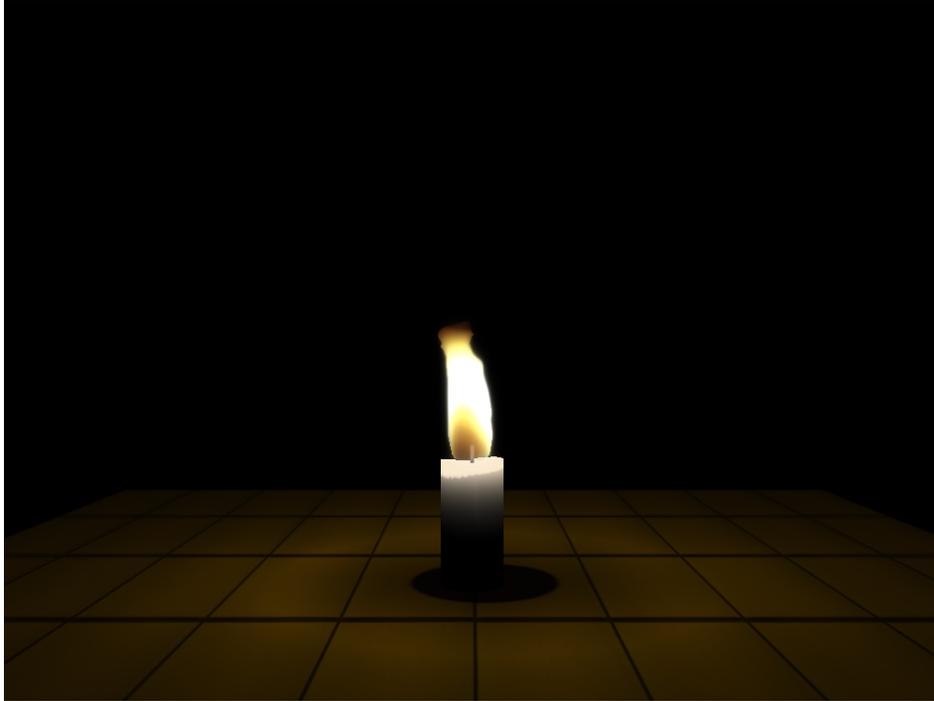


FIGURE 5.1 – Scène de base de 4653 triangles, utilisée pour les tests avec une flamme de 175 particules. Performances : 137 images/s (avec éblouissement, flamme et ombres), 143 images/s (avec éblouissement et ombres) et 294 images/s (avec éblouissement et flamme).

sur 30 trames. Il est à noter que les temps écoulés entre chaque début de trame, fluctuent entre eux malgré l'équivalence des tâches, notre programme n'ayant pas l'exclusivité des ressources et doivent ainsi être considérés que comme une approximation surévaluant le coût réel. Sauf sous avis contraire, les mesures ont été prises sur une scène simple de 4652 triangles, pour une unique flamme de 75 particules modélisée par une surface NURBS de type DD2 (voir section 5.1) et sans ombres ou effets d'éblouissement. Tous les tests ont été effectués sur un ordinateur portable (offrant donc des performances 3D plus modestes), de la configuration suivante :

Modèle : Portable MacBook Pro

Cpu : Intel Core 2 Duo 2.4 GHz

Ram : 2 Go de RAM

Carte graphique : NVidia GeForce 8600M GT 256Mo RAM

OS : OSX 10.5

Résolution : 1024 × 768

Compilateur : Gcc 4.0.1 sous QT 4.6.2 avec OpenGL 2.0 et *GL shading language* 1.20



FIGURE 5.2 – Scène Sponza de 193990 triangles avec une flamme de 175 particules. Performances : 55 images/s (avec éblouissement, flamme et ombres), 55 images/s (avec éblouissement et ombres) et 108 images/s (avec éblouissement et flamme).

5.1 Paramétrisation NURBS

L’affichage d’une surface NURBS requiert une discrétisation de la surface en une liste de triangles qu’une librairie OpenGL réalise lors du rendu. La paramétrisation de cette étape influe grandement sur l’apparence de la flamme et des performances de sa génération. Afin d’adéquatement juger son impact, nous avons examiné quatre niveaux de précision associés à trois types de discrétisation, entraînant douze combinaisons différentes. Ces trois types de discrétisation sont décrits par la documentation de OpenGL comme :

GLU_DOMAIN_DISTANCE (DD) : Discrétise en spécifiant le nombre de points à prendre par unité de coordonnées paramétriques. Nous avons observé les résultats avec 1 (DD1), 2 (DD2), 4 (DD4) et 8 (DD8) points.

GLU_PATH_LENGTH (PL) : Discrétise en spécifiant la longueur maximale de chaque arête de la géométrie générée (en pixels). Nous avons observé les résultats avec une longueur de 40 (PL40), 30 (PL30), 20 (L20) et 10 (PL10) pixels. Cette mesure étant dans l’espace projection, le nombre de triangles générés varie selon la taille de la flamme à l’écran, ce qui permet d’automatiquement ajuster la précision en fonction de sa distance à la caméra. Une seconde variante permet de spécifier cette taille dans l’espace objet, conservant ainsi



FIGURE 5.3 – Scène Sibenik de 238909 triangles avec une flamme de 175 particules. Performances : 45 images/s (avec éblouissement, flamme et ombres), 45 images/s (avec éblouissement et ombres) et 87 images/s (avec éblouissement et flamme).

un nombre de triangles constant.

GLU_PARAMETRIC_ERROR (PE) : Discrétise en spécifiant la distance maximale de chaque point à sa position réelle sur la surface (en pixels). Nous avons observé les résultats avec une précision de 40 (PE40), 20 (PE20), 10 (PE10) et 5 (PE5) pixels. Tout comme `GLU_PATH_LENGTH`, la mesure s’effectue dans l’espace projection avec l’avantage associé, et une variante utilisant l’espace objet existe.

Outre leurs valeurs décroissantes, la précision entre chaque type de discrétisation n’est pas corrélée dans nos tests. La figure 5.4 présente l’apparence du maillage d’une flamme de 40 particules pour chaque type de discrétisation. Nous observons sur la figure 5.5 que la méthode `GLU_DOMAIN_DISTANCE` (DD) présente habituellement un coût de calcul inférieur et qu’il croît linéairement avec son indice de précision (nombre de points par unité). Nous constatons sur la figure 5.6 et la figure 5.7 que les performances des trois méthodes se dégradent linéairement en fonction du nombre de particules. L’impact est uniformément réparti sur le CPU et GPU avec la méthode `GLU_DOMAIN_DISTANCE`, contrairement aux méthodes `GLU_PATH_LENGTH` et `GLU_PARAMETRIC_ERROR` où le CPU accuse la majorité de l’augmentation. Au final, nous avons opté pour l’utilisation de la méthode `GLU_DOMAIN_DISTANCE` conjuguée à une

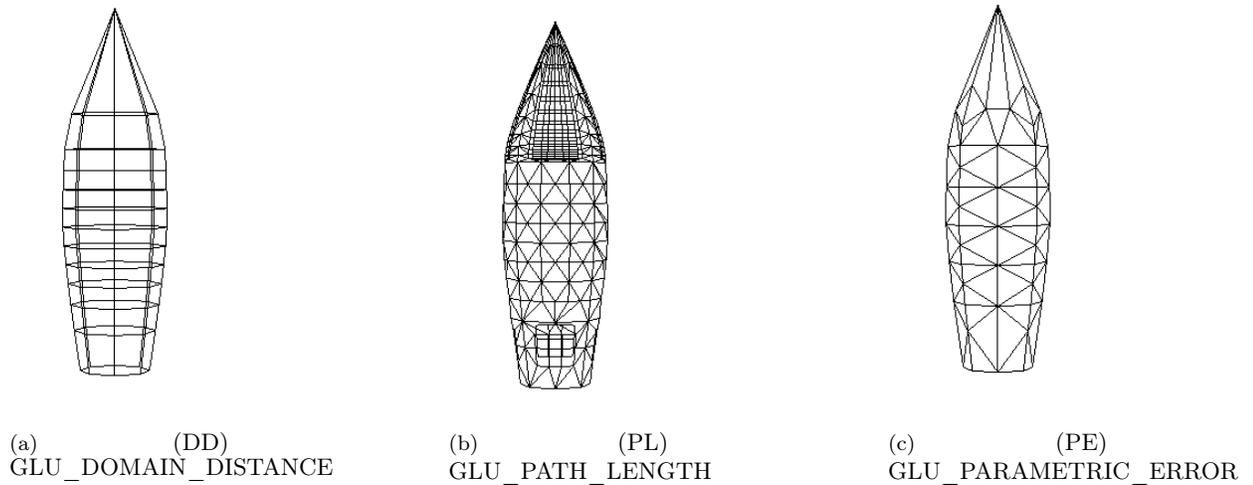


FIGURE 5.4 – Maillage selon différents algorithmes de discrétisation de surface NURBS.

précision de 2 points par unité. Malgré tout, nous notons sur la figure 5.9 que le principal coût du système, demeure la création de la surface NURBS.

5.2 Simulation de ressorts

Nous avons constaté qu'une grille de vitesse de taille $6 \times 6 \times 6$ suffisait amplement à restituer les mouvements aléatoires d'une flamme de façon réaliste et à moindre coût. La mise à jour de ce système requiert l'évaluation de 448 noeuds et 2344 ressorts (voir équation 5.1 et la section 4.2.1) ne nécessitant que 0.09ms, soit 0.54% d'une trame, à raison de 60 images/s. La figure 5.10 présente le temps d'évaluation de différentes tailles de système, soient des grilles de tailles 5, 6, 7, 8, 9, 10 et 15 respectivement.

$$\text{Grille de } 6 \times 6 \times 6 \Rightarrow (6 + 2) \times (6 + 2) \times (6 + 1) \Rightarrow 448 \text{ noeuds} \quad (5.1)$$

5.3 Nombre de flammes

Nous avons démontré que la simulation d'une flamme unique pouvait s'accomplir à un coût raisonnable et affectant minimalement les performances du système. La figure 5.11 confirme que cette affirmation se maintient en présence d'un grand nombre de flammes, la simulation pouvant maintenir un taux de 60 images/s avec 20 flammes, et plus de 30 images/s avec 50

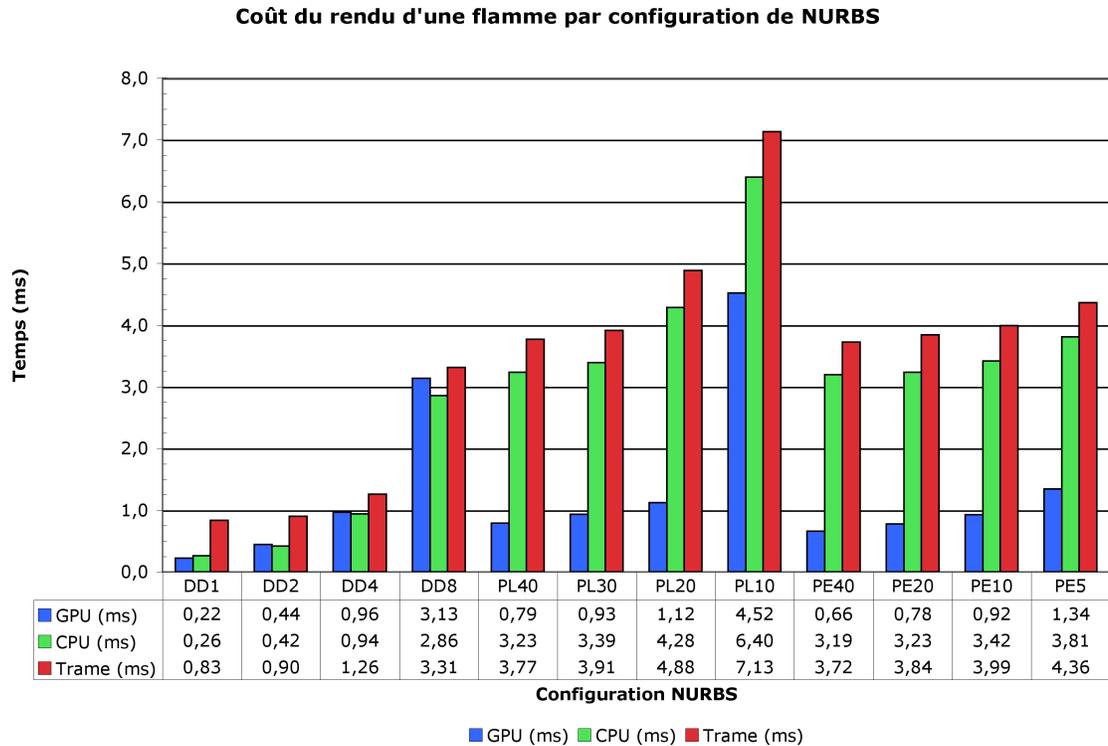


FIGURE 5.5 – Coût du rendu d'une flamme, par configuration de surface NURBS.

flammes. Aux environs de 10 flammes, le coût de rendu présente un comportement plutôt linéaire d'approximativement 0.49ms par nouvel ajout, le coût initial d'une trame et le temps de calcul des effets étant résorbé, tel que l'on peut remarquer sur la figure 5.13.

5.4 Performances avec les effets

Le rendu d'ombres et l'effet d'éblouissement nécessitent un temps de calcul substantiel. Le temps d'évaluation des ombres est tributaire à la résolution de la texture utilisée pour notre *shadow mapping*, la quantité de lissage appliquée, la complexité de la géométrie pouvant obstruer la lumière et le nombre de sources lumineuses. Pour ce qui est de l'effet d'éblouissement, il possède un coût lié à la résolution du rendu et la quantité de lissage appliquée. Ceci signifie que notre rendu de 1024×768 utilisant des textures d'ombres de 512×512 dans une scène de 4652 triangles, présente un coût fixe d'effets spéciaux, sans lien avec la quantité de flammes rendues ou leur complexité, en supposant la présence d'une source lumineuse unique. La figure 5.12 présente le coût de rendu CPU et le temps écoulé entre chaque trame, pour différentes combinaisons de nombre de flammes et d'effets. Nous pouvons clairement constater sur la figure 5.13 que le coût initial des ombres et de l'éblouissement se résorbe lorsque la quantité de flammes augmente

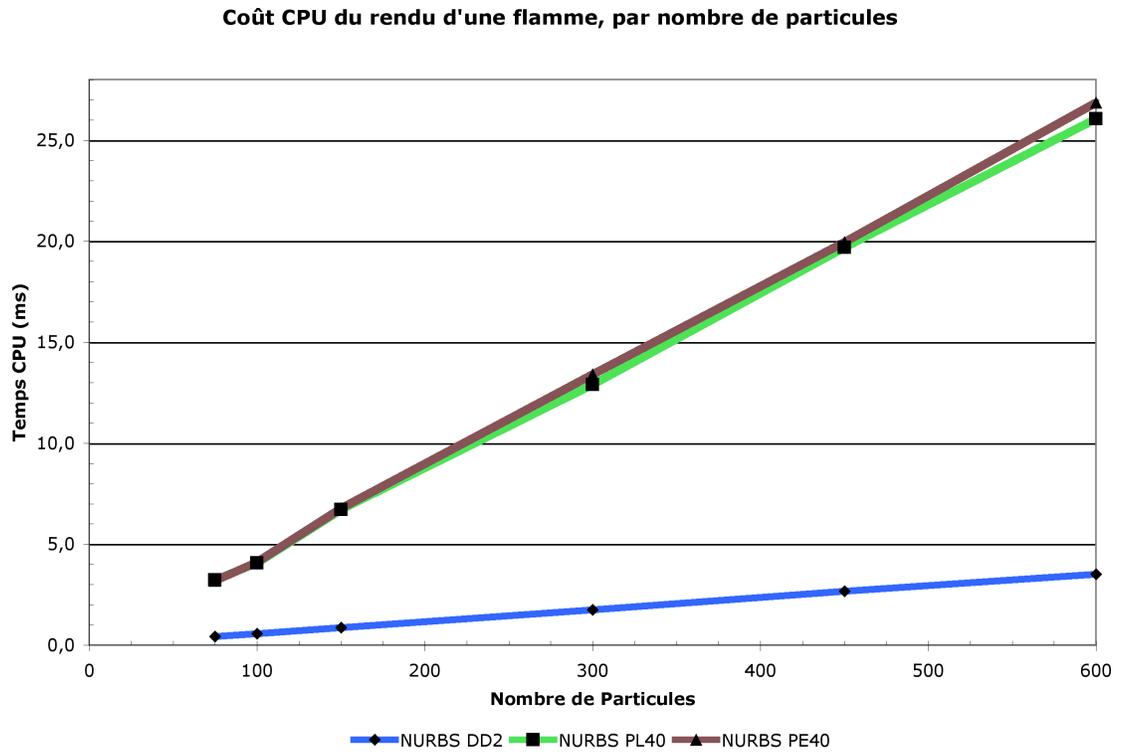


FIGURE 5.6 – Coût CPU du rendu d'une flamme, par nombre de particules. Note : Les configurations PL40 et PE40 se chevauchent.

suffisamment pour exiger la majorité du temps de la trame.

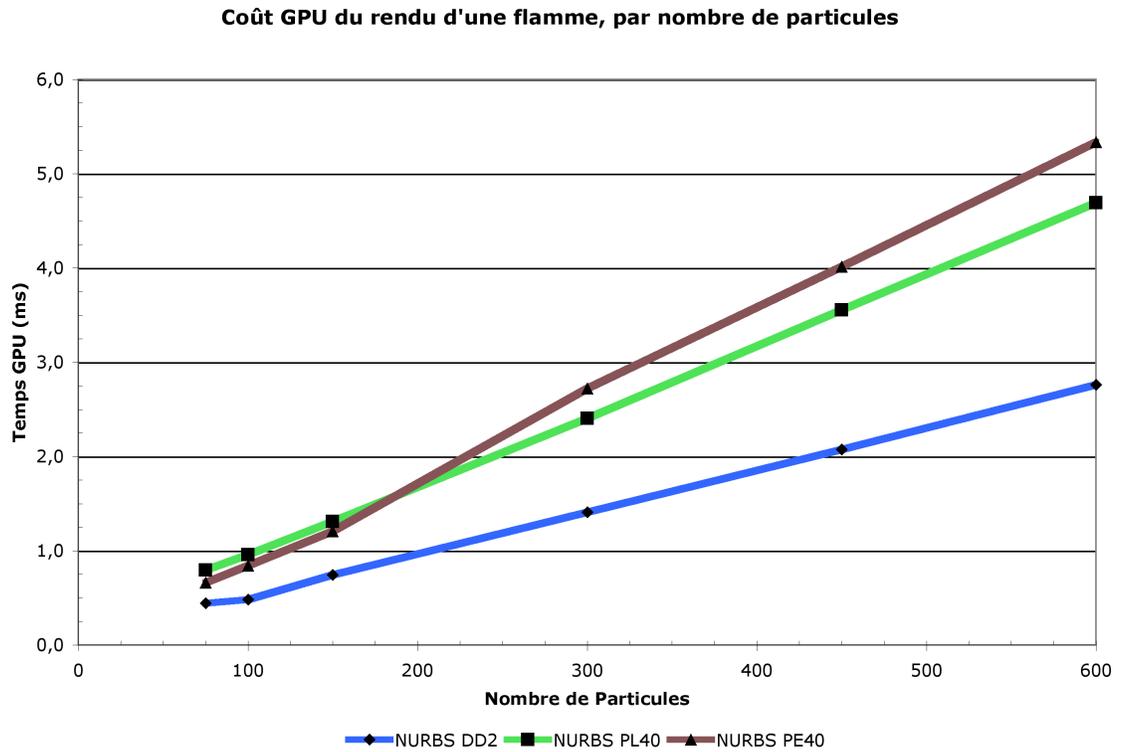


FIGURE 5.7 – Coût GPU du rendu d'une flamme, par nombre de particules.

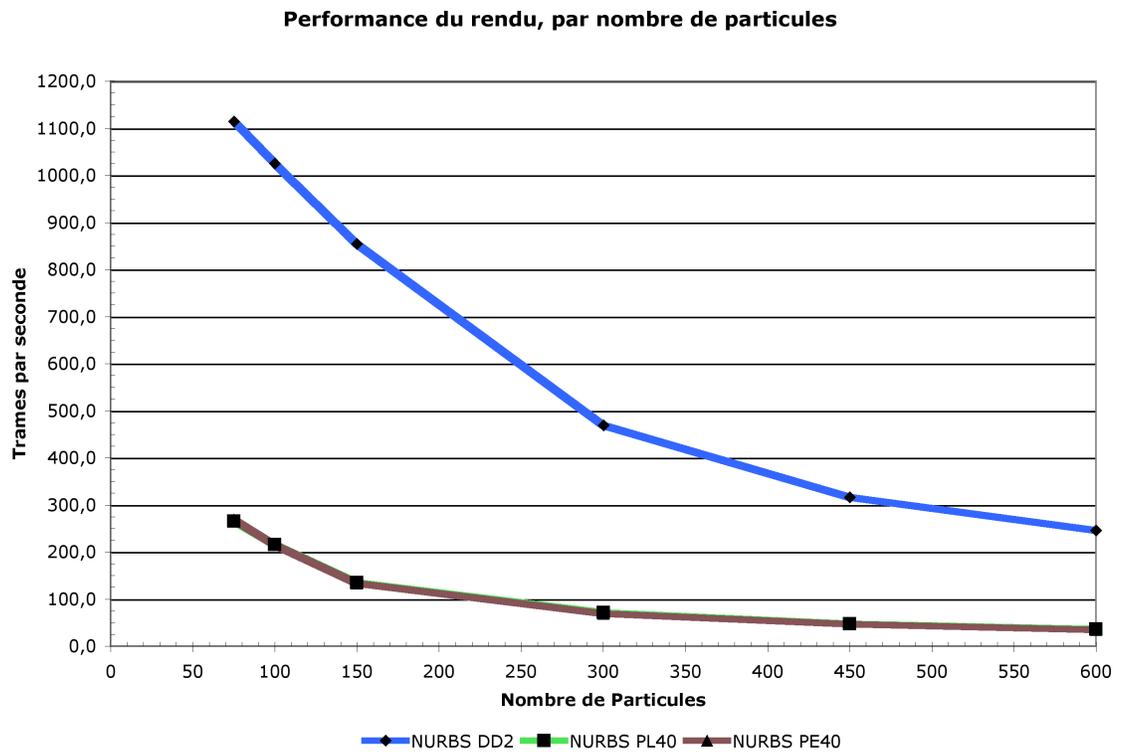


FIGURE 5.8 – Performances du rendu, par nombre de particules. Note : Les configurations PL40 et PE40 se chevauchent.

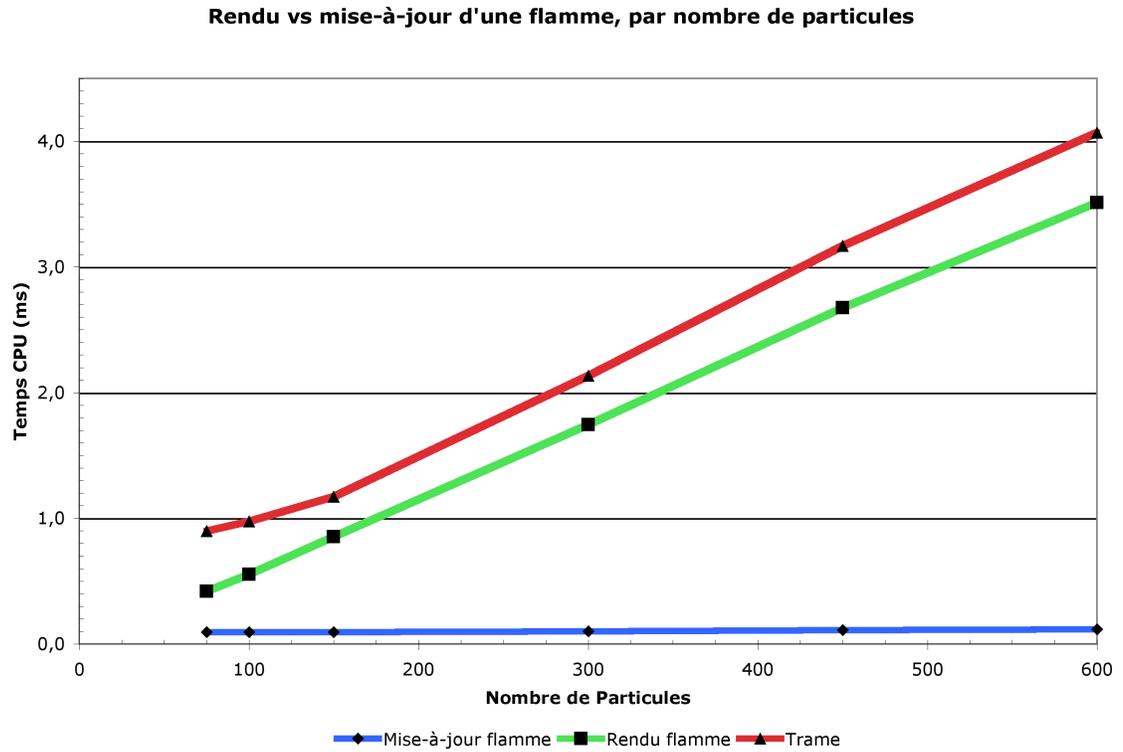


FIGURE 5.9 – Comparatif du rendu versus la mise-à-jour d'une flamme, par nombre de particules.

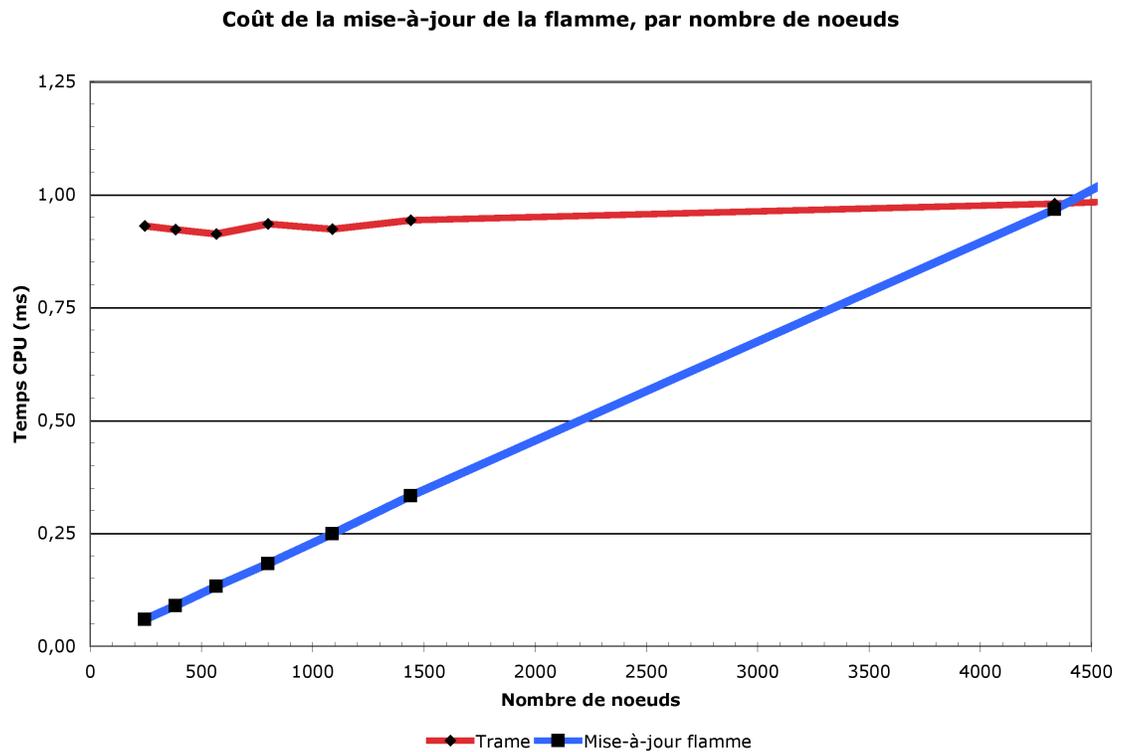


FIGURE 5.10 – Coût de la mise-à-jour de la flamme, par nombre de noeuds de la simulation de ressorts. Valeurs pour une grille de vitesse de tailles 5, 6, 7, 8, 9, 10 et 15.

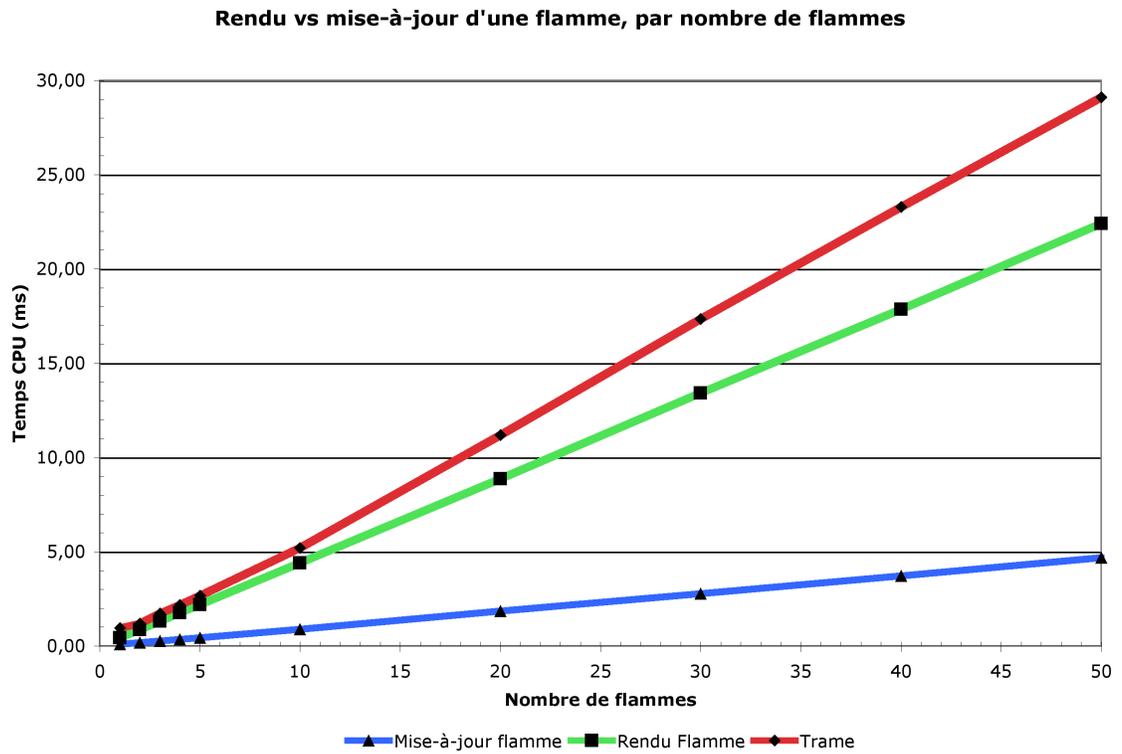


FIGURE 5.11 – Comparatif du rendu versus la mise-à-jour, par nombre de flammes.

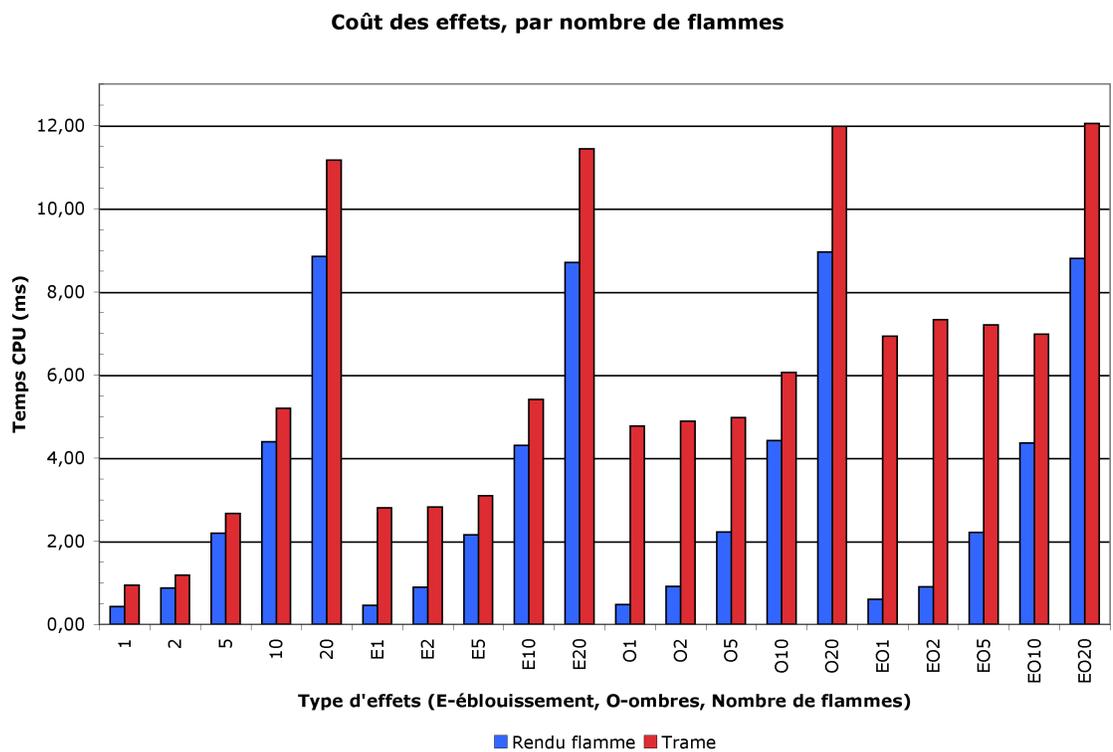


FIGURE 5.12 – Coût des ombres et de l'effet d'éblouissement, par nombre de flammes.

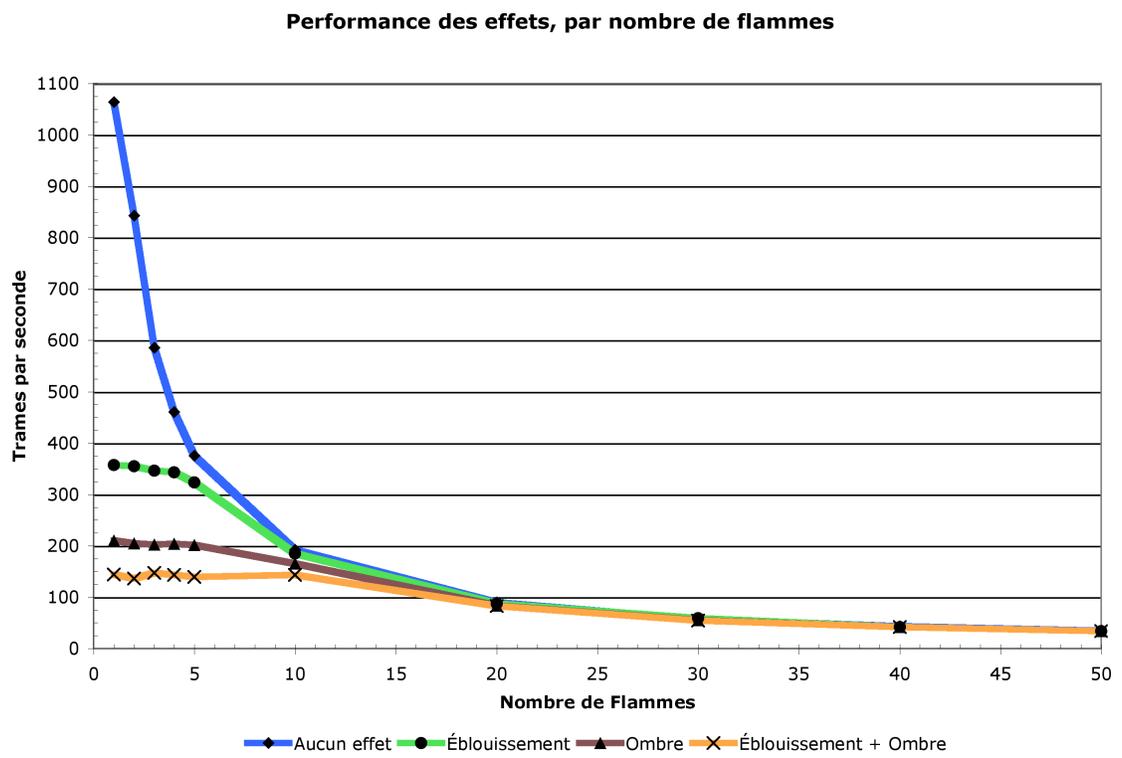


FIGURE 5.13 – Performances des ombres et de l'effet d'éblouissement, par nombre de flammes.

Chapitre 6

Conclusion

[...]
The time to hesitate is through
No time to wallow in the mire
Try now we can only lose
And our love become a funeral pyre
Come on baby, light my fire
Come on baby, light my fire
Try to set the night on fire, yeah
-THE DOORS (LIGHT MY FIRE)

Notre système parvient à générer une flamme laminaire visuellement satisfaisante à un coût réduit, facilitant ainsi une intégration à un système de production 3D sans affecter indûment ses performances. Ce coût est principalement lié à la génération d'une surface NURBS pour le rendu de chaque flamme, mais tend malgré tout vers un taux de 1063 images/s sans effets, et 144 images/s avec ombres et éblouissement. Comme plusieurs systèmes de production intègrent déjà ces effets à leur rendu, leurs non-exclusivités à notre flamme diminuent l'importance de leur impact à coût fixe sur les résultats. L'utilisation de particules dans la modélisation, conjuguée à la pondération de leur vitesse par émetteur et l'utilisation du placage de texture, offre une flexibilité de configuration facilitant la génération de formes et d'apparences variées. Ces particules permettent également d'aisément modéliser l'interaction de la flamme à son environnement, l'impact du vent et du déplacement. Lorsqu'en dehors du champ de vision, la mise à jour et l'affichage de la flamme peuvent être ignorés, permettant de réduire à néant son coût, nonobstant le rendu des ombres et le coût d'illumination par fragment.

6.1 Comparaison

Comme nous visons une synthèse en temps réel, l'étendue des travaux baromètres s'en trouve considérablement réduite. En l'absence de l'accès à leurs implémentations et considérant l'amélioration des capacités matérielles au fil du temps, l'évaluation de leurs performances relatives n'est jugée que sommairement ici.

Dans le rendu d'une flamme avec effets d'éblouissement, Bridault [Bri07] obtient des performances de 290 images/s, versus nos 357 images/s dans une situation comparable sur un matériel inférieur. Leur simulation de fluide apporte un dynamisme convaincant aux mouvements de la flamme, mais à un coût de 2.44ms, bien supérieur à notre mise-à-jour ne nécessitant que 0.06ms. Ce point est important, puisque sur des systèmes de rendu en temps réel, tel qu'un moteur de jeu vidéo, le temps CPU est précieux pour l'exécution de tâches telles que la physique ou l'intelligence artificielle. Il propose toutefois d'en mitiger l'importance en transposant son évaluation sur le GPU, en supposant qu'il n'est pas déjà surchargé par le rendu de la surface NURBS. Il note que la visualisation issue de cette dernière, pose l'inconvénient de rapidement augmenter de coût en fonction de la complexité, ce qui s'applique également à notre situation (voir figure 5.8). Leurs mesures ont été réalisées sur une configuration atteignant 345.6GFlop/s pour la carte graphique et obtenant 1917 points pour l'évaluation standardisée de CPU fait par *PassMark*¹, face à notre carte graphique de 60.8 GFlop/s et nos 1506 points sur *PassMark*. Il est alors raisonnable de conclure que leur approche offre un meilleur dynamisme, mais à un coût supérieur et d'une apparence similaire.

Balci et Foroosh [BF06] présentent une flamme laminaire offrant un taux de rafraichissement de 80 images/s sur du matériel inférieur datant de 2006. Leur publication ne présentant pas les détails techniques de l'implémentation, il est difficile de situer le goulot d'étranglement des performances, mais il est envisageable qu'elles puissent être améliorées en simplifiant le résultat. Le recours à un champ de vitesse procédural exhibant des caractéristiques de turbulence leur permet d'afficher un dynamisme de flamme supérieur à nos résultats lorsque soumise aux forces de vent, mais une technique similaire peut être intégrée à notre système. Outre les performances supérieures, nous nous distinguons clairement par notre capacité à afficher une flamme tridimensionnelle non restreinte à un plan déformé, trahissant sa nature synthétique lors du changement de point de vue.

Fuller et al. [FKM⁺07] parviennent au rendu volumétrique d'une flamme en temps réel. La performance de leurs résultats dépend de la superficie visible à l'écran et peut être ajustée en

1. <http://www.cpubenchmark.net/>

fonction de la qualité désirée. Ils présentent une métrique d'environ 157 images/s pour la seule visualisation sur une carte graphique pouvant atteindre environ 200 GFlop/s, comparativement à nos 1063 images/s avec animation. La dynamique de la flamme requiert un second système venant assigner la position des points de déformations d'un volume englobant, et peut être issue d'un système de particules. Une incorporation du bruit de Perlin permet de générer des flammes complexes plus chaotiques, telle une torche, mais elle est moins appropriée pour une simple flamme laminaire. Au final, nous obtenons de meilleures performances sur une flamme laminaire, mais leur méthode permet possiblement une meilleure apparence pour un feu.

6.2 Perspectives

Notre système offre une flamme laminaire de belle apparence au repos, mais sa réaction au vent et au déplacement ne respecte pas fidèlement la réalité visuelle, ce qui s'explique en partie par l'absence de turbulence. De futurs développements gagneraient à explorer de nouvelles pistes de solutions adressant cet aspect. Malgré nos excellentes performances, la figure 5.9 montre clairement que notre goulot d'étranglement se situe au niveau de la visualisation de la surface NURBS. Dès lors, il semble avantageux d'explorer des alternatives présentant un coût réduit tout en offrant une qualité d'apparence similaire. Une piste potentielle peut être l'utilisation d'un modèle de flamme avec un squelette où l'animation serait issue du positionnement de ses joints selon les déplacements de notre système de particules. Une seconde option consiste à simplement ajuster la durée de vie des particules en fonction de la distance d'une flamme à la caméra. Par exemple, une flamme ne couvrant que 10×20 pixels à l'écran, pourrait réduire son nombre à 5 particules par chaîne, puisque trop distante pour que nous puissions apprécier les subtilités d'une forme modélisée par 50 particules par chaîne.

Comme nos recherches se concentraient sur la simulation d'une flamme, nous n'avons pas exploré l'utilisation d'une technique de segmentation de l'espace. Son implémentation permettrait d'obtenir un gain considérable de performances en réduisant la quantité de géométrie traitée. Présentement, la scène complète est rendue 7 fois par trame, afin de générer nos ombres omnidirectionnelles en plus du point de vue de la caméra.

Nous avons concentré nos travaux sur la synthèse de la flamme d'une bougie, mais il est aisé d'adapter le système à la reproduction de flammes plus complexes, telles qu'une torche, en combinant plusieurs flammes entre elles, aussi proposé par Bridault [Bri07]. Cette voie nécessite par contre une technique de gestion de la semi-transparence, aux implications décrites à la

section 4.3.3. Quant à Fuller et al. [FKM⁺07], ils parviennent à modéliser la nature chaotique des détails d'une flamme à l'aide d'un *bruit de Perlin amélioré* venant moduler l'accès à la texture et il serait alors intéressant d'explorer une technique similaire pour ce type de modélisation.

Beaudoin et al. [BPP01] décrivent dans leurs travaux, une technique modélisant les courts segments de flammes se détachant sporadiquement de la flamme principale. Comme nous utilisons également un système de particules, nous pouvons en transférer l'application à notre implémentation. Cependant, considérant de leur brève espérance de vie, simplement les simuler à l'aide d'une technique de *billboarding* orienté écran peut être plus adéquat, comme elles disparaissent trop rapidement pour que l'on puisse réellement noter leur aspect tridimensionnel. Une technique similaire pour la fumée est également envisageable.

Finalement, un aspect de la combustion laissé en suspens est l'émission de chaleur affectant la trajectoire des rayons de lumière par réfraction. Visuellement, nous percevons ce phénomène par le chatolement de l'air avoisinant, sous les courants thermiques induits. Bien qu'une flamme de chandelle ne possède pas une intensité suffisante pour son occurrence, nous associons fortement ce phénomène à la présence de chaleur. Dès lors, il serait intéressant d'inclure une technique existante de post-traitement (*heat shimmer*) le simulant. Puisque la méthode est similaire à l'effet d'éblouissement et peut réutiliser des étapes de son rendu, son coût additionnel est limité et fixe puisqu'elle ne dépend que de la résolution de l'écran, comme tout post-traitement.

Bibliographie

- [BF06] Murat BALCI et Hassan FOROOSH : Real-time 3d fire simulation using a spring-mass model. *In MMM (Multi media modeling)*, 2006.
- [BM08] L. BAVOIL et K. MYERS : Order independent transparency with dual depth peeling. *In Tech. rep., NVIDIA Corporation*, 2008.
- [Boy06] Louis BOYER : *Feu et flammes*. Pour la science. Belin, 2006.
- [BPP01] Philippe BEAUDOIN, Sébastien PAQUET et Pierre POULIN : Realistic and controllable fire simulation. *In Graphics Interface 2001*, pages 159–166, Juin 2001.
- [Bri07] Flavien BRIDAULT : *Animation et Rendu temps réel de flammes*. Thèse de doctorat, Université du Littoral Côte d’Opale, 2007.
- [DL06] William DONNELLY et Andrew LAURITZEN : Variance shadow maps. *In I3D ’06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 161–165, New York, NY, USA, 2006. ACM.
- [FKM⁺07] Alfred R. FULLER, Hari KRISHNAN, Karim MAHROUS, Bernd HAMANN et Kenneth I. JOY : Real-time procedural volumetric fire. *In I3D ’07 : Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 175–180, New York, NY, USA, 2007. ACM.
- [HG09] Christopher HORVATH et Willi GEIGER : Directable, high-resolution simulation of fire on the GPU. *In SIGGRAPH ’09 : ACM SIGGRAPH 2009 papers*, pages 1–8, New York, NY, USA, 2009. ACM.
- [HL97] W. HUANG et B. LEIMKUEHLER : The adaptive verlet method. *In SIAM J. Sci. Comput.*, pages 239–256, 1997.

- [HSF07] Jeong-Mo HONG, Tamar SHINAR et Ronald FEDKIW : Wrinkled flames and cellular patterns. *ACM Trans. Graph.*, 26(3):47, 2007.
- [KCR99] Scott A. KING, Roger A. CRAWFIS et Wayland REID : Fast animation of amorphous and gaseous phenomena. In *Proceedings of Volume Graphics (VG99)*, pages 333–346, 1999.
- [LF02] Arnauld LAMORLETTE et Nick FOSTER : Structural modeling of flames for a production environment. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 729–735, New York, NY, USA, 2002. ACM.
- [MK02] Zeki MELEK et John KEYSER : Interactive simulation of fire. In *PG '02 : Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, pages 65–70, Washington, DC, USA, 2002. IEEE Computer Society.
- [NFJ02] Duc Quang NGUYEN, Ronald FEDKIW et Henrik Wann JENSEN : Physically based modeling and animation of fire. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 721–728, New York, NY, USA, 2002. ACM.
- [Per85] Catherine PERLÈS : *L'homme et le feu*. 1985.
- [Ree83] William T. REEVES : Particle systems—a technique for modeling a class of fuzzy objects. In *SIGGRAPH '83 : Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 359–375, New York, NY, USA, 1983. ACM.
- [SF95] Jos STAM et Eugene FIUME : Depicting fire and other gaseous phenomena using diffusion processes. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 129–136, New York, NY, USA, 1995. ACM.
- [SM05] Timo STICH et Marcus MAGNOR : Learning flames. In *Proc. 10th International Fall Workshop - Vision, Modeling, and Visualization (VMV)*, pages 65–70, 2005.
- [SSSE00] Arno SCHÖDL, Richard SZELISKI, David H. SALESIN et Irfan ESSA : Video textures. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer gra-*

phics and interactive techniques, pages 489–498, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[Sta99] Jos STAM : Stable fluids. *In SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[WLMK02] Xiaoming WEI, Wei LI, Klaus MUELLER et Arie KAUFMAN : Simulating fire with texture splats. *In VIS '02 : Proceedings of the conference on Visualization '02*, pages 227–235, Washington, DC, USA, 2002. IEEE Computer Society.