

UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

Les ombres au sein des jeux et des animations

Bruno Rocha Pereira
Pierre Gérard
Quentin Ravau
Antoine Carpentier

Superviseurs :

Tom Lenaerts et Jean-Sébastien Lerat

Contents

Chapter 1

Introduction

Une ombre est une “zone sombre résultant de l’interception de la lumière ou de l’absence de lumière”¹. C’est un élément indispensable au réalisme d’une scène d’animation ou de jeu vidéo. En effet ce sont les ombres qui vont apporter l’information quand à la position relative et à la taille des objets qui créent l’ombre. Dans le cas d’objets complexes, elles permettent d’obtenir des informations sur la forme des objets. Dans le monde réel, on est souvent confronté à plusieurs sources lumineuses qui apportent chacune leur lot d’informations en plus.

Nous allons, lors de ce projet d’année, étudier l’impact des ombres sur le réalisme des animations et des jeux vidéos. Cette étude sera réalisée en utilisant et comparant différents algorithmes de génération d’ombre, plus ou moins réalistes.

Nous nous intéresserons plus particulièrement aux algorithmes en temps réel car ceux-ci sont plus intéressant et plus attractifs pour une présentation pour le Printemps des Sciences².

La littérature scientifique distingue deux types d’ombres : les *soft shadows* et les *hard shadows*. Les premières ont des bords diffus et les secondes n’en ont pas. Plus la source de lumière est proche d’un objet, plus les bords d’une ombre réaliste sont diffus et inversement. Dans ce projet, nous nous intéresserons uniquement aux algorithmes de génération des *soft shadows* car ils permettent également de générer des *hard shadows*.

Nous testerons les différents aspects et effets des ombres dans différents scénarios que nous mettrons en application. Nous ferons varier les objets, les sources lumineuses, leur nombre et leur mouvement afin de présenter des situations se rapprochant de la réalité.

¹<http://www.larousse.fr/dictionnaires/francais/ombre/55933>

²<http://www.printempsdessciences.be>

Chapter 2

Présentation des articles

Shadow Algorithms Data Miner

[?]

Ce livre se veut la référence pour les différents types d'ombres et d'algorithmes existants dans la littérature spécialisée. Il explique les *hard shadows* et les *soft shadows*, ainsi que les principaux types d'algorithmes utilisés comme le *shadow mapping*, le *shadow volume* et le *ray tracing*. De plus, l'article va aussi plus en profondeur en expliquant les problèmes que l'on pourrait rencontrer, notamment dus au *self shadowing* ou au *bump mapping*.

Ainsi, même si tous les chapitres ne nous intéresseront peut-être pas, cet article répondra certainement à beaucoup de nos questions et nous a déjà permis d'orienter nos recherches dans une certaine direction.

A survey of Real-Time Soft Shadows Algorithms

[?]

Cet article décrit le principe des ombres et discute leur importance. Il nous explique que les ombres influencent notre perception de la position relative et la taille des objets.

Nous avons décidé de retenir cet article car il décrit les différences existantes entre les *soft shadows* et les *hard shadows*. Il décrit également deux types d'algorithmes (le *shadow mapping algorithm* et le *shadow volume algorithm*) de manière claire et précise, en spécifiant les points faibles et les points forts de chacun. Cet article présente également des améliorations possibles aux différents algorithmes, qui permettent de réduire le coût des algorithmes et donc se rapprocher encore plus du temps réel.

Cet article nous aidera donc beaucoup pour les algorithmes de *shadow mapping* et de *shadow volume*.

A Survey of Shadow Algorithms

[?]

Bien que cet article date un peu, il nous sera utile car il caractérise les différents types d'ombres. Il décrit également les algorithmes de génération d'ombres à la base des techniques actuelles et discute de leurs complexités, leurs avantages et leurs inconvénients. Il examine les *hard shadows*, *soft shadows* ainsi que les ombres d'objets transparents. Cet article devrait donc nous indiquer quel algorithme serait le mieux adapté à différentes situations que nous pourrions rencontrer.

Algorithms for Dynamic Shadows

[?]

Cet article part d'une base très simple, c'est à dire simplement dessiner un rond en dessous des objets qui bougent vers des exemples d'algorithmes plus complexes et réalistes. De part sa manière progressive d'expliquer la problématique et les solutions relativement simples de la représentation des ombres, cet article nous aidera à mieux comprendre les différentes difficultés qui nous attendent et à aborder les algorithmes de manière didactique.

GEARS: A General and Efficient Algorithm for Rendering Shadows

[?]

Cet article décrit un algorithme de *shadow mapping* qui permet d'obtenir une scène interactive tout en gardant une ombre correcte et précise. De plus, il offre un niveau de *FPS* tout à fait correct. Il est donc tout à fait compatible avec notre optique de nous orienter vers des algorithmes en temps réel. Cet algorithme est, selon ses auteurs, le plus performant parmi les algorithmes actuels de *soft shadows* en temps réel.

Cet article nous conforte dans l'idée de nous diriger vers les algorithmes de représentation de l'ombre en temps réel et d'éviter l'algorithme *ray tracing* qui fournit moins de *FPS* et moins de précision que les algorithmes de *shadow mapping* et *shadow volume*.

A Survey of Shadow Volume Algorithms in Computer Graphics

[?]

Cet article présente les différentes recherches publiées sur le sujet des algorithmes de *shadow volume*. Pour chaque publication un avis critique est donné. Cela nous sera utile pour, par exemple, savoir quel papier n'est plus d'actualité. L'auteur passe ensuite en revue les différents algorithmes disponibles et précise les améliorations que les chercheurs y ont apportées, avec les avantages et les inconvénients que chacune de ces améliorations peut apporter.

Il nous sera utile pour implémenter et optimiser nos algorithmes de *shadow volume*.

An Improved Physically-Based Soft Shadow Volume Algorithm

[?]

Cet articles identifie et analyse plusieurs problèmes de performance dans les algorithmes *shadow volume* de haute qualité et présente une méthode améliorée qui atténue ces problèmes en remplaçant une structure d'accélération spatiale trop conservatrice par une autre plus efficace. Il nous permettra d'améliorer de façon substantielle nos algorithmes de *shadow volume*.

A Survey of Real-Time Hard Shadow Mapping Methods

[?]

Nous nous sommes également penchés vers les algorithmes de génération des *hard shadows*. Pour des raisons de polyvalence, de vitesse et de robustesse, les algorithmes de type *shadow mapping* sont les plus adaptés. Cet article propose une vision générale des *hard shadows*

Toutefois, nous n'implémenterons pas d'algorithme de *hard shadows* en tant que telles, car celles-ci n'apparaissent que rarement dans la nature, et sont donc moins réalistes. Nous les utiliserons dans le but de les comparer avec les *soft shadows* et de démontrer l'apport de réalisme dans ces dernières.

Chapter 3

Description de l'implémentation

Dans ce chapitre, nous décrirons les langages et bibliothèques utilisés ainsi que les algorithmes que nous allons implémenter.

3.1 Render 2D et 3D

OpenGL s'est imposé comme l'API de choix étant donné sa spécification ouverte, ses fonctions bas niveau et sa disponibilité sur un grand nombre de plate-formes. Nous utiliserons OpenGL pour créer des scènes 3D, animer une/des caméra(s) et une/des source(s) de lumière mais également pour générer des ombres avec les algorithmes retenus.

Nous essayerons d'utiliser un maximum des fonctions modernes d'OpenGL majoritairement utilisées dans les jeux et animations d'aujourd'hui. Cependant, les consignes nous imposent virtualbox qui ne permet pas, au premier abord, d'utiliser de tels outils. Nous allons donc faire un maximum de recherche pour passer outre cette limitation.

3.2 Langages et bibliothèques utilisés

Dans cette section, nous allons présenter rapidement les langages de programmation et bibliothèques que nous allons utiliser dans notre projet.

- Nous allons utiliser **Python 2.7** comme langage de programmation car il permet un développement rapide et possède des bindings vers les bibliothèques OpenGL, OpenCL, numPy etc... écrites en C/C++. Il permet donc d'allier la rapidité d'écriture des langages de scripts à la rapidité d'exécution des langages compilés. Nous utiliserons aussi Python Package Index et les virtualenv pour nous faciliter la tâche.
- Nous allons utiliser **pyQt** comme librairie qui s'interfacera avec la célèbre bibliothèque **Qt 5**. Nous utiliserons cette interface graphique étant donnée sa facilité d'utilisation et son caractère complet. De plus nous l'avons pour la plupart déjà utilisée durant notre cursus.
- Nous allons également utiliser **OpenCL** pour profiter de puissance de calcul des cartes graphiques modernes et ainsi utiliser des algorithmes puissant en temps réel. OpenCL nous permettra d'être très polyvalent puisque le même programme peut s'exécuter indifféremment sur un CPU ou n'importe quel GPU muni du pilote approprié sans devoir être recompilé.

3.3 Algorithmes retenus

Dans cette section, nous présentons les deux types d'algorithmes que nous allons implémenter et qui sont les plus répandus et les plus efficaces dans le monde de la génération d'ombre. Pour chaque type

d'algorithmes, nous commencerons par implémenter une version basique et puis nous l'améliorerons en nous aidant des recherches les plus récentes. Nous comparerons ensuite les performances et le réalisme entre la version basique et la version améliorée ainsi qu'entre les deux types d'algorithmes.

3.3.1 Shadow volume

Le *shadow volume* est une technique utilisée en graphisme 3D pour générer les ombres. Il divise le monde en deux zones: celles qui sont dans l'ombre et celles qui ne le sont pas. Il consiste à tracer un vecteur entre la source de lumière et chaque objet générant de l'ombre afin d'obtenir un volume correspondant à la zone placée dans l'ombre. L'avantage de ces algorithmes est leur précision et leur réalisme mais ils peuvent être moins efficaces que le *shadow mapping*. Nous allons les implémenter car ils sont très répandus dans le monde du jeu vidéo.

3.3.2 Shadow mapping

Le principe de base de cet algorithme est que si l'on regardait depuis le point lumineux, tous les objets seraient éclairés. Par contre, tout ce qui se trouve derrière ces objets est ombragé. On crée ainsi une *shadow map* qui stocke la profondeur de ces objets. Ensuite, on compare cette *shadow map* avec les pixels que l'on voit depuis la vue de la caméra et on obscurcit les pixels nécessaires. Ces algorithmes permettent de ne générer les ombres que du point de vue de la caméra et donc de s'épargner un grand nombre de calculs inutiles, ce qui améliore les performances mais leur réalisme dépend fortement de la résolution de la *shadow map*.