



**HAL**  
open science

## Simulation réaliste de ruisseaux en temps réel

Antoine Pericchi

► **To cite this version:**

Antoine Pericchi. Simulation réaliste de ruisseaux en temps réel. Synthèse d'image et réalité virtuelle [cs.GR]. 2004. inria-00598419

**HAL Id: inria-00598419**

**<https://inria.hal.science/inria-00598419>**

Submitted on 6 Jun 2011

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

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

# Simulation réaliste de ruisseaux en temps réel



Antoine Pericchi

DEA Imagerie, Vision, Robotique

Stage effectué au sein de l'équipe Evasion du laboratoire  
GRAVIR

Sous la direction de Fabrice Neyret



# Table des matières

|         |  |    |
|---------|--|----|
| 1       | Introduction.....  | 4  |
| 2       | État de l'art.....   | 6  |
| 2.1     | Animation des fluides.....   | 6  |
| 2.1.1   | Simulation des fluides.....  | 6  |
| 2.1.2   | Approches spectrale et procédurales.....                                   | 6  |
| 2.1.3   | Approche phénoménologique.....   | 6  |
| 2.2     | Rendu.....   | 8  |
| 2.2.1   | Lois de l'optique géométrique.....   | 8  |
| 2.2.1.1 | Réflexion.....   | 8  |
| 2.2.1.2 | Réfraction.....  | 9  |
| 2.2.1.3 | Coefficient de Fresnel.....  | 10 |
| 2.2.2   | Lancer de rayon.....   | 10 |
| 2.2.3   | Rendus adaptatifs.....   | 11 |
| 2.2.4   | Environment mapping.....   | 11 |
| 2.2.5   | Bump mapping.....  | 12 |
| 2.2.6   | Fonctionnalités des cartes graphiques utilisables pour le rendu de l'eau . | 12 |
| 2.2.6.1 | Shaders.....   | 13 |
| 2.2.6.2 | Carte de hauteur.....  | 13 |
| 2.2.6.3 | Environment mapping.....   | 13 |
| 2.2.6.4 | Reflective bump mapping.....   | 13 |
| 3       | Notre approche pour le rendu des ruisseaux.....                            | 14 |
| 3.1     | Interaction lumière/surface de l'eau.....                                  | 14 |
| 3.2     | Modélisation des ondes.....  | 15 |
| 3.2.1   | Identification des phénomènes à modéliser.....                             | 15 |
| 3.2.2   | Géométrie des ondes.....   | 17 |
| 3.2.3   | Courbure des ondes.....  | 18 |
| 3.2.3.1 | Représentation des normales à la surface des ondes .....                   | 18 |
| 3.2.3.2 | Exploitation de l'information d'une carte de hauteur.....                  | 22 |
| 3.2.3.3 | Avantages d'une représentation par carte de hauteur.....                   | 22 |
| 3.3     | Rendu adaptatif.....   | 22 |
| 3.3.1   | Identification des différentes résolutions.....                            | 22 |
| 3.3.2   | Simplification et transition de la géométrie .....                         | 23 |
| 3.3.3   | Le problème de sous-échantillonnage.....                                   | 23 |
| 3.3.4   | Transition des normales.....   | 24 |
| 4       | Resultats.....   | 25 |
| 5       | Conclusion et Perspectives .....   | 27 |

# 1 Introduction

Les effets spéciaux ou les jeux vidéos sont de plus en plus demandeurs de scènes naturelles. Cela s'explique par le désir des artistes d'aller vers un réalisme de plus en plus saisissant. Les techniques habituellement utilisées pour modéliser les scènes naturelles sont issues de l'ingénierie, et sont initialement destinées aux simulations physiques (ex: simulation de barrage). Mais il faut garder toutefois à l'esprit que le but d'un artiste est bien différent de celui de l'ingénieur. Il faut qu'il soit capable de générer rapidement des images plausibles et "belles", et non des images qui soient "physiquement exactes".

La modélisation de phénomènes naturels tels que les nuages, le feu, la fumée, ou les rivières constitue un défi pour l'informatique graphique. En effet les méthodes traditionnelles se révèlent la plupart du temps inefficaces face à la complexité du phénomène. Soit parce que la résolution numérique de la physique sous-jacente est trop complexe, ou mal connue, soit parce que les phénomènes se produisent dans un domaine spatial trop large.

Notre ambition à long terme est de modéliser les ruisseaux de manière photo-réaliste et en temps réel. L'étude se porte ici sur les ruisseaux, car ils constituent le cas pourtant simple, mais visuellement déjà très riche, de fluides en écoulement quasi-stationnaire non-dispersif. Cela permet de nombreuses simplifications dans la résolution numérique des champs de vitesse.

L'approche épistémologique de ce mémoire est une approche phénoménologique, essentiellement basée sur l'observation directe de cours d'eau, ou de matériel photographique et vidéo. Il s'agit d'une approche originale qui offre de nombreux avantages par rapport aux simulations classiques, en terme de contrôle, de qualité visuelle et d'efficacité.

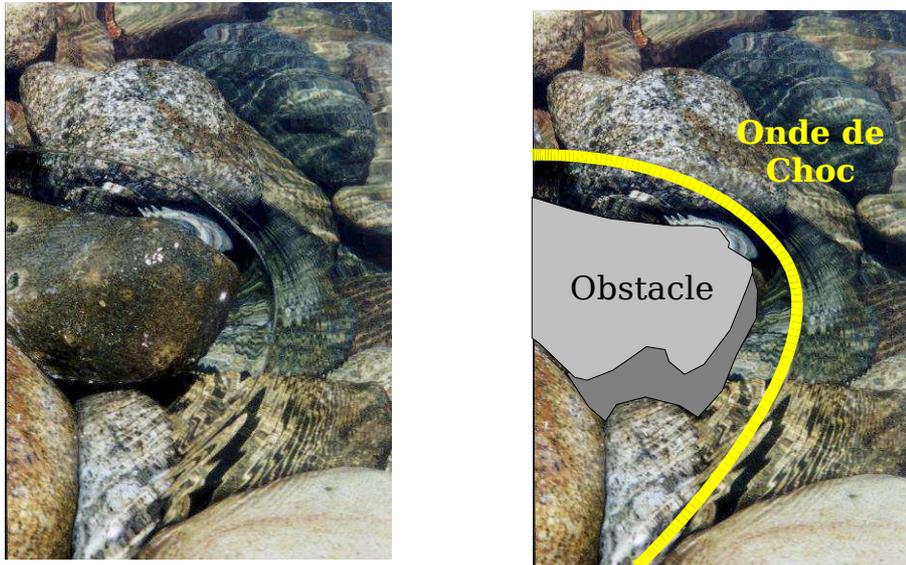
L'un des souhaits des artistes est de pouvoir contrôler directement et visuellement certaines caractéristiques du phénomène, sans avoir de connaissances préalables de la physique sous-jacente. La plupart des techniques actuelles de rendu se basent sur des simulations physiques, dont les paramètres sont difficiles à contrôler. L'étude proposée atteint justement cet objectif en séparant les phénomènes en primitives aisément modifiables.

Cette approche pourrait être qualifiée également de "minimaliste" dans le sens où seul le phénomène attendu est modélisé. La géométrie est créée uniquement là où nécessaire, et des détails peuvent être ajoutés de manière adaptée. Enfin l'animation fait appel à des solutions numériques qui peuvent être spécifiques et peu coûteuses. En contrepartie, la possibilité d'apparition de phénomènes émergents est donc alors complètement exclue.

D'autre part, cette phénoménologie des fluides peut conduire à une modélisation qui n'est pas quantitativement exacte. Mais la simulation précise n'est pas le but de notre démarche, qui ne cherche seulement qu'à obtenir une image animée visuellement crédible. Le cerveau humain est en effet incapable dans la plupart des cas de déceler les inexactitudes liées à des paramètres erronés, et saisit plutôt un phénomène dans son ensemble. Réciproquement les artefacts dus à la résolution limitée des approches classiques sont très visibles.

Le cours d'un ruisseau présente plusieurs phénomènes à modéliser. Tout d'abord, l'eau est un milieu transparent dont la principale caractéristique visible pour l'oeil humain est son interface avec l'air, qui réfléchit le ciel et réfracte le fond de l'eau.

Ensuite des ondes de chocs se produisent en amont des obstacles, et présentent une géométrie oscillant autour d'une position d'équilibre du fait du régime quasi-statique du cours d'eau. Dans cette étude, nous allons nous concentrer exclusivement sur ces deux phénomènes.



Pour l'instant le ruisseau modélisé est plan, et seuls quelques obstacles y provoquent des perturbations. D'autres phénomènes devraient être modélisés dans le futur, comme par exemple les tourbillons, la mousse, les sillages de von Karman, ou l'interaction de la lumière avec le fond du ruisseau (caustiques).

Ce stage s'intéresse exclusivement au rendu de l'aspect visuel de la surface du ruisseau et non de l'animation du fluide en lui-même. Cet aspect a été étudié en détail lors du stage de Nathalie Praizelin, qui propose des méthodes de construction phénoménologiques et temps réel pour simuler le mouvement des ondes de choc. Mon stage s'effectue donc dans la continuité de ces travaux déjà engagés.

Le but de ce mémoire est de proposer un modèle complet des ondes de choc permettant de visualiser ces phénomènes de manière interactive. Pour cela, j'ai choisi de m'appuyer sur les fonctionnalités des cartes graphiques, qui offrent des performances élevées.

Je dresserai dans un premier temps un état de l'art des techniques qui ont déjà été utilisées pour le rendu de l'eau, ainsi que les principales fonctionnalités des cartes graphiques actuelles. Dans un deuxième temps je proposerai une modélisation des ondes de chocs.

## 2 État de l'art

### 2.1 Animation des fluides

Les différentes approches ayant pour but de simuler les fluides se divisent principalement en trois grandes catégories.

#### 2.1.1 Simulation des fluides

Cette approche correspond à la tendance générale dans le domaine de la synthèse d'image [1][2][3]. Elle génère des résultats visuellement très riches et reproduit la complexité du mouvement des fluides au prix de calculs coûteux.

Le principe repose souvent sur une résolution d'équations différentielles issue directement de la mécanique des fluides [4][5](Navier-Stokes ou formes simplifiées) aux sommets d'une grille. D'autres modèles physiques peuvent être utilisés comme le principe d' Huygens [6] .

Bien que cette approche modélise très bien les fluides turbulents, elle ne tient pas compte des ondes de chocs ou des ondes de tension superficielles, d'une résolution fine. Cela nécessiterait en effet une grille de très haute résolution et des calculs supplémentaires conséquents.

#### 2.1.2 Approches spectrale et procédurales

Ces approches tentent de reproduire des phénomènes sans s'intéresser aux causes, contrairement à l'approche précédente.

L'approche spectrale [7][8][9] utilise basiquement un champ de hauteur (cf plus loin) ayant le même spectre que la surface à animer. Cela peut être fait en filtrant un bruit blanc avec les filtres de Pierson-Moskowitz ou d'Hasselmann, et calculer ensuite sa transformée de Fourier. Le principal avantage est que plusieurs ondes de fréquences différentes sont simultanément simulées. On peut noter que ce modèle a été implémenté pour des films tels que Titanic ou Waterworld[10].

L'approche procédurale utilise des fonctions mathématiques explicitement définies par l'utilisateur qui possèdent des propriétés particulières comme l'auto-similarité dans le cas des animations fractales.

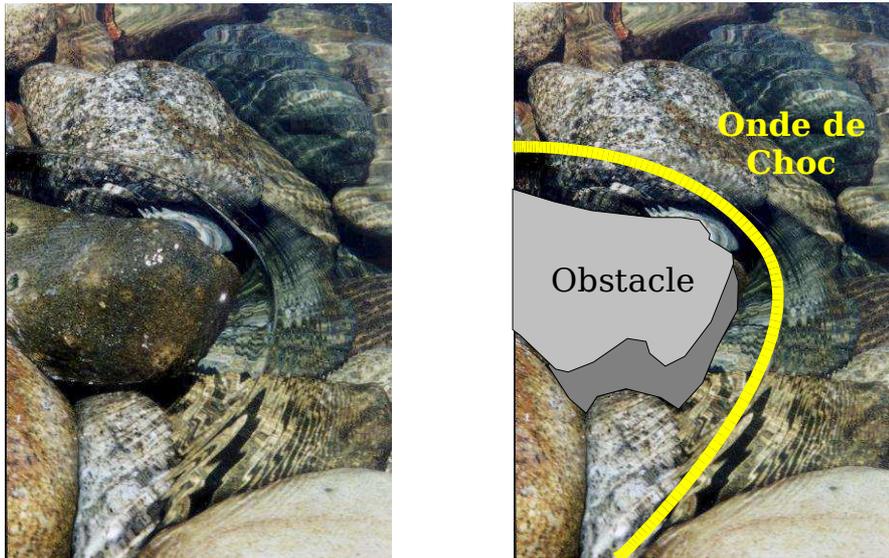
Ces deux approches donnent généralement de très bons résultats sur de larges surfaces comme l'océan ou le vent, mais sont plus difficilement applicables sur des écoulements étroits. Dans les deux cas, le contrôle local de l'animation s'avère extrêmement complexe, voire impossible.

#### 2.1.3 Approche phénoménologique

L'approche phénoménologique, ou encore descriptive, se préoccupe essentiellement des manifestations "de surface" d'un phénomène préalablement clairement identifié.

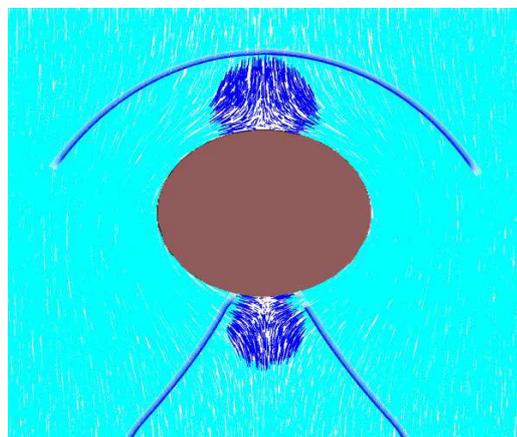
Elle cherche à modéliser ces phénomènes à l'aide de modèles analytiques simplifiés. Nous pouvons donner comme exemples les trochoïdes [11][12] pour les vagues d'un océan, ou les champs laplaciens pour le vent.

La modélisation des ruisseaux effectuée dans [13], constitue la base de notre étude. Elle s'intéresse aux ondes de chocs en amont d'obstacle, ou d'irrégularité du fond du ruisseau.



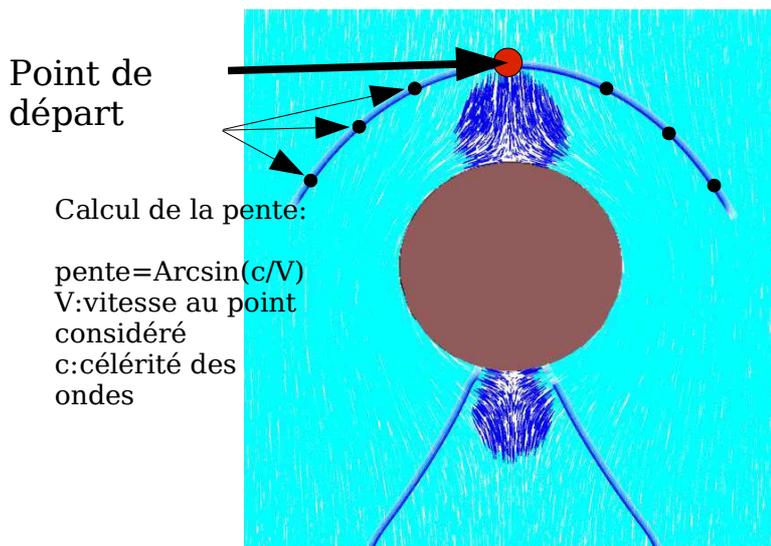
Ces ondes, appelées "ondes de Froude", sont similaires aux ondes de chocs provoquées par la pénétration des avions dans l'air. En effet, le nombre de Mach est similaire au nombre de Froude

$Fr = V/c$  où  $V$  est la vitesse du fluide à un point considéré, et  $c$  est la célérité des ondes dans le milieu. On observe donc une forme de cône en amont de l'obstacle similaire au cône de Mach, et d'une ouverture égale à  $\arcsin(c/V)$ .



Les ruisseaux sont généralement "super-critiques" (ie Froude  $>1$ ), mais dans les zones proches de l'obstacle, le ruisseau est "sous-critique" (ie Froude  $<1$ ). Dans les zones sous-critiques, les ondes peuvent remonter, mais pas dans les zones sur-critiques. Les ondes s'accablent alors selon une ligne, qui correspond à l'iso-Froude  $Fr=1$ , et forment l'onde principale observée.

L'apport essentiel de [13], est d'avoir mis en évidence la propriété géométrique remarquable de ces ondes, car il est possible de les construire par itération successive à partir d'un point de départ selon une pente égale à  $\arcsin(1/Fr)$  |



En prenant comme hypothèse que le fluide est stationnaire, plan, incompressible, irrotationnel, et idéal, les vitesses sont pré-calculées en évaluant une équation de Laplace simple aux sommets d'une grille large. Les vitesses à l'intérieur des cellules sont alors simplement interpolées. Ce champ de vitesse statique est alors perturbé localement, et les lignes d'iso-froude sont géométriquement construites à partir de celui-ci.

Cette méthode s'effectue en temps réel, et produit des résultats vectoriels, indépendants de la résolution.

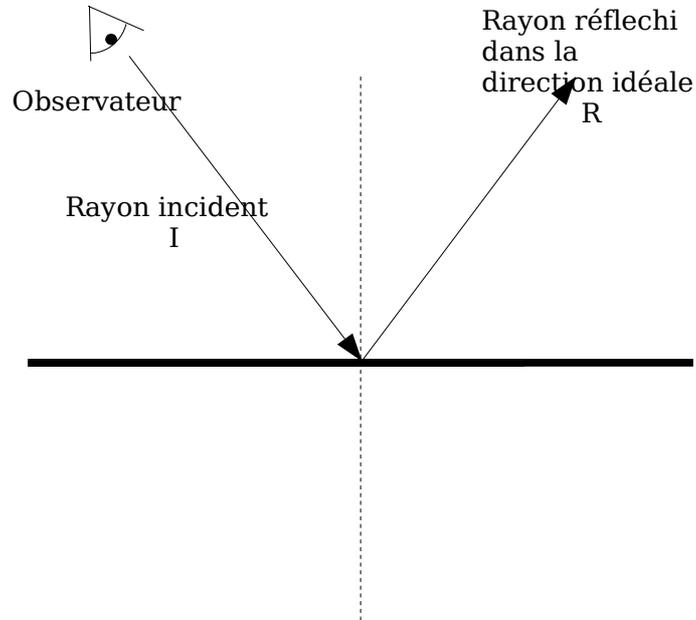
## 2.2 Rendu

### 2.2.1 Lois de l'optique géométrique

Ces lois sont connues depuis Descartes. Elles décrivent bien le comportement de la lumière si l'on considère que la lumière se propage en ligne droite dans un milieu homogène. Mais elles ne permettent pas de modéliser les phénomènes d'interférence ou de diffraction, qui font appel à la nature ondulatoire de la lumière.

#### 2.2.1.1 Réflexion

Un objet hautement réfléchissant, comme une sphère chromée, ne fait que refléter son environnement. Si l'objet est parfaitement réfléchissant, alors il reflète le rayon d'observation, qui part de l'oeil, touche un point de la surface, se réfléchit, et continue dans l'environnement. Le rayon incident est réfléchi symétriquement par rapport à la normale à la surface.



Un rayon réfléchi obéit à l'équation:

$$R = I - 2N(N \cdot I)$$

avec  $R$  est le vecteur du rayon réfléchi

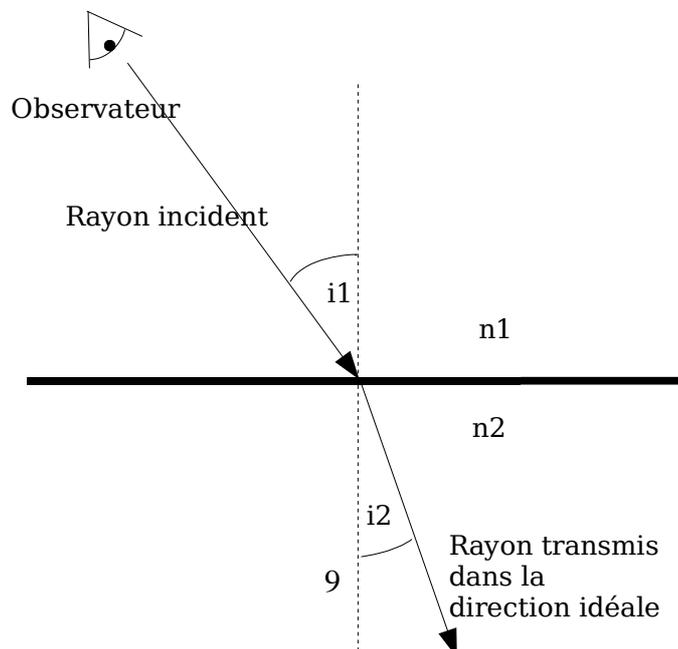
$N$  la normale à la surface et  $I$  le rayon incident.

### 2.2.1.2 Réfraction

La direction de la lumière change au passage d'interface entre deux milieux transparents différents (ex air/eau). Le rayon dévié qui se propage dans le milieu de destination est appelé rayon réfracté ou transmis. L'indice de réfraction d'un milieu caractérise la manière dont le milieu affecte la vitesse de la lumière. Plus l'indice est élevé, plus la lumière est freinée. La correspondance entre les directions du rayon incident et transmis est donnée par la loi de Snell-Descartes:

$$n_1 \sin(i_1) = n_2 \sin(i_2)$$

avec  $n_1$ , et  $n_2$  les indices des milieux respectifs et  $i_1$  et  $i_2$  les angles des rayons



Un rayon réfracté obéit à l'équation:

$$\mathbf{R} = \eta \mathbf{I} + (\eta \alpha - \sqrt{|\beta|}) \mathbf{N}$$

avec  $\alpha = \langle -\mathbf{I}, \mathbf{N} \rangle$

et  $\beta = 1 - \eta^2 (1 - \langle -\mathbf{I}, \mathbf{N} \rangle^2)$

seulement si  $\beta > 0$  (angle limite)

### 2.2.1.3 Coefficient de Fresnel

Une partie de la lumière est réfléchie, et une autre est transmise, lorsque la lumière atteint la surface. C'est le phénomène de Fresnel.

Lors d'observation à la verticale de l'eau, toute la lumière est transmise. A l'inverse toute la lumière est réfléchie lorsque l'on observe en angle rasant. On peut approximer cette loi en simplifiant des équations physiques plus complexes.

Le coefficient de Fresnel spécifie la quantité de rayons réfléchis et vérifie l'équation:

$$c = b + s(1 + \mathbf{I} * \mathbf{N})^p$$

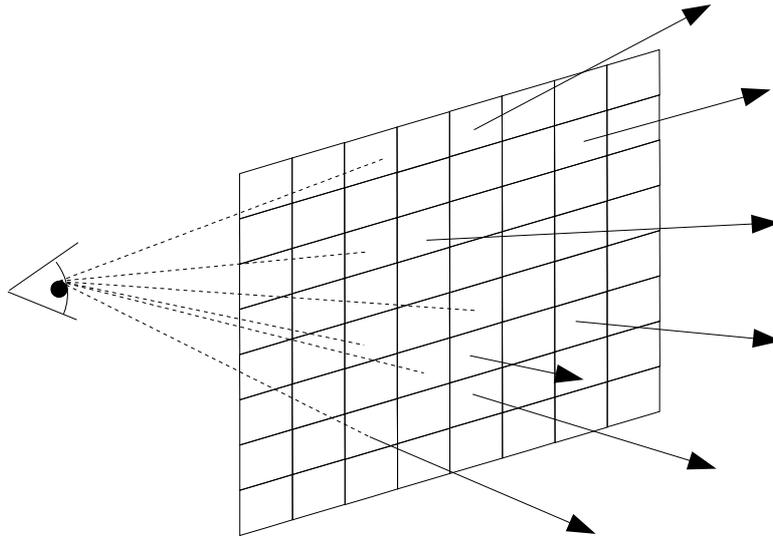
où  $b$  est un décalage,  $s$  un coefficient multiplicatif, et  $p$  une puissance dépendant du matériaux.

## 2.2.2 Lancer de rayon

Il s'agit d'une technique couramment utilisée en infographie. Elle génère des images de haute qualité, et se base sur les lois de l'optique géométrique précitées.

Un rayon de lumière parcourt des chemins rectilignes dans un milieu de même indice. Il suit la loi de réflexion classique et la loi de Snell-Descartes aux interfaces pour la réfraction.

A chaque pixel d'une image est associée une couleur qui est la somme de toutes les couleurs des objets et matériaux rencontrés sur le chemin. On envoie une série de rayons "primaires" dans chaque pixel de l'image. Ceux-ci intersectent ou non des objets qui réfléchissent ou transmettent à leur tour. L'algorithme est récursif et la couleur en un point est égale à la somme des couleurs réfléchies, transmises et d'éclaircissements (cf ci dessous). La récursion s'arrête quand le rayon ne rencontre plus d'objet, que le seuil maximal de suivi est dépassé, ou que son intensité est inférieure à un certain seuil.



Envoi des rayons primaires à travers l'image

Cette technique est donc parfaitement adaptée pour le rendu de milieux transparents et réfléchissant. Le rendu est précis. Mais les calculs d'intersections sont très lourds. Cette technique n'est donc jamais utilisée dans une application qui nécessite des fréquences de rafraîchissement proches des 25 Hertz (rendu en "temps interactif" ou "temps réel").

### 2.2.3 Rendus adaptatifs

Les rendus adaptatifs désignent des techniques de rendu qui s'adapte à la position de l'observateur. Deux types de rendu adaptatifs nous intéressent dans notre étude:

Une première classe de rendu est la simplification de modèles. Un objet n'a pas besoin d'être affiché dans une grande résolution lorsqu'il est vu de loin, car ses détails fins ne sont pas alors perceptibles. Le but est alors de simplifier un modèle en décimant de manière judicieuse des sommets, afin d'en générer plusieurs versions à des résolutions décroissantes. Les difficultés de ce genre de technique résident dans la génération de versions pertinentes et adaptées au niveau de résolution souhaité, et dans la transition entre les versions lors de déplacements, qui doit s'effectuer continuellement. On peut noter que des techniques similaires s'appliquent aux texture ("MIP mapping")

Une deuxième classe traite des transitions entre différents modes de rendu, comme par exemple [14], dans lequel la technique de rendu passe successivement de displacement mapping (cf plus loin), bump mapping (cf ), puis BRDF, en fonction de la profondeur et de l'angle de vision.

Enfin, [15] étend les notions de multirésolution et niveaux de détails en introduisant le concept de hiérarchie de modèle.

### 2.2.4 Environment mapping

L'observation d'un objet parfaitement réfléchissant n'est pas une observation de la surface de l'objet lui-même, mais de la manière dont il reflète son environnement.

Cet environnement est constitué de tous les objets de la scène plus ou moins proches de l'objet,

et l'objet lui-même. La méthode conceptuellement la plus simple pour rendre des objets réfléchissant en infographie est la méthode de lancer de rayon. Elle gère les effets de parallaxe pour les objets de cet environnement, et l'auto-réflexion. Mais elle reste coûteuse compte-tenu des nombreux calculs d'intersection avec les objets de la scène, plus où moins complexes.

Elle peut être simplifiée dans le cas particulier où les objets de la scène sont supposés suffisamment loin pour être considérés "à l'infini" par rapport à l'objet. Il suffit alors de créer une "carte d'environnement" ("environment mapping" en anglais) représentant le monde du point de vue de l'objet, ayant la forme d'une sphère ("Sphère mapping") ou d'un cube ("Cube mapping"). Les rayons sont alors intersectés avec la carte d'environnement uniquement, et les intersections se trouvent alors réduites à des simple intersections droite-sphère ou droite-cube. Le gain de temps de calcul est conséquent.

Le principal inconvénient de cette technique réside dans l'impossibilité d'avoir des effets de parallaxe, ou d'auto-réflexion. Cependant cette méthode est souvent utilisée lorsque le temps de rendu doit être minimisé, et les imprécisions ne sont pas toujours décelables pour un observateur non-averti.

### **2.2.5 Bump mapping**

De part l'utilisation importante de techniques d'interpolation dans le processus de génération d'image de synthèse, les surfaces rendues présentent souvent un aspect lisse et peu naturel, bien différent de ceux de la vie courante, qui présentent de nombreuses imperfections. Or il n'est pas envisageable de modéliser chaque imperfection séparément avec de la géométrie supplémentaire compte-tenu des coûts que cela impliquerait.

La technique de bump-mapping [16] permet justement de résoudre ce problème. Elle consiste à perturber légèrement les normales avant leur utilisation dans le calcul d'éclairage.

L'avantage de cette technique est donc de simuler une géométrie qui n'existe pas, et qui n'aurait presque pas pu exister autrement, et cela à moindre coût. Elle va donc servir dans notre étude pour modéliser les petites ridules.

Cependant cette technique ne permet pas de tenir compte des problèmes d'occlusion pour des angles rasants par exemples, et peut provoquer des phénomènes d'aliasing (cf 3.3.3) lorsque les normales varient trop violemment. Une solution au problème des occlusions est proposée dans [14] grâce à une technique baptisée "redistribution bump mapping", qui consiste à pré-calculer les normales en tenant compte des occlusions pour différents angles de vue.

### **2.2.6 Fonctionnalités des cartes graphiques utilisables pour le rendu de l'eau**

La carte graphique permet de traiter les informations concernant l'affichage afin de les envoyer au moniteur. Initialement, ces cartes servaient uniquement à traiter l'affichage. Avec la généralisation de la 3D, elles soulagent désormais le processeur pour le traitement des calculs 3D, grâce à une architecture entièrement dédiée. De nos jours, certaines des techniques évoquées précédemment ont été implémentées directement dans le hardware des cartes graphiques, et exploitent la puissance de calcul grandissante de ces dernières. La tendance actuelle dans le domaine de la visualisation temps-réel consiste donc à effectuer le maximum de calculs graphiques avec le GPU ("Graphic Processing Unit", processeur de la carte graphique), tant les performances de ces

derniers sont importantes.

### 2.2.6.1 Shaders

Les deux parties principales du pipeline graphique concernent le traitement des sommets ("vertex" en anglais), et le traitement des pixels. Un vertex contient les informations de position, de couleur, et de normale. Il subit des transformations dans le pipeline graphique pour être finalement affiché dans le repère de l'écran. Puis les sommets sont assemblés en primitives (triangle, carré ...) qui sont ensuite texturées et colorées. Les pixels correspondant à l'intérieur de ces primitives sont appelés "fragments" ou plus simplement "pixels". Ces opérations sont appelées opération de shading ("ombrage" en français).

Les premières et deuxièmes générations de cartes graphiques ont un pipeline graphique complètement automatisé. Les opérations de texturage et de colorisation se font par interpolation.

Avec l'arrivée des cartes de troisième et quatrième génération (Geforce 4, Geforce FX), il est désormais possible de programmer ces deux parties du pipeline graphique. De tels programmes sont appelés respectivement "vertex shader" et "pixel shaders" [17].

Cela offre une flexibilité supplémentaire, la possibilité d'implémenter ses propres opérations d'ombrage directement pour le GPU, et de profiter de l'architecture entièrement optimisée pour les traitements 3D. Les gains de calculs sont considérables, comparés à une implémentation des mêmes techniques d'ombrage pour un CPU. Techniquement il est possible d'agir directement sur les normales, les positions et les couleurs des pixels, et donc de court-circuiter le processus de rendu habituel tout en bénéficiant de l'architecture optimisée.

### 2.2.6.2 Carte de hauteur

Pour des surfaces présentant une certaine horizontalité comme les terrains, ou l'eau calme, il est possible de définir une fonction  $f(x, y) = z$  où  $(x, y)$  représente les coordonnées dans un plan moyen, et  $z$  la hauteur de la surface en ce point.

Cette fonction peut être encodée dans une texture qui est alors appelée "carte de hauteur" ou "height map". Elle peut être utilisée ensuite en conjonction avec une grille dont chaque sommet est déplacé de la hauteur correspondante (technique de "displacement mapping").

### 2.2.6.3 Environment mapping

Cette technique fait partie des premières à être implémentée sur les cartes graphiques modernes, et est de nos jours il s'agit d'une fonctionnalité de base de tout GPU.

Il est possible d'utiliser des cubemaps très simplement avec un shader en fournissant une référence des six faces du cube à celui-ci. L'environnement mapping peut être utilisé pour la réflexion mais aussi la réfraction, le calcul des rayons réfléchis et réfractés se faisant selon les formules déjà mentionnées, directement dans le shader lui-même.

Dans [18] la cubemap proposée est découpée en deux, de telle sorte qu'une partie contient la carte de réfraction, et l'autre de réflexion. Cela permet un gain de place mémoire sur la carte graphique.

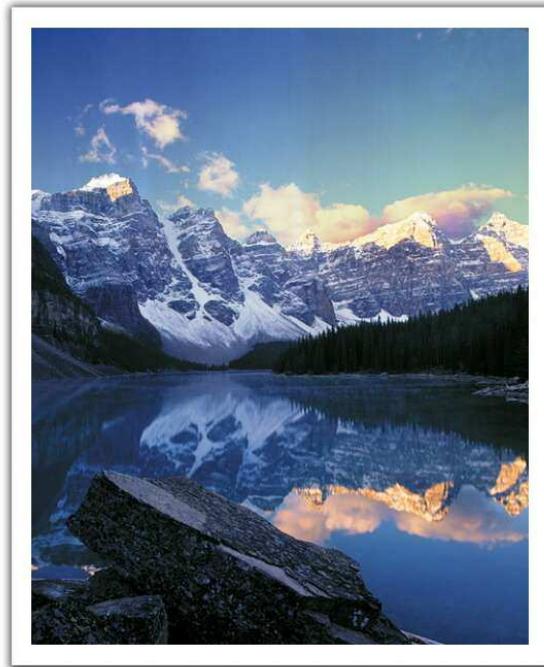
### 2.2.6.4 Reflective bump mapping

Avec les pixel shaders, il est très facile d'implémenter la technique de bump mapping, puisque la normale en chaque pixel  $y$  est accessible directement. La combinaison de cette technique avec une technique d'environnement mapping par réflexion s'appelle le reflective bump mapping, et permet de rendre des surfaces d'aspect métallique très bosselées.[19]

## 3 Notre approche pour le rendu des ruisseaux

### 3.1 Interaction lumière/surface de l'eau

L'objectif est de gérer l'interaction de la lumière avec la surface de l'eau, surtout sur le lieu des ondes, car il s'agit du principal vecteur d'information sur la configuration de l'eau. Nous souhaitons aussi la gérer en temps réel.

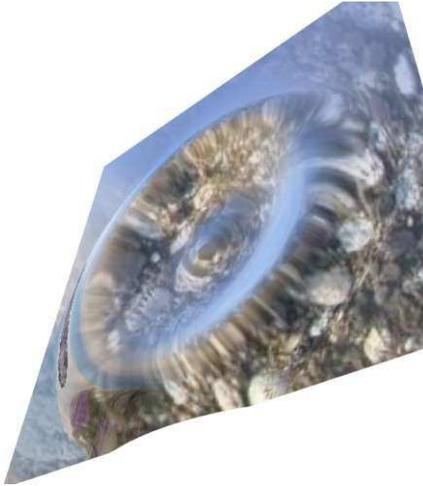


L'utilisation de la technique de lancer de rayon bien que précise, est trop coûteuse pour les raisons expliquées précédemment . La solution la plus efficace que nous ayons trouvée est d'utiliser les trois lois de l'optique géométrique citées précédemment, à savoir la réflexion, la réfraction et le coefficient de Fresnel, et de les implémenter dans un shader, afin de basculer les calculs sur la carte graphique . Les rayons réfléchis et réfractés calculés vont chercher respectivement l'information dans deux cube map différentes, une correspondant à un décor naturel (des montagnes dans la brume), et l'autre, à un fond de ruisseau (des pierres).

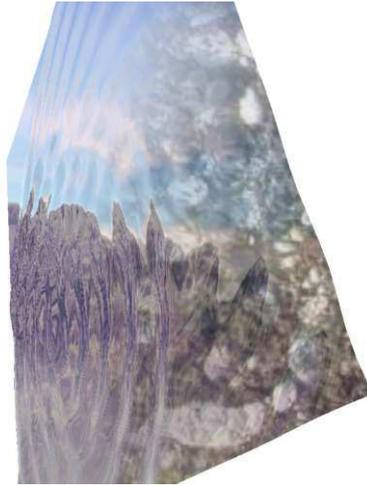
L'approximation sous-jacente de la technique d'environnement mapping reste assez valable pour la réflexion du ciel des montagnes éloignées. Cependant elle l'est beaucoup moins pour la réfraction, compte-tenu du fait que le ruisseau est supposé peu profond, et des problèmes de parallaxe surviennent. D'autre part cette technique ne permet pas d'observer des phénomènes d'auto-réflexion.

Voici ce que nous pouvons obtenir avec des shaders (il s'agit d'une application qui anime les ondes à l'aide de sinus et n'a rien à voir avec [13]):

21Hz

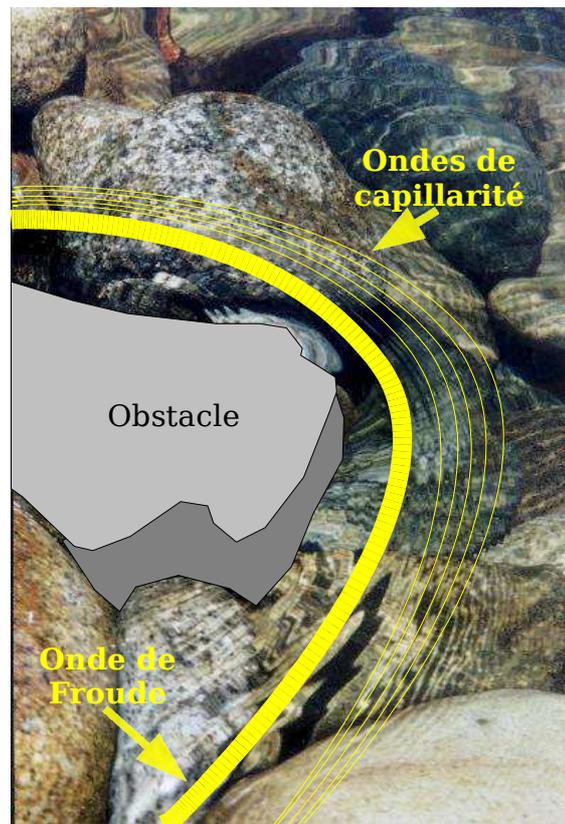


27Hz



## 3.2 Modélisation des ondes

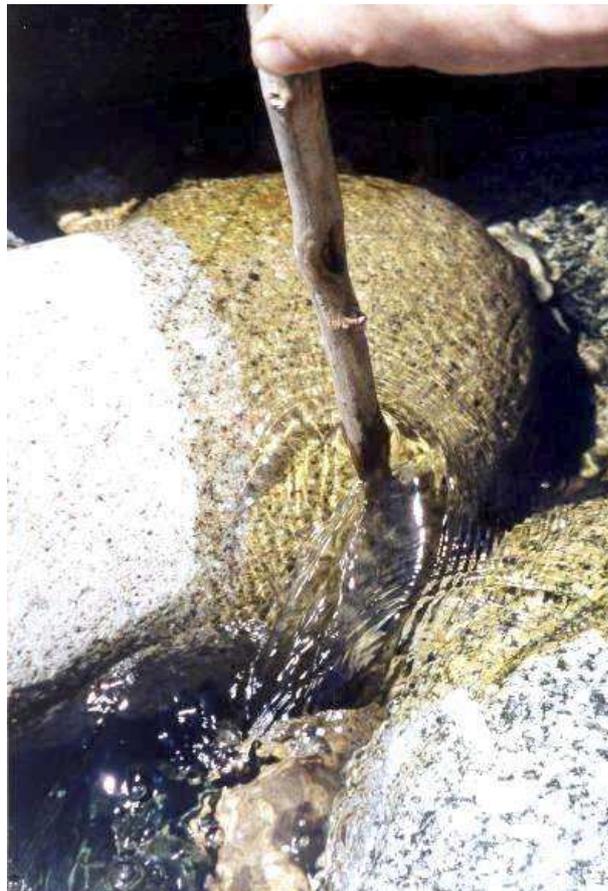
### 3.2.1 Identification des phénomènes à modéliser



Notre approche se base avant tout sur une approche phénoménologique. Il faut donc dans ce cas identifier précisément les phénomènes à étudier. Sur la photo ci-dessous, une onde principale de quelques centimètres de hauteur, appelée onde de Froude, est visible en amont de l'obstacle.

Elle est suivie à une certaine distance de plusieurs ondelettes très fines, que nous qualifierons par la suite "d'ondes de capillarité".

Sur les photos suivantes, deux ondes de Froude s'intersectent. La loi qui régit l'intersection de ces ondes semble être linéaire.



Enfin sur la photo suivante, une multitude d'ondes se superposent et s'enchevêtrent.



Nous pouvons alors prendre conscience que les ondes de Froude sont vraiment des éléments caractéristiques de l'aspect de la surface des ruisseaux.

Nous n'avons pas trouvé d'information sur la distance entre l'onde de froude et la première onde de capillarité, ainsi que sur le nombre d'ondelette à générer. Nous laisserons donc ces valeurs arbitraires. D'autre part ces ondes s'affaiblissent progressivement dans la direction de l'aval. Là encore nous utiliserons un affaiblissement arbitraire.

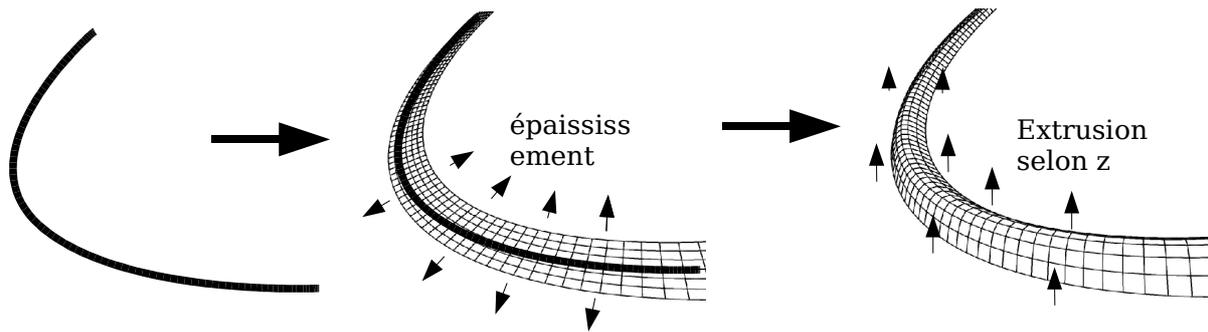
Nous avons à notre disposition la méthode de construction géométrique de [13] qui fournit les résultats vectoriels en temps réel de la position des ondes principales de Froude. Il s'agira donc dans un premier temps de créer un maillage en 3D à partir de ces informations en 2D.

### **3.2.2 Géométrie des ondes**

L'objectif est de trouver une méthode rapide pour créer un maillage à partir des informations 2D obtenues par la construction géométrique de la méthode de [13], et qui permette également de générer différents modèles plus ou moins simplifié d'une même onde, en vue d'un rendu adaptatif. Le profil des ondes devrait ressembler à une gaussienne ou un cosinus.

Une première solution pourrait être d'utiliser les surfaces implicites, qui sont des iso-surfaces de potentiels, car elles offrent des résultats visuellement bien adaptés pour des phénomènes fluides. Mais cela reste lourd à mettre en place et très coûteux.

La solution que nous proposons donc ici, est purement géométrique. Il s'agit d'épaissir la ligne dans le plan selon des normales interpolées à chaque segment de la ligne. Puis de déplacer les sommets ainsi créés selon  $z$ , de manière à obtenir le profil souhaité.



Il faut préciser que cette méthode présente quelques limites. En effet le rayon de courbure de l'onde ne doit pas être supérieur à l'épaisseur de l'onde, sinon la géométrie se chevauche, et produit des aberrations d'affichage. Dans la pratique, ce cas limite est peu probable, compte-tenu de la faible épaisseur de l'onde.

Ce maillage est indispensable pour rendre les effets d'occlusion, lorsque l'observateur est proche de l'onde. Les autres paramètres importants à contrôler sont les normales, qui donnent une information visuelle sur la courbure de la surface. Nous allons voir que la création du maillage est subordonnée, à priori, à celle des normales. Nous effectuerons dans la suite quelques essais, pour s'assurer du résultat visuel final.

### 3.2.3 Courbure des ondes

#### 3.2.3.1 Représentation des normales à la surface des ondes

L'information de courbure des surfaces réfléchissantes est entièrement donnée par les normales. Il est donc important de représenter de manière précise l'information de courbure, car l'aspect visuel est très sensible aux fluctuations des normales lors de réflexions.

Dans la pratique, une normale est assignée pour chaque vertex, et la carte graphique crée des normales interpolées linéairement par défaut. Cela peut poser problème dans certains cas. Par exemple, si le nombre de sommets est insuffisant.

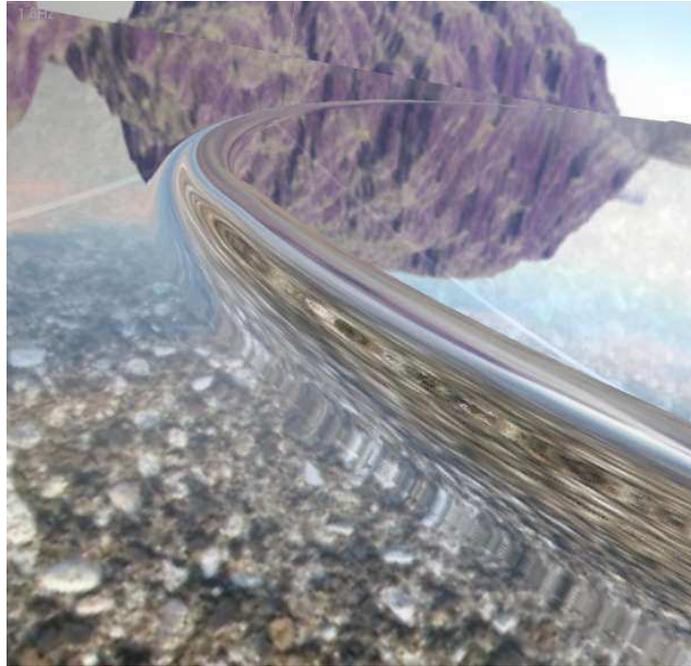
Si l'on place des sommets aux extrema d'une fonction sinusoïdale par exemple, et l'on assigne des normales de même direction alors la normale interpolée ne va pas varier alors que la courbure change brutalement. Nous sommes donc confrontés à un problème classique d'échantillonnage de signal. La solution est alors d'augmenter l'échantillonnage, c'est-à-dire d'ajouter des sommets intermédiaires afin d'avoir une variation plus conforme des normales. Mais cette introduction de sommets a un coût supplémentaire important, lors du traitement de la géométrie par la carte graphique.

Il est cependant possible de stocker des normales dans une "normal map", et d'éviter ainsi l'introduction de nouveaux sommets. Cette "normal map" est alors utilisée ensuite de manière similaire à la carte de perturbation dans la technique du bump-mapping. Grâce aux pixels shaders, ces calculs peuvent être facilement implémentés pour la carte. Il suffit de fournir la texture correspondant à la normal map, et de récupérer pour chaque pixel l'information de normales correspondante.

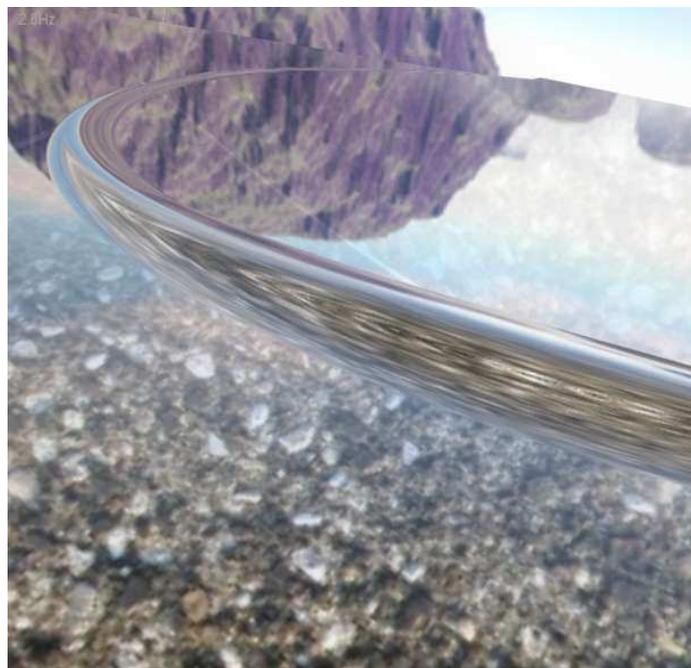
La méthode que je propose ici est légèrement différente, et dévoilera son intérêt un peu plus tard. Je propose de représenter la surface du ruisseau par une carte de hauteur. Cela est rendu possible parce que l'eau est supposée plane. Pour obtenir la normale en chaque pixel, nous

utiliserons une technique décrite plus loin, basée sur un échantillonnage local de la carte de hauteur.

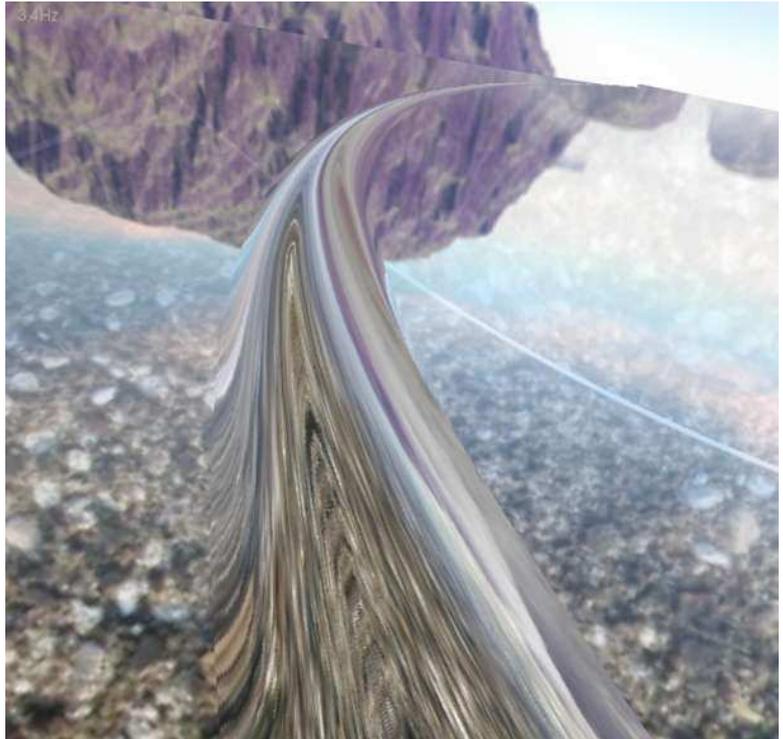
Nous pouvons maintenant comparer quelques résultats. Pour cela, nous testons successivement un profil triangulaire, puis sinusoïdal, et une gaussien:



*triangle*

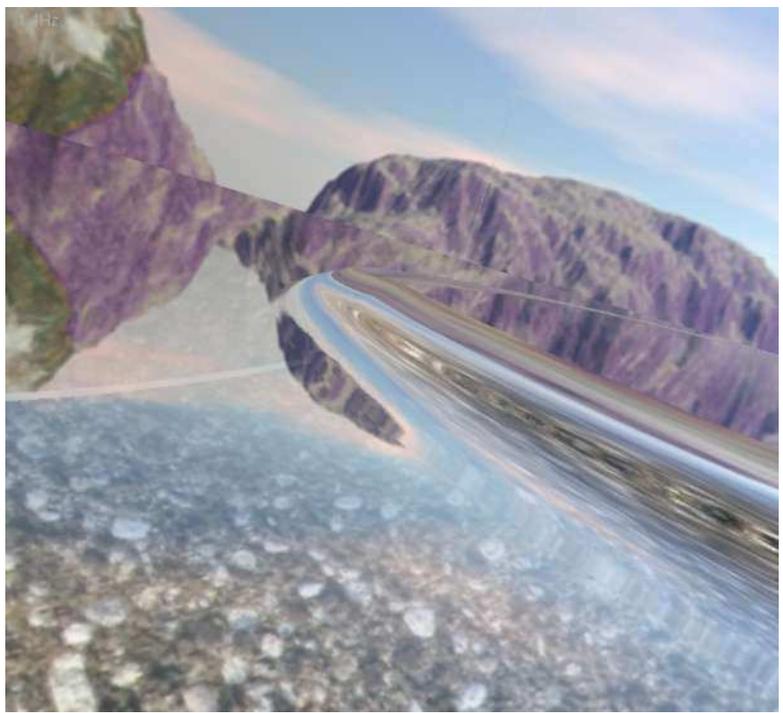


*cosinus*

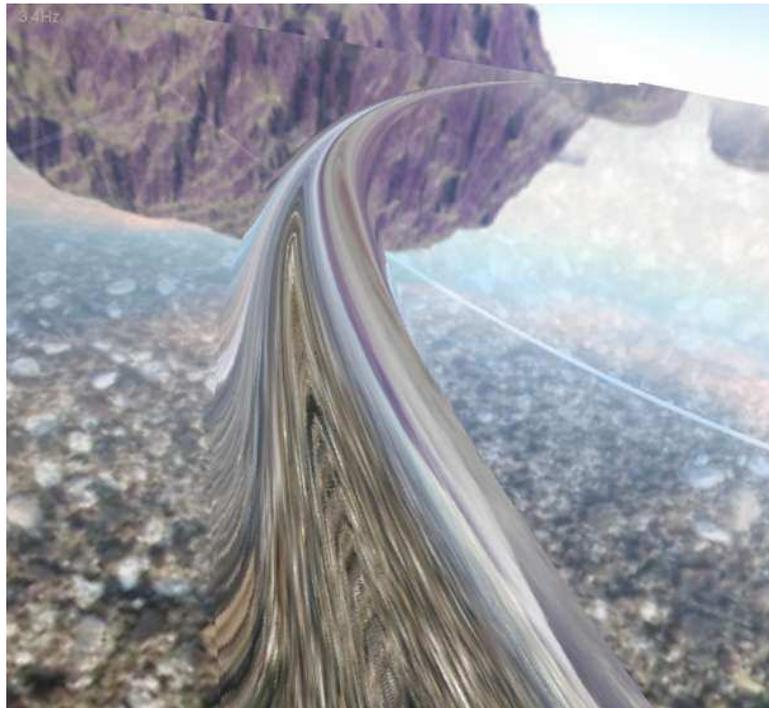


*gauss*

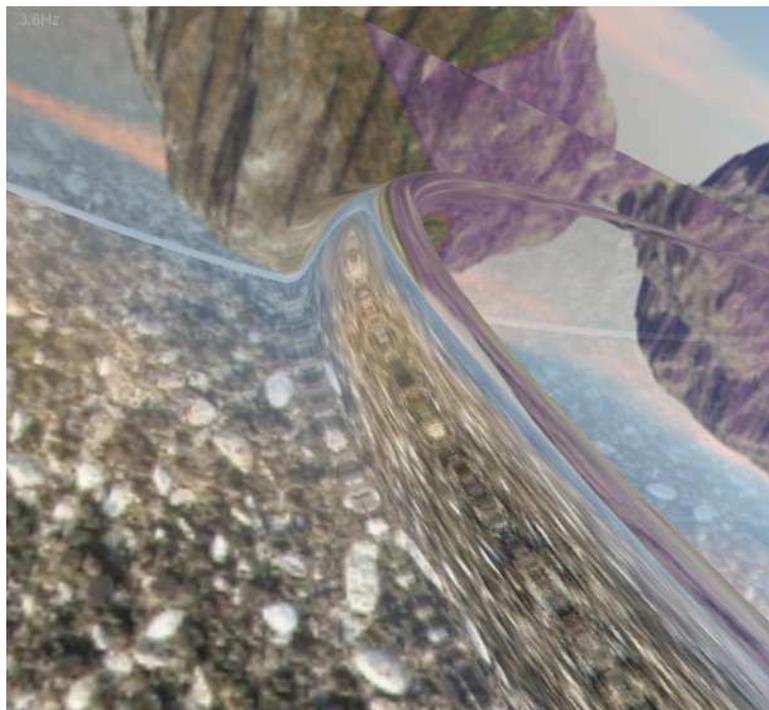
Puis nous modifions le nombre de lignes d'extrusion.



*2 ligne d'extrusion*



*5 lignes d'extrusion*



*30 lignes d'extrusion*

On observe que la forme et la résolution du maillage est peu importante, et reste toujours crédible même avec une résolution faible.

On observe également une apparition d'artefacts (facettes discontinues) de près dans le cas d'interpolation linéaire simple. La technique par carte de hauteur est plus précise, plus coûteuse, mais reste cependant raisonnablement utilisable

Enfin lorsque les ondes bougent, ou que l'observateur s'éloigne, la géométrie exacte des ondes est difficile à percevoir, compte-tenu de la complexité visuelle déjà obtenue.

### **3.2.3.2 Exploitation de l'information d'une carte de hauteur**

L'objectif est d'exploiter correctement la carte de hauteur pour en extraire les normales. La solution consiste à utiliser une technique de différence finie, et échantillonner, pour chaque pixel, trois points dans la carte de hauteur. Le premier point est situé sur le pixel, et les deux autres sont décalés d'un pas  $dx$  et  $dy$  selon les  $x$ , et  $y$  respectivement dans le repère de la carte de hauteur. Le choix des paramètres  $dx$  et  $dy$  est crucial. Ils doivent être de l'ordre de la moitié de la largeur du signal (loi de Nyquist).

Nous construisons alors deux vecteurs, correspondant aux vecteurs tangent et binormal. Par produit vectoriel de ces deux vecteurs, on obtient ainsi la normale à la surface en ce point.

Cette méthode de calcul de la normale est encore une fois implémentée directement dans un pixel shader, afin d'accélérer à nouveau les calculs.

### **3.2.3.3 Avantages d'une représentation par carte de hauteur**

Une carte de hauteur centralise et compresse la représentation interne d'une onde, car elle est utilisable pour obtenir la hauteur d'un point donné, ou la normale en fonction des besoins. Cela en fait une représentation adaptée à la multirésolution.

De plus l'addition d'onde devient aisée par simple addition scalaire des hauteurs.

Une carte de hauteur étant représentée par une texture, nous avons la possibilité de la pré-calculer, et de la stocker sur la carte.

Pour le moment, les vertex shaders ne permettent pas de récupérer l'information d'une texture. Nous ne pouvons donc pas utiliser la technique de displacement mapping pour chaque sommet (cf 2.2.6.2). Mais cette fonctionnalité sera bientôt disponible sur les prochaines cartes graphiques. Il sera donc possible de transférer la totalité des calculs de transition sur le GPU.

## **3.3 Rendu adaptatif**

### **3.3.1 Identification des différentes résolutions**

Dans notre étude, nous souhaitons une liberté du mouvement de l'observateur, afin qu'il puisse observer des phénomènes compris dans un large domaine spatial. Nous souhaitons également que la fréquence de rafraîchissement de l'image soit proche des 25 Hertz.

Nous avons vu qu'un objet n'a pas besoin d'être affiché avec une grande résolution lorsqu'il est loin. Dans notre étude, nous avons choisi d'utiliser deux niveaux de représentation. Lorsque l'objet est petit (loin), il est représenté par la technique de bump mapping. Lorsqu'il est grand (proche), il est représenté par un maillage.

La transition entre ces deux mode de représentation doit se faire de manière continue et dynamiquement lorsque l'observateur se déplace dans la scène.

### 3.3.2 Simplification et transition de la géométrie

L'objectif est toujours de diminuer le nombre de sommets à traiter, pour les raisons déjà expliquées.

Nous sommes dans un cas singulier où la géométrie n'est surtout utile que pour l'occlusion, (profil peu important visuellement), et les normales sont porteuses de l'information principale. Les algorithmes élaborés de simplification de modèle ou de multirésolution classique, sont donc peut-être trop complexes ici .

Nous proposons de délimiter deux niveaux de résolution: Au premier plan, les ondes sont entièrement représentées par de la géométrie ( nombre de ligne d'extrusion constante). En arrière plan, elles sont représentées par les deux lignes d'extrusion extrêmes, et forment donc une surface plane dont l'intérieur est rendu grâce à la technique du bump mapping.

La transition entre deux niveaux est délimités par une droite de transition située à une distance constante de l'observateur. Les hauteurs des sommets du maillage décroissent de manière linéaire en fonction de la distance au delà d'un seuil.

Cette transition dépend donc de la distance à l'observateur qui doit être fonction elle-même de la grandeur du phénomène (onde de Froude ou capillaire): La distance de transition est petite pour les ondes de capillarité, et plus grande pour les ondes de Froude.

La transition peut dépendre également de l'angle d'observation. En effet si l'observateur est en angle rasant, le profil de l'onde devrait être assez visible et provoquer des occlusions. En revanche si l'angle est normal, il n'y a pas d'occlusion.

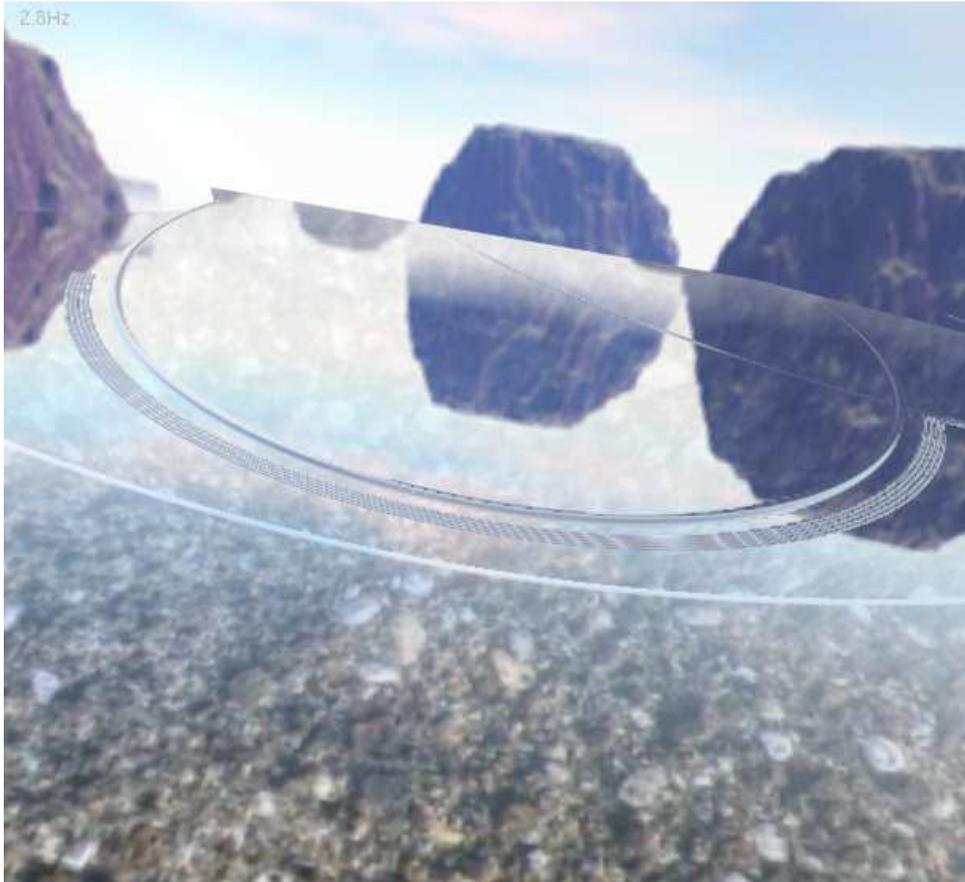
Dans nos essais, l'angle d'observation n'a pas été expérimenté, mais il est toujours possible de s'inspirer de la formule de [14]

Il est à noter que l'on pourrait envisager quelques raffinements comme par exemple diminuer le nombre de ligne d'extrusion des ondes en fonction de la distance, pour obtenir une transition plus douce de la géométrie.

### 3.3.3 Le problème de sous-échantillonnage

Chaque pixel porte l'information sur une zone de la scène dont la taille est proportionnelle à la distance. Un pixel ne couvre donc pas la même zone de la scène en fonction de la distance. L'envoi d'un seul rayon à travers un pixel représentant une partie éloignée de la scène, conduit à un sous-échantillonnage de l'environnement, et donc des variations discontinues de couleur: En effet, il faudrait idéalement tenir compte de la totalité de la surface visible à travers ce pixel.

Ce phénomène de sous-échantillonnage est un exemple d'un phénomène général appelé aliasing. En particulier, nous pouvons observer des figures régulières, appelées figures de Moiré, témoignant d'un sous-échantillonnage. Cette disposition si caractéristique des figure de Moiré provient du fait qu'un signal sous-échantillonné est reconstruit avec une fréquence inférieure à celle du signal initial.



*Exemple de figure de Moiré sur les ondes de capillarité*

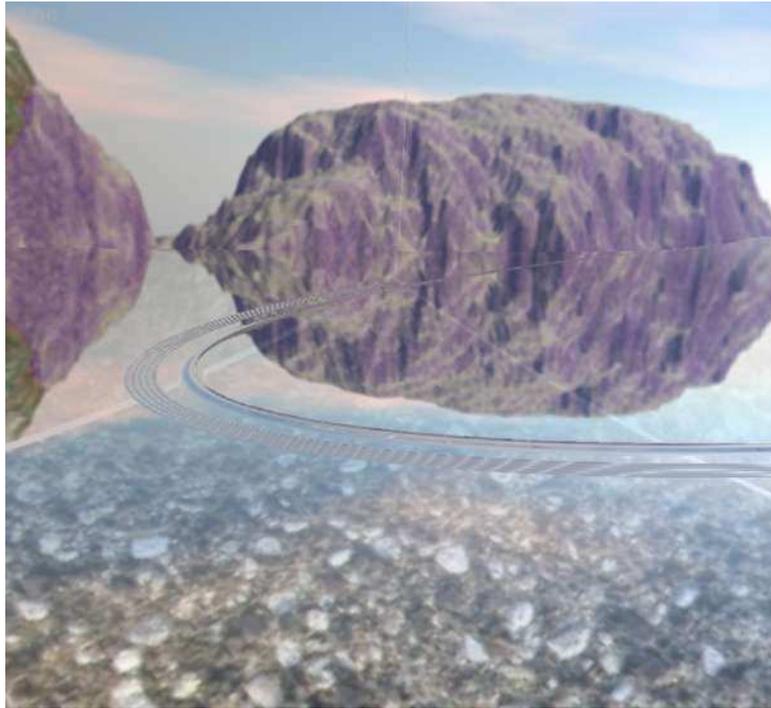
Nous allons voir par la suite comment supprimer ces problèmes d'aliasing.

### **3.3.4 Transition des normales**

Pour obtenir une meilleure qualité d'image, il faut donc supprimer ces phénomènes d'aliasing, et plusieurs solutions sont envisageables:

Les textures d'environnement peuvent être pré-filtrées. Sur les cartes graphiques modernes la génération automatique de différentes versions simplifiées d'une même texture est possible. Une sélection de la bonne version de texture en fonction de la distance est alors automatiquement effectuée (technique de "MIPmapping"). Celle-ci est similaire aux sélections de version de modèle dans les rendus adaptatifs de modèle.

Nous avons choisi de faire varier les informations des cartes de hauteur de manière similaire à la variation utilisée pour la géométrie: Au delà d'un seuil, les hauteurs de la carte diminuent linéairement pour s'aplatir complètement. Ainsi les normales varient moins rapidement au fur et à mesure que l'on s'éloigne, jusqu'à atteindre une position parallèle à l'axe des  $z$ . Les ondes "s'éteignent" alors progressivement en fonction de la distance. En réglant correctement la distance de disparition des ondes, de telle sorte qu'elles ne soient plus visibles lorsque les premières figures de Moiré apparaissent, on supprime alors simplement l'aliasing. Évidemment il faudrait idéalement tenir compte quand même des ondes, mais cela nécessiterait des techniques particulières et souvent coûteuse, donc inappropriées à notre cas d'étude.



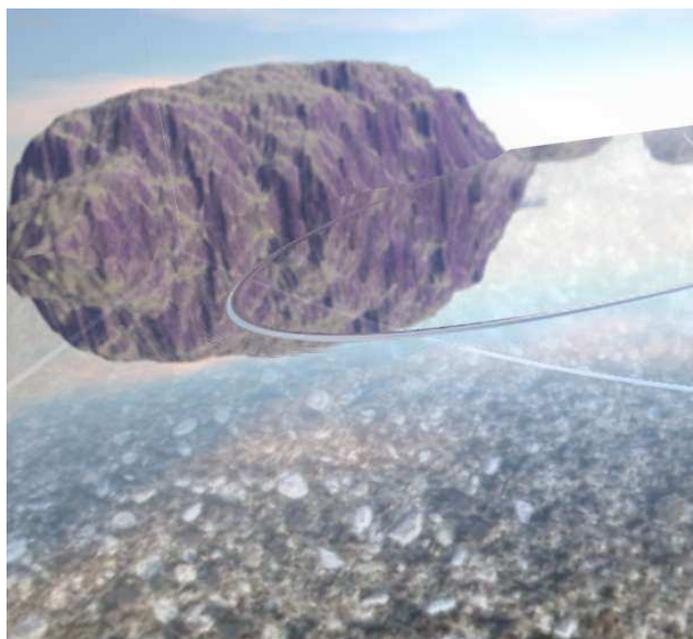
On observe les figures de Moiré sur les ondes de capillarité.



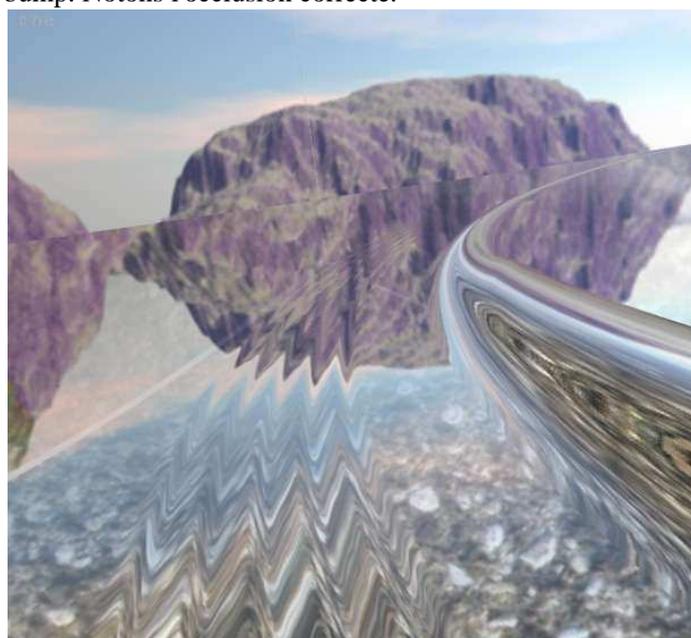
Les figures de Moiré ont disparu, grâce à la décroissance linéaire de la carte de hauteur.

## 4 Résultats

Voici une première vue de loin, l'onde est rendue en bump:



Voici un vue approchée, l'onde principale est rendue avec de la géométrie, les ondes de capillarité sont en bump. Notons l'occlusion correcte:



Nous avons donc utilisé et coordonné plusieurs techniques existantes en synthèse d'image. L'interaction de la lumière avec la surface de l'eau a été traitée directement par la carte graphique à l'aide de shader. Le maillage des ondes, a été construit par une technique simple d'épaississement/extrusion. Les informations sur les normales ont été stockées dans des cartes de hauteurs afin d'avoir une représentation compacte et facilement utilisable, pour une technique de rendu adaptatif. Nous avons choisi d'effectuer une transition linéaire entre une modélisation par géométrie pure (pour les ondes proches de l'observateur) et par bump-mapping (pour les ondes situées au loin). En modifiant de manière adaptée le paramètre de distance de cette transition, nous avons pu éviter les apparition de phénomènes d'aliasing ou de figure de Moiré, à moindre

coût.

On obtient des fréquences d'affichage proches de 10 Hertz. A L'avenir des optimisations manuelles de codes pourraient être effectuées et augmenteraient certainement cette fréquence de rafraîchissement.

## **5 Conclusion et Perspectives**

L'apport de cette étude réside principalement dans la coordination et l'utilisation de différentes techniques pré-existantes dans le domaine du rendu des images, et de l'utilisation particulière des fonctionnalités des cartes graphiques modernes, pour visualiser les ondes à la surfaces des ruisseaux, de manière interactive.

L'utilisation originale des cartes de hauteurs nous a permis d'avoir une représentation compacte des normales, qui s'en trouvent ainsi aisément manipulables et sommables lorsque des ondes s'intersectent. Cette représentation facilite l'utilisation de technique de rendu adaptatif, et permettra à terme de transférer l'intégralité des calculs de transition entre différentes résolutions sur le GPU. Enfin nous avons obtenu des images d'assez bonne qualité, avec des vitesses d'affichage qui permettent une bonne interactivité.

Il serait intéressant de compléter cette étude, en étudiant en détail les intersections des ondes. D'autre part plusieurs autres phénomènes pourraient à l'avenir être modélisés comme par exemple les tourbillons, l'écume ou les sillages de von Karman.

Enfin, nous espérons avoir démontré dans cette étude la pertinence de l'étude des ondes de Froude dans la modélisation des ruisseaux, et plus généralement, de l'intérêt d'une approche phénoménologique dans la modélisation de phénomènes naturels.

# Bibliographie

- [1] Michael Kass and Gavin Miller. *Rapid, stable fluid dynamics for computer graphics*. In Forest Baskett, editor. Computer Graphics (SIGGRAPH '90 Conference Proceedings), volume 24, pages 49-57. August, 1990.
- [2] Nick Foster and Demetri Metaxas. *Realistic animation of liquids*. In Wayne A. Davis and Richard Bartels, editors, Graphics Interface '96, pages 204-212. Canadian Information Processing Society, Canadian Human-Computer Communications Society, May 1996.
- [3] Jos Stam. *Stable fluids*. In Alyn Rockwood, editor, Proceedings of the Conference on Computer Graphics (SIGGRAPH '99), pages 121-128, N.Y. ACM Press. August 1999.
- [4] M.Carlier. *Hydraulique générale et appliquée*. Eyrolles,1980.
- [5] Chassaing P.. *Mécanique des fluides. Eléments d'un premier parcours*. Cepadues éditions,1997.
- [6] Jörn Loviscach. *Complex water effects at interactive frame rates*. Journal of WSCG 11, pp. 298–305 (2003).
- [7] Gary A. Mastin , Peter A. Watterberg , John F. Mareda. *Fourier synthesis of ocean scenes*. IEEE Computer Graphics and Applications, v.7 n.3, p.16-23, March 1987.
- [8] Ken Perlin. *An image synthesizer*. ACM SIGGRAPH Computer Graphics, v.19 n.3, p.287-296, Jul. 1985.
- [9] Ken Perlin and Fabrice Neyret. *Flow noise: textural synthesis of animated flow using enhanced Perlin noise*. In SIGGRAPH 2001 Technical Sketches and Applications, August 2001.
- [10] Tessendorf J. . *Simulating ocean water*. In Siggraph Course Notes, Addison-Wesley, 1999.
- [11] Alain Fournier , William T. Reeves. *A simple model of ocean waves*. Proceedings of the 13th annual conference on Computer graphics and interactive techniques, p.75-84, August 1986.
- [12] Darwyn R. Peachey. *Modeling waves and surf*. Proceedings of the 13th annual conference on Computer graphics and interactive techniques, p.65-74, August 1986.
- [13] Fabrice Neyret , Nathalie Praizelin. *Phenomenological simulation of brooks*. Proceedings of the Eurographic workshop on Computer animation and simulation, p.53-64, September 02-03, 2001, Manchester, UK. <http://www-imagis.imag.fr/Publications/2001/NP01>.
- [14] Barry G. Becker and Nelson L. Max. *Smooth transitions between bump rendering algorithms*. International Conference on Computer Graphics and Interactive Techniques. Proceedings of the 20th annual conference on Computer graphics and interactive techniques. Pages: 183 - 190. Year of Publication: 1993.
- [15] James T. Kajiya. *Anisotropic reflection models*. ACM SIGGRAPH Computer Graphics archive. Volume 19 , Issue 3 (July 1985). Pages: 15 - 21. ACM Press New York, NY, USA. 1985.
- [16] Blinn J.. *Simulation of wrinkled surfaces*. Proceedings of the 5th annual conference on Computer graphics and interactive techniques. Pages: 286 - 292. 1978.
- [17] Kilgard Mark J. and Fernando Randima. *The CG tutorial*. Addison-Wesley.
- [18] Jens Schneider and Rüdiger Westermann. *Towards Real-Time Visual Simulation of Water Surfaces*. Scientific Visualization & Imaging Group. Aachen University of Technology.
- [19] Cass Everitt. *Reflective Bump Mapping*. Nvidia Corporation. White paper..