PROGRAMME JEUNES CHERCHEUSES ET JEUNES CHERCHEURS

EDITION 2010

Projet SimOne

DOCUMENT SCIENTIFIQUE

Acronyme	**SimOne**		
Titre du projet en français	Modèles scalables pour l'exploration interactive de la nature à l'échelle de la Terre		
Titre du projet en anglais	Scalable Interactive Models Of Nature on Earth		
Comité d'Evaluation référence (CE)[1]	SIMI 2		
Aide totale demandée		**Durée du projet**	36 mois

SOMMAIRE

[1] Indiquer la référence du CE choisi pour l'évaluation du projet (cf. tableaux page 3 et 4 du texte de l'appel à projets)

1. Contexte et positionnement du projet / Context and positionning of the proposal

The overall goal of the project is to provide representations and algorithms for the real-time navigation, on consumer hardware, in a realistic and plausible **virtual Earth model**. We target the rendering of terrain, vegetation, water surfaces and clouds (we exclude human artefacts), all highly detailed at all scales from ground to space, with physically based motion and illumination at all scales, and without visible transitions between scales. We do *not* target the best possible physical accuracy as in radiative transfer models or computational fluid dynamics methods (like for instance in remote sensing, climate modelization, meteorology, etc). Instead, we target physical *plausibility*, i.e., shape, illumination and motion models that *look* realistic and are efficient enough for real-time applications. Our main goals are:

- **Scalability**: we want to show a virtual Earth at scales varying between millimeters and thousands of kilometers. In this context storing, transferring, computing, rendering, simulating and controlling 3D models are all fundamentally impacted. The representation of the shape, of the motion and of the illumination of objects must be adapted with scale. For instance multi-scale physical models of light transport for terrains, vegetation, water surfaces and clouds are required, as well as multi-scale physical models of motion for ocean waves and clouds.
- **Visual quality and realism**: we target not only physically plausible and detailed shape, motion and illumination of all elements at all scales, but also seamless transitions between scales.
- **Real-time** performance on consumer hardware: this is required for most applications of a virtual Earth model, movies excepted.

Many international companies and actors are competing in this field, including Google (Google Earth), Microsoft (Bing Maps 3D, Flight Simulator X), Nasa (Nasa WorldWind), Crytex (Crysis). Several French companies and actors are also present in this field, including IGN (Geoportail), RSACosmos (planetariums), VWorld (virtual globe). Several research projects also addressed parts of the problem (including ANR projects like Vertigo, Prodige and Natsim - see next section for details). However, no existing products or research results can fully achieve the above goals. Indeed, existing products and research results either:

- don't target real-time performance: this is the case for movie special effects, and for synthetic world builders like Terragen, MojoWorld, Bryce, and Vue d'Esprit.
- limit the navigation area: for instance recent games such as Crysis achieve very realistic renderings in real-time. But they restrict players to move on a small area (missions or levels of a few square kilometers), and often to stay at the ground level.
- limit the amount of details: many applications provide details at only one scale. For instance flight simulators do not provide many details for ground views outside airports.
- don't target or achieve realism: 3D maps applications like Google Earth, Nasa WorldWind and Geoportail use unrealistic illumination models for the terrain, the atmosphere or clouds. The lack of vegetation is also unrealistic. Even applications targeting realism (such as Celestia, a virtual Universe model including a virtual Earth) do not provide fully realistic illumination models, due to real-time constraints.
- don't target animated elements like water and clouds.

Hence, applications that could benefit from the scientific breakthroughs we want to address are numerous: 3D maps such as Goole Earth, simulators for planes, boats, trains or cars, video games, impact studies, virtual Earth model for planetariums, enriched meteo presentation, etc.

2. Description scientifique et technique / Scientific and technical description

2.1. État de l'art / Background, state of art

2.1.1 State of the art in industry

Numerous tools used for movies like Avatar, and dedicated world builder products (Terragen, Vue d'Esprit, MojoWorld, Bryce, etc) can generate large and detailed virtual worlds. But they do not target real-time at all: rendering complex scenes requires hours, several orders of magnitude more than the tens of milliseconds required for real-time.

Video games like Fuel and Crysis (probably the state of art in its domain) display very realistic outdoor environments in real-time. In Crysis the player can navigate in real-time in a whole island with highly detailed and realistic terrain, vegetation, waving and flowing water. However, transitions artifacts are visible between levels of detail (both on shape and illumination), flowing water and clouds are not very realistic, ocean waves are limited to the shore, and the whole environment is restricted to a few square kilometers (ten thousands for Fuel, but with less details). Outside this domain (for example in the associated game level editor, where users can move anywhere, at any speed) visible transitions between levels of detail become very visible.

Many civil, military and game simulators exists for planes, boats, trains and cars (Thales TSS, Corys, Flight Simulator X, Need for Speed, etc). They generally limit the environment to a given area, (except for flight simulators). When foreground details are provided, rough levels of detail are generally used, showing transition artifacts between levels (objects commonly appear suddenly when the user is close enough). Clouds are not 3D and the most realistic ones are not evolving and are not realistically illuminated (even in flight simulators).

3D maps applications like Google Earth, Bing Maps 3D, Nasa WorldWind and Geoportail display a virtual Earth model in real-time, at all scales from ground to space. They are highly scalable and can display many kind of geographic data. However, the transitions between level of details are very visible (new details appear suddenly when they have been downloaded over the network). Although details as small as 50cm can be visible, they are simply projected on the 3D terrain (which is much less detailed), giving a flat unnatural look in perspective views. No vegetation or other 3D details are provided at ground level, except coarse buildings in some areas. Overhangs and arches are an issue. Ocean, clouds and the atmosphere are displayed with a very basic shading that does not look realistic.

In summary, applications providing highly realistic renderings are restricted to small domains and limited user movements, while those providing full navigation capabilities suffer from rendering and animation problems. There is not currently a virtual Earth model with real-time highly realistic rendering and animation, where users are free to move and look anywhere. This is due to several scientific locks that we want to address in this project. Such a virtual Earth model would be useful for more realistic Earth browsers and flight simulators, and for realistic games on very large terrains with unrestricted users movements.

2.1.2 State of the art in academic research

Our goals in this project pose several **scientific challenges**. These challenges are linked to the need for shape, motion and illumination models that can scale seamlessly from millimeters to thousands of kilometers. We recall here that we do *not* target the best possible physical accuracy as in radiative transfer models or computational fluid dynamics methods (like for instance in remote sensing, climate modelization, meteorology, etc). Instead, we target physical *plausibility*, i.e., shape, illumination and motion models that *look* realistic and are efficient enough for real-time applications. This is a different but equally challenging problem.

Scalable shape models. Some shapes must be reproduced with high accuracy (like the Earth topography), while others can be plausible without being exact, like vegetation, water waves, clouds, etc. The first ones require mass of data that largely exceed the computer or graphics card memory. Many representation, compression, storage, loading and streaming techniques have been proposed to extract on the fly the data for a given viewpoint, from a huge but slow storage. The terrain specific algorithms [PH93, Paj98, DWS+97, LH04, AH05, GMC+06] are now mature and provide continuous transitions between the discrete levels of detail (using morphing, still visible on silhouettes). Shapes that can be plausible without being exact can be generated on the fly from a small number of parameters and prior knowledge, using procedural techniques [EMP+94]. Several algorithms using fractals, L-systems, noise functions, point distributions, etc have been proposed for terrains [LH04], grass [GPR+03, SKP05, BPB09], plants and trees [WP95, DCSD02], plant distributions [DHL+98, LP02, AM09], forests [DN04, FMU05], clouds [HL01, NDN96, Ney97, REK+04, BNL06, BNM+08], etc. The first challenge is to provide *scalability*: many algorithms address only one scale (e.g., an individual tree or a forest, an individual cloud or a cloud layer). Addressing many scales often requires to use different representations for the different scales, adapted to the apparent size. The second challenge is then to provide *seamless shape transitions*. This is a very general scientific lock in Computer Graphics and visualization that only few papers address [GPR+03, BPB09].

Scalable illumination models. The appearance of objects results from the incoming light, from the object's reflection properties, and from its shape (responsible for self shadowing). All scales are involved up to microns, but for efficiency reasons we do not want to consider elements smaller than a pixel. We then need to average the illumination contributions of all the subpixel details, up to microns (using one sample per pixel gives aliasing and flickering; using a small number of samples – i.e., oversampling – improves the result but only shifts the problem). Averaging the contributions of micro-scale details ("micro-facets") for surfaces with known statistical properties yields Bidirectional Reflection Distribution Functions (BRDF) models [CT81, Bli82, Kaj85, HTSG91, War92, Sch94, AS00]. But these models are only valid for one scale. At larger scales we need to average the contributions of macroscopic details (e.g., leaves in a tree, trees in a forest), each with its own

BRDF, orientation, incoming light, shadows, etc (which is much more complicated than averaging the shape alone). This is again a hard scientific lock that only few papers address [Fou92, BM93, ON94, TLQ+05, HSRG07, TLQ+08, BNH10].

Computing the outgoing light requires to know the incoming light, which includes the light outgoing from other objects. This leads to global illumination, a field which has been hugely investigated for indoor environments. Still, global illumination remains costly, even when precomputations are used [SKS02]. And global illumination is even more complicated in outdoor environments with very long range interactions. Only few papers addressed this issue for specific cases (terrain [OS07], trees [HPAD06, BBP08], cloud layers [BNL06]) and limited scale. Much work remains to extend this to Earth scale.

Scalable animation models. Our goal is to get water (ocean, rivers) and clouds animated in physically plausible ways, at all scales. Their motion results from physical equations that have been simulated with grids (Eulerian methods like [Sta99, DKY+00]), particles (Lagrangian methods like [PTB+03]), or a mix of the two. The challenge here is again the scalability and the real-time constraint (instabilities due to numerical precision errors have been solved [Sta99], at the price of accuracy and energy dissipation). Eulerian methods can use non uniform grids or multiple grids at different resolutions, which can provide more resolution in regions of interest (finer grid over France for meteorology, near a plane wing for aerodynamic studies). Lagrangian methods, less frequently used, adapt the size and density of particles to their energy, apparent size or location [APKG07]. Despite these techniques, real-time fluid simulations are still limited to small domains. Representing high-resolution evolving clouds in real-time is thus challenging, and doing it in a scalable way is a hard scientific lock.

A possible approach to the problem is to simulate large scale motions by using “macroscopic” physics in addition to the Eulerian and Lagrangian approaches. Indeed large scale phenomena such as river flows, ocean waves and the evolution of air masses can be described with “macrophysics” like hydraulics, wave theory, frontology, etc. Another idea is to use procedural approaches to simulate simple motions and motion details. Some algorithms have been proposed for that [PN01, Che04, BHN07, KTJG08, NSCL08, SB08], but they do not scale to Earth ranges (or address only the largest scale [DYN06]). The challenge is to extend them and to combine seamlessly 2D simulation at global scale, macrophysics at mesoscale (air masses), 3D simulation at cloud scale, procedural details at small scales, etc.

Our preliminary results include a scalable shape model for vector-based data on terrains (using procedural techniques to amplify the terrain shape and appearance [BN08b]), a scalable illumination model of the atmosphere, from all viewpoints from ground to space [BN08a], a scalable shape, illumination and animation model of ocean waves in deep ocean [HNC02, BNH10], a scalable animation model of rivers [YNBH09], and some steps towards scalable shape and illumination models for trees and forests [MNP01, DN04, GMN05]. We integrated some of these results [BN08a, BN08b, YNBH09, BNH10] in a prototype Virtual Earth browser called Proland [Pro09] (see Fig. 1). We sold a license of this software to a planetarium company and we use it in an industrial project for flight simulations.

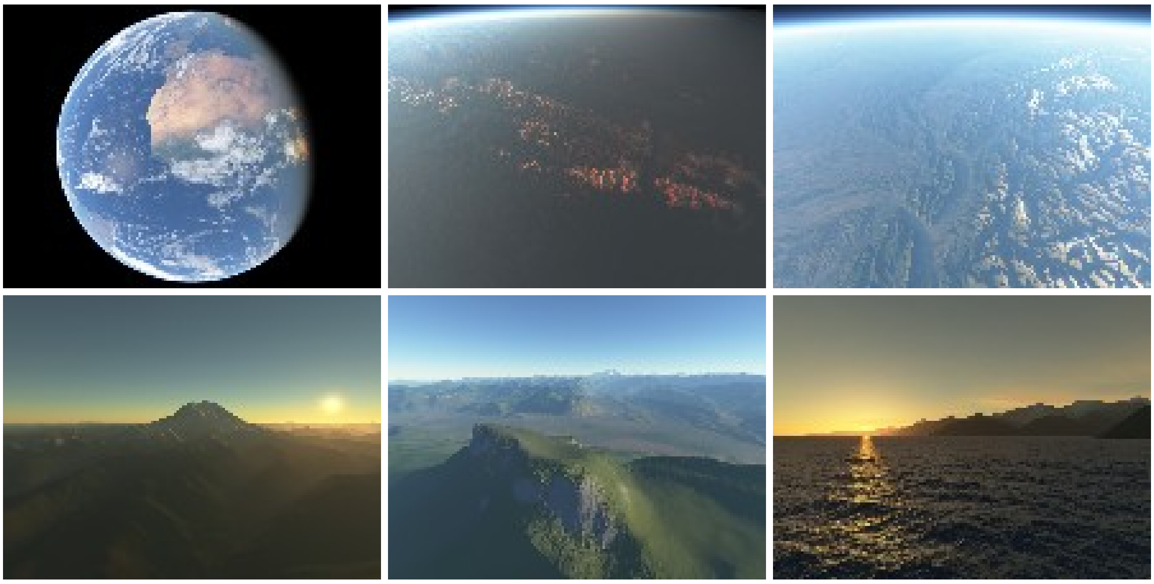

Figure 1: screenshots from our virtual Earth prototype, called Proland

2.2. Objectifs et caractère ambitieux/novateur du projet / Rationale highlighting the originality and novelty of the proposal

Our scientific goals are to find representations and algorithms to get real-time and realistic shape, illumination and motion models for natural objects (mainly vegetation, water and clouds) that can be displayed at all scales from centimeters to thousands of kilometers, without visible transitions between scales. We recall again that we do *not* target the best possible physical accuracy as in radiative transfer models or computational fluid dynamics methods (like for instance in remote sensing, climate modelization, meteorology, etc). Instead, we target physical plausibility, i.e., shape, illumination and motion models that look realistic and are efficient enough for real-time applications. This is a different but equally challenging problem.

Although natural elements are part of the research areas of the project's coordinator team (namely the EVASION team at the LJK laboratory), the scalability that we target here is not. Also the other members of the team are either not working on natural elements (e.g. scientific visualization), or on different elements than those targeted here (small animals, humans, etc). Hence our goals in this project are novel and original with respect to the research areas of the LJK laboratory.

These goals are also ambitious, as several scientific locks must be unlocked to reach them. Solving these hard problems, even for some specific cases only, would be important scientific breakthroughs:

- Scalable shape models are hard to design, especially when they must scale on several orders of magnitude. And providing seamless transitions between scales greatly complicates the problem. In fact these goals have been reached only in few cases, such as terrains [LH04], grass [BPB09] and deep ocean [BNH10] (and there are still some limitations, like visible transitions on silhouettes). Vegetation models are available for each scale separately (leaf, tree, forest, canopy), and some transitions have been investigated (for instance between the leaf and tree scales, and between the tree and forest scales), but providing seamless transitions between all scales remains an open problem.
- Scalable illumination models is an even harder problem. Indeed, averaging the shapes inside a pixel is much easier than averaging the illumination contribution of all these shapes, which can have different orientation, visibility (due to self occlusions), lighting (due to self shadowing and inter-reflections) and reflection properties. These problems have been tackled at the micro-scale, yielding BRDF models. The transitions between two scales have been studied in very few papers, generally using simplifying assumptions (for instance by ignoring self-occlusion and/or self shadowing and/or variable reflection properties). Some results are available for sand [TLQ+05] and ocean [BNH10], but scalable illumination models for terrains and vegetation remains an open problem. Scalable global illumination models for outdoor scenes is also an open problem ([BNL06] simulates inter-reflections between clouds and ground, but only for bounded flat terrains).
- Scalable motion models for fluids (water and clouds) is also a hard problem, especially when seamless transitions are needed across several orders of magnitude. Although multi-resolution techniques have been proposed for grid-based (Eulerian) and particle-based (Lagrangian) methods, providing real-time fluid motions on large domains remains difficult. Some methods have been proposed to amplify simulation results with noise-based motions [KTJG08, NSCL08, SB08] but they are not real-time. Procedural models and macroscopic physical models from hydraulics, wave theory, frontology, etc. could be used to simulate large scale motions at low cost, but the challenge is then to combine seamlessly 2D simulation at global scale, macrophysics at mesoscale (air masses), 3D simulation at cloud scale, procedural details at small scales, etc.

Our expected results are (see Section 3 for details):

- Scalable local illumination models for terrains and vegetation
- Scalable global illumination models for the clouds and terrain inter reflections
- Scalable motion models for ocean waves near shores, potentially including breaking waves
- Scalable motion models for clouds (at least at landscape and Earth scales)

We plan to integrate these results in our existing Virtual Earth prototype, called Proland [Pro09], in order to better spread these results and to increase the technical value of this demonstrator (a software license of the current code base has already been sold to a planetarium company).

3. Programme scientifique et technique, organisation du projet / Scientific and technical programme, project management

3.1. Programme scientifique et structuration du projet / Scientific programme, specific aims of the proposal

The goal of the project is the real-time unconstrained exploration of a realistic animated Earth from close to far viewpoints. We will only address natural elements: terrain and vegetation, ocean and rivers, clouds and atmosphere. Our preliminary results integrated in the Proland platform [Pro09] already provides some elements. Some of them are mature (scalable terrain, atmosphere rendering, static Earth-scale cloud layer rendering, deep ocean rendering and animation), while others are still very basic (animated rivers, forest models). We will not contribute to the above mature results. Instead, we plan to contribute on four tasks, presented below. We plan to integrate the new results in our prototype Proland, which will also be used as a testbed for new ideas.

Task 1: **scalable shape and illumination for terrains, forests and deep ocean**. The goal is provide continuous transitions from geometry to normal distributions (NDF) and then to reflectance models (BRDF), in the spirit of what we did in the "simple" case of deep ocean [BNH10]. This case was "simple" because we had an explicit spectral representation of the homogeneous surface, from which we could compute statistical properties for a BRDF. Still, we want to improve the transition quality and the surface spectrum sampling. We then want to apply the same ideas to vegetation, again using statistical properties (here of the distribution of leaves in trees). The problem here is harder because the apparent topology changes with scale (from disconnected leaves to a continuous 2D canopy surface; this requires different representations for the different scales). Finally we want to apply the same ideas to get a scalable terrain illumination model (we already have a scalable shape model). The main problem here is to average self shadows inside a pixel. It is similar to the anti-aliasing of shadowmap-based shadows (averaging the shadow map instead of the shadow test results gives wrong results, since the shadow test is not linear), except that we use horizon maps instead of shadow maps. Note that these "geometry to normals to BRDF transitions" are a very general and hard lock in Computer Graphics, especially when billions of details at the smallest scale project to the same pixel.

Task 2: **scalable global illumination between terrain, clouds and atmosphere**. The goal is to render in a realistic way the light interactions between the terrain, the clouds and the atmosphere. The Sun and sky light reflected by the terrain illuminates the clouds from below, which reflect this light back towards the ground, and so on. The light is also scattered by the atmosphere as it travels between the ground and the clouds. This gives visible effects like more luminous cloud bottoms above snow (due to its high albedo) and conversely, less luminous cloud bottoms above water (small albedo). Our preliminary work [BNL06] addressed this problem, but was limited to a flat cloud layer, a flat terrain, and a few dozens of kilometers. It was based on symmetry hypotheses that are no longer valid with an arbitrary terrain topography. Extending this work to the Earth scale while taking its topography into account is a challenging task.

Task 3: **scalable animated water surfaces**. The goal is to animate oceans and rivers in real-time and in a scalable way. Our preliminary work on scalable illumination and animation of the ocean [BNH10] is limited to deep ocean. Coasts and shores appear very unrealistic with this model. Indeed, wave properties change near shores due to depth variations, which cause the reflection and refraction of waves (and the breaking of waves at the shore). Some methods have been proposed to account for reflection and refraction of waves [GS97, GS00] but they rely on long precomputations using wave-tracing. They are not adapted to our needs (users can quickly move from one coast to another). We would also like to improve and extend our scalable real-time river animation model [YNBH09].

Task 4: **scalable cloud animation** (shape, motion, distribution and evolution). We will first investigate cloud animation at three independent scales: small scale (kilometers), medium scale (dozens of kilometers) and large scale (Earth). We will then work on seamless transitions between them. We do not target exact simulation, but only physical plausibility. At small scale we want to reproduce, in particular, the rising of thermals and cloud puffs, accounting for thermodynamics and phase changes ("small scale" means clouds seen from short distances, *not* a small domain: we want to animate clouds at this scale *anywhere on Earth*). At medium scale we want to reproduce the effects of topography and winds on the distribution and type of clouds. They are influenced by instability patterns (Benard cells, trails, etc). At large scale we want to reproduce the motion of air masses and fronts, resulting from large scale patterns like Rossby waves, Hadley cells, etc. The medium and large scales macrophysical models have never been studied in Computer Graphics, or even in physics (for a global animation model).

3.2. Coordination du projet / Project management

This project involves only two academic researchers of the same laboratory who work in neighbor teams, a PhD student and PostDoc / engineers hired on the project, plus another PhD student and master students (not financed by this project). The project management is therefore trivial. Moreover, tasks and sub-tasks are well separated and independent. A problem in one task will not affect the others. As soon as possible, our results will be integrated into the Proland [Pro09] platform (the first prototypes might be implemented outside this platform), which is managed by Eric Bruneton with the help of engineers (we plan to hire short term engineers on the projet to replace the current ones, whose contracts end in December 2010).

3.3. Description des travaux par tâche / Detailed description of the work organised by tasks

3.3.1 Tâche 1 / Task 1

Leader: Eric Bruneton
Participants: Eric Bruneton (18 h.m), PhD student (30 h.m), PostDoc / Engineer (5 h.m), master students

Goal: scalable shape and illumination for terrains, forests and deep ocean. The goal is to provide continuous transitions from geometry to normal distributions (NDF) and then to reflectance models (BRDF), in the spirit of what we did in the case of deep ocean [BNH10].

Subtask 1: scalable shape and illumination of ocean. Our preliminary work [BNH10] already provides good results, but we want to improve the quality of transitions between scales, as well as the sampling of the surface spectrum. Indeed, we currently build the ocean surface with a finite number of wave trains sampled from the continuous surface spectrum. This can give regular patterns due to wave interferences, which do not look very realistic.

Subtask 2: scalable illumination of vegetation. Several shape models are already available to display trees at various scales (from scales at which individual leaves are visible, to the canopy scale). We will therefore not contribute to this aspect. However, there is not currently a corresponding illumination model providing seamless transitions across all these scales. We plan to use the statistical properties of the distribution of leaves inside trees to get a BRDF model of the canopy at large distance. Some models have been proposed by physicists [SS93], but they need to be revisited for real-time rendering. A challenging problem is then the transition between large and small scales, where the number of leaves that project inside a pixel is too large to draw them one by one, but not large enough to use statistical properties. This general problem is complicated here by the fact that the apparent topology changes at the same time (from disconnected leaves to a continuous 2D canopy surface).

Subtask 3: scalable illumination of terrains. Unlike the above cases, here we can not rely on statistics to find a BRDF for the terrain seen at large distance (even at these distances, at most a few mountains can project inside a single pixel, not enough to derive statistics). In order to simplify the problem we suppose here that the terrain has a Lambertian BRDF at small scale, and that we can average separately inside a pixel the terrain reflectance and the terrain shadows (i.e., we assume that reflectance and shadows are uncorrelated, which is not always true: snow is more frequently found in shadowed regions). We also ignore inter-reflections and clouds (they are the subject of Task 2). The problem is then “reduced” to find a method to compute the fraction of a pixel that is covered by shadowed terrain parts (ignoring those masked by self-occlusions). We currently use horizon maps (more precisely ambient aperture maps [OS07]) to compute shadows at small scales. The problem is then how to efficiently average the result of several horizon map shadow tests (this is not the same as averaging the horizon map itself, due to non linearities). We plan to extend techniques that have been found to anti-alias shadowmap-based shadows (averaging a shadow map instead of the shadow test results also gives wrong results, since the shadow test is not linear).

3.3.2 Tâche 2 / Task 2

Leader: Eric Bruneton
Participants: Eric Bruneton (3 h.m), PhD student (6 h.m), PostDoc / Engineer (5 h.m), master students

Goal: scalable global illumination between terrain, clouds and atmosphere. The goal is to render in a realistic way the light interactions between the terrain, the clouds and the atmosphere.

Our preliminary work [BNL06] addressed this problem, but was limited to a flat cloud layer, a flat terrain, and a few dozens of kilometers. This removed terrain inter-reflections, and provided symmetries that we used to get an algorithm that did not need any precomputations (it was therefore compatible with cloud animation - task 4), but which was limited to a small domain. In order to extend this work to an arbitrary terrain topography and to the whole Earth, we think that some precomputations are necessary. We hope to be able to precompute potential terrain inter-reflections (the terrain is not animated), and radiance transfer functions between the terrain and a cloud layer decomposed on some functions basis (we do not want to enforce a specific cloud distribution, in order to allow animation). Also, due to the low altitude of clouds, light interactions between the ground and the clouds remain quite localized, which should allow us to perform tile-based computations during the creation of new terrain and cloud tiles, if necessary, as we did for terrains [BN08b].

3.3.3 Tâche 3 / Task 3

Leader: Eric Bruneton
Participants: Eric Bruneton (9 h.m), Fabrice Neyret (3 h.m), PhD student (6 h.m), PostDoc / Engineer (5 h.m), master students

Goal: scalable animated water surfaces. The goal is to animate oceans in real-time and in a scalable way.

Subtask 1: refraction of waves near shores. Coasts and shores appear very unrealistic with our preliminary work on scalable illumination and animation of the ocean [BNH10], which was limited to deep ocean. Indeed, wave properties change near shores due to depth variations, which cause the reflection and refraction of waves. Some methods have been proposed to account for reflection and refraction of waves [GS97, GS00] but they rely on long precomputations using wave-tracing for a given wind and a given current direction. They are not adapted to our needs (users can quickly move from one coast to another). Instead, we plan to use procedural methods using only local computations (for instance by interpolating potential functions as we did in [YNBH09]) to reproduce the phenomenological effects of wave refraction, such as the progressive alignment of wave fronts with the shore line.

Subtask 2: breaking waves. Another missing element in our work [BNH10] for a realistic model at shores is the breaking of waves, due the increase of wave heights, itself due to a decrease of the water depth. Several models have been proposed to render and animate breaking waves, mainly by trying to reproduce the phenomenological aspects of breaking waves (instead of trying to let them emerge from a low level and costly numerical fluid simulation – see for instance [FR86, GS97]). However these models are not adapted to a context where users can quickly move from one shore to another anywhere one Earth. For instance, [MMS04] relies on artist controls to select wave breaking profiles, which is clearly not possible in a large scale context. [TMF+07] relies on a shallow water fluid simulation on a grid, from which wave front lines are extracted to generate breaking waves where necessary. The use of a simulation grid is not adapted to a large domain. We also want to reproduce the phenomenological aspects instead of hoping them to emerge from a numerical fluid simulation, but we want to find plausible models that can be quickly evaluated for any shore anywhere on Earth.

Subtask 3: animated rivers. We want to extend and improve our scalable river animation model [YNBH09].

3.3.4 Tâche 4 / Task 4

Leader: Fabrice Neyret
Participants: Fabrice Neyret (15 h.m), Eric Bruneton (6 h.m), PhD student (30 h.m), PostDoc / engineer (3 h.m)

Goal: scalable cloud animation (shape, motion, distribution and evolution). We want to investigate cloud animation at three independent scales: small scale (kilometers), medium scale (dozens of kilometers) and large scale (Earth). Here the scale is related to the viewing distance, not to the size of the animated domain: we want to animate clouds at each scale on the whole Earth. We recall that we do not target exact simulation, but only physical plausibility.

Subtask 1: small scale cloud animation. Here we only address convective clouds such as cumulus, i.e., those with a well defined and fast evolving shape. A single thermal (or plume, or “puff”) rising in a static environment is easy to animate with particle dynamics and thermodynamics laws. However thermals influence each other since they modify their environment, and small thermals can rise in the environment of a big thermal. This suggests a hierarchical model mixing particles and grids, the grids being used to store the environmental conditions and to speed-up the computations of the interactions between neighbor particles. This is a difficult task in the general case, but since we only target plausibility, we can make simplifying assumptions, like only one-way coupling between scales.

In order to animate these thermals anywhere on Earth for ground level views, we plan to use an adaptive representation, to animate only the air parcels that are in the view frustum, at a resolution adapted to their distance to the viewer (in the spirit of [BN08b, YNBH09]). Variations will be provided by varying terrain properties (humidity, temperature, etc), yielding varying thermal emission rates. The main problem is to get a plausible state for new parcels, when the view changes. We plan to perform a few simulation steps *backward* to recover this state. This is a totally new approach, not expected to be easy. We hope that allowing plausible but not fully accurate animations will simplify the problem (for instance we won't require a cloud to show up at the same place when going back to a previous location).

Subtask 2: medium scale macrophysical models. Here we want to reproduce the phenomenological effects of the topography and of the global wind on the motion and distribution of clouds (like cloud channeling in valleys, wind triggered thermals, orographic clouds, hexagonal cloud patterns resulting from Benard cells, cloud stripes resulting from Lee waves, etc). The conditions and characteristics of these patterns are well known, but hoping them to emerge from a numerical fluid simulation is not reasonable, and would not yield controllable results. Instead, we plan to use models inspired from texture synthesis to reproduce these patterns directly where and when they should appear. A reasonable idea is to rely on a 2D cellular automaton with relaxation. It is not expected to be easy since it is a totally new topic.

We also want here to use the phenomenological patterns at the next scale, which are air masses, separated horizontally by front lines. We plan to use as input a vector representation of the front lines (keyframed by an artist, coming from weather forecasts or from a larger scale simulation), and of the macroscopic parameters of air masses and fronts (temperature, humidity, vertical slope, etc), in order to control the previous phenomenological cloud patterns (the formation of clouds at fronts is well understood in meteorology [Vas02, Meteo]).

Subtask 3: large scale models. Here we want to reproduce the motion and evolution of high and low pressure areas, which give the air masses and fronts. The atmospheric physics literature [Vas02, Hol04] describes the characteristics and evolution of winds and fronts across them, and also explains the phenomenological patterns at this scale, like Hadley cells (which involve vertical motions), Rossby waves, etc. It is quite easy to simulate highs and lows with a 2D ½ fluid simulation [DKY+00] taking into account Coriolis forces, where users can specify directly the location and properties of highs and lows, and then let the system simulate their evolution. But it is not so easy to get the phenomenological patterns as emerging properties of this simulation. We might therefore need to find models allowing users to enforce some patterns (which would be a plus anyway for applications with scenarios – games, special effects, flight simulation training).

The risks and difficulties in Task 4 are linked to set of new representations and algorithms to be invented, from the few existing tools in Computer Graphics, or directly from the physics literature. Another problem is to tune the algorithms so that they produce plausible visual phenomena. Our strategy to reproduce the phenomenological patterns directly at each scale, instead of hoping them to emerge from low level fluid simulation is a big advantage here. Note that the models envisioned in this task are independent. Each can be seen as a “fluid amplification” bringing details to the larger model. So, risks are local: beside simulating automatically the whole Earth from ground to satellite scale, the techniques we propose can also be used to visually enrich weather forecast simulations, or to offer high scale handles to control the look of detailed simulation, as expect by scenarists (for games, special effects, flight simulations training).

4. Annexes

4.1. Références bibliographiques / References

4.1.1 Our related publications

[BN08a] Eric Bruneton and Fabrice Neyret. Precomputed atmospheric scattering. Comput. Graph. Forum, 27(4):1079–1086, June 2008. Special Issue: Proceedings of the 19th Eurographics Symposium on Rendering 2008.

[BN08b] Eric Bruneton and Fabrice Neyret. Real-time rendering and editing of vector-based terrains. Comput. Graph. Forum, 27(2):311–320, April 2008. Special Issue: Eurographics ’08.

[BNH10] Eric Bruneton, Fabrice Neyret, and Nicolas Holzschuch. Real-time realistic ocean lighting using seamless transitions from geometry to brdf. Comput. Graph. Forum, 29(2), May 2010. to appear.

[BNL06] Antoine Bouthors, Fabrice Neyret, and Sylvain Lefebvre. Real-time realistic illumination and shading of stratiform clouds. In Eurographics Workshop on Natural Phenomena, sep 2006.

[BNM+08] Antoine Bouthors, Fabrice Neyret, Nelson Max, Eric Bruneton, and Cyril Crassin. Interactive multiple anisotropic scattering in clouds. In ACM Symposium on Interactive 3D Graphics and Games, I3D, February, 2008, Redwood City, CA, Etats-Unis, 2008.

[DN04] Philippe Decaudin and Fabrice Neyret. Rendering forest scenes in real-time. In Alexander Keller and Henrik Wann Jensen, editors, 15th Eurographics Workshop on Rendering, Rendering Techniques'04, June, 2004, pages 93–102, Norköping, Suède, 2004. Eurographics Association.

[GMN05] Guillaume Gilet, Alexandre Meyer, and Fabrice Neyret. Point-based rendering of trees. In P. Poulin E. Galin, editor, Eurographics Workshop on Natural Phenomena, 2005.

[GPR+03] Sylvain Guerraz, Frank Perbet, David Raulo, Francois Faure, and Marie-Paule Cani. A procedural approach to animate interactive natural sceneries. In 16th International Conference on Computer Animation and Social Agents, CASA 2003, May, 2003, pages 73–78, Rutgers University, New Brunswick, NJ, Etats-Unis, May 2003. IEEE.

[HNC02] Damien Hinsinger, Fabrice Neyret, and Marie-Paule Cani. Interactive animation of ocean waves. In ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA), july 2002.

[MNP01] Alexandre Meyer, Fabrice Neyret, and Pierre Poulin. Interactive rendering of trees with shading and shadows. In Proceedings of the 12th Eurographics Workshop on Rendering Techniques, pages 183–196, London, UK, 2001. Springer-Verlag.

[Ney97] Fabrice Neyret. Qualitative simulation of cloud formation and evolution. In D. Thalmann and M. Van de Panne, editors, 8th Eurographics Workshop on Computer Animation and Simulation (EGCAS'97), pages 113–124, New York City, NY, September 1997. Eurographics, Springer Wein.

[Ney98] Fabrice Neyret. Modeling animating and rendering complex scenes using volumetric textures. IEEE Transactions on Visualization and Computer Graphics, 4(1):55–70, January–March 1998. ISSN 10772626.

[Ney03] Fabrice Neyret. Advected textures. In ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA), july 2003.

[PN01] 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.

[Pro09] Eric Bruneton, Antoine Begault, Proland
http://www.projet-plume.org/relier/proland
http://www-evasion.imag.fr/Membres/Eric.Bruneton

[YNBH09] Qizhi Yu, Fabrice Neyret, Eric Bruneton, and Nicolas Holzschuch. Scalable real-time animation of rivers. Comput. Graph. Forum, 28(2):239–248, April 2009.

4.1.2 Other references

[AH05] Arul Asirvatham and Hugues Hoppe. GPU Gems 2, chapter Terrain rendering using GPU-based geometry clipmaps. Addison Wesley, 2005.

[AM09] Yacine Amara and Xavier Marsault. A GPU tile-load-map architecture for terrain rendering: theory and applications. The Visual Computer, pages 805-824, 2009.

[AN05] Alexis Angelidis and Fabrice Neyret. Simulation of smoke based on vortex filament primitives. In ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA), 2005.

[APKG07] Bart Adams, Mark Pauly, Richard Keiser, and Leonidas J. Guibas. Adaptively sampled particle fluids. ACM Transactions on Graphics, 26(3), 2007.

[AS00] Michael Ashikhmin and Peter Shirley. An anisotropic Phong BRDF model. Journal of Graphics Tools, 5:25–32, 2000.

[BBP08] Kevin Boulanger, Kadi Bouatouch, and Sumanta N. Pattanaik. Rendering trees with indirect lighting in real time. Comput. Graph. Forum (proceedings of 19th Eurographics Symposium on Rendering 2008), 27(4):1189–1198, 2008.

[BCF+05] S. Behrendt, C. Colditz, O. Franzke, J. Kopf, and O. Deussen. Realistic real-time rendering of landscapes using billboard clouds. volume 24, pages 507–516, September 2005.

[BHN07] Robert Bridson, Jim Houriham, and Marcus Nordenstam. Curlnoise for procedural fluid flow. ACM Transactions on Graphics, 26(3), 2007.

[BJ91] Michel Beigbeder and Ghassan Jahami. Managing levels of detail with textured polygons. In COMPUGRAPHICS '91, volume I, pages 479–489, 1991.

[Bli78] James F. Blinn. Simulation of wrinkled surfaces. In Computer Graphics (SIGGRAPH '78 Proceedings), volume 12(3), pages 286–292, August 1978.

[Bli82] J. F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In Computer Graphics (SIGGRAPH '82 Proceedings), volume 16(3), pages 21–29, July 1982.

[BM93] Barry G. Becker and Nelson L. Max. Smooth transitions between bump rendering algorithms. SIGGRAPH 93, 27:183–190, 1993.

[BPB09] Kevin Boulanger, Sumanta N. Pattanaik, and Kadi Bouatouch. Rendering grass in real time with dynamic lighting. IEEE Computer Graphics and Applications, 29:32–41, 2009.

[CCDH05] Carsten Colditz, Liviu Coconu, Oliver Deussen, and Christian Hege. Realtime rendering of complex photorealistic landscapes using hybrid level-of-detail approaches. In Trends in Real-Time Landscape Visualization and Participation, pages 97–106, 2005.

[Che04] Stephen Chenney. Flow tiles. In Symposium on Computer Animation, pages 233–242, 2004.

[CT81] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. SIGGRAPH 81, 15(3):307–316, 1981.

[DCSD02] Oliver Deussen, Carsten Colditz, Marc Stamminger, and George Drettakis. Interactive visualization of complex plant ecosystems. In VIS '02: Proceedings of the conference on Visualization '02, pages 219–226, Washington, DC, USA, 2002. IEEE Computer Society.

[DHL+98] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomir Mech, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 275–286, New York, NY, USA, 1998. ACM.

[DKY+00] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A simple, efficient method for realistic animation of clouds. In SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pages 19–28, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[DWS+97] Mark A. Duchaineau, Murray Wolinsky, David E. Sigeti, Mark C. Miller, Charles Aldrich, and Mark B. Mineev-Weinstein. ROAMing terrain: real-time optimally adapting meshes. In IEEE Visualization, pages 81–88, 1997.

[DYN06] Y. Dobashi, T. Yamamoto, and T. Nishita. A controllable method for animation of earth-scale clouds. Proceedings of CASA 2006, pages 43–52, 2006.

[EMP+94] David Ebert, Kent Musgrave, Darwyn Peachey, Ken Perlin, and Worley. Texturing and Modeling: A Procedural Approach. Academic Press, October 1994. ISBN 0-12-228760-6.

[FFC82] A. Fournier, D. Fussell, and L. Carpenter. Computer rendering of stochastic models. In Comm. ACM, volume 25(6), pages 371 – 384, June 1982.

[FMU05] A Fuhrmann, S Mantler, and E Umlauf. Extreme model simplification for forest rendering. in Proceedings of EGWNP - Eurographics Workshop on Natural Phenomena, 2005.

[Fou92] Alain Fournier. Normal distribution functions and multiple surfaces. In Graphics Interface '92 Workshop on Local Illumination, pages 45–52, May 1992.

[FR86] Alain Fournier and William T. Reeves. A simple model of ocean waves. In David C. Evans and Russell J. Athay, editors, Computer Graphics (SIGGRAPH '86 Proceedings), volume 20, pages 75–84, August 1986.

[GMC+06] Enrico Gobbetti, Fabio Marton, Paolo Cignoni, Marco Di Benedetto, and Fabio Ganovelli. C-BDAM - compressed batched dynamic adaptive meshes for terrain rendering. Computer Graphics Forum, 25(3), Sep 2006. Eurographics 2006 conference proceedings.

[GS97] Jean-Christophe Gonzato and Bertrand Le Saëc. A phenomenological model of coastal scenes based on physical considerations. In Computer Animation and Simulation '97, Eurographics, pages 137–148, 1997.

[GS00] Jean-Christophe Gonzato and Bertrand Le Saëc. On modelling and rendering ocean scenes. The Journal of Visualization and Computer Animation, 11(1):27–37, 2000.

[HL01] Mark J. Harris and Anselmo Lastra. Real-time cloud rendering. Computer Graphics Forum, 20(3):76–84, 2001.

[Hol04] James R. Holton. An Introduction to Dynamic Meteorology. Fourth Edition. Elsevier Academic Press.

[HPAD06] Kyle Hegeman, Simon Premoze, Michael Ashikhmin, and George Drettakis. Approximate ambient occlusion for trees. In I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games, pages 87–92, New York, NY, USA, 2006. ACM.

[HSRG07] Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. Frequency domain normal map filtering. SIGGRAPH 07, 26(3):28, 2007.

[HTSG91] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. SIGGRAPH 91, 25(4):175–186, 1991.

[Kaj85] James T. Kajiya. Anisotropic reflection models. SIGGRAPH 85, 19(3):15–21, 1985.

[KM90] Michael Kass and Gavin Miller. Rapid, stable fluid dynamics for computer graphics. In SIGGRAPH ’90, pages 49–57, 1990.

[KTJG08] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. In SIGGRAPH ’08: ACM SIGGRAPH 2008 papers, pages 1–6, New York, NY, USA, 2008. ACM.

[KVHS00] Jan Kautz, Pere-Pau Vazquez, Wolfgang Heidrich, and Hans-Peter Seidel. Unified approach to prefiltered environment maps. In Rendering Techniques, pages 185–196, 2000.

[LD06] Ares Lagae and Philip Dutré. A comparison of methods for generating Poisson disk distributions. Report CW 459, Department of Computer Science, K.U.Leuven, Leuven, Belgium, August 2006.

[LGF04] Frank Losasso, Frédéric Gibou, and Ron Fedkiw. Simulating water and smoke with an octree data structure. ACM Transactions on Graphics, 23(3):457–462, 2004.

[LH04] Frank Losasso and Hugues Hoppe. Geometry clipmaps: terrain rendering using nested regular grids. In ACM SIGGRAPH, pages 769–776, 2004.

[LP02] Brendan Lane and Przemyslaw Prusinkiewicz. Generating spatial distributions for multilevel models of plant communities. In Proceedings of Graphics Interface, pages 69–80, 2002.

[Meteo] Manual of Synoptic Scale Meteorology. http://www.zamg.ac.at/docu/Manual/SatManu/main.htm

[MMS04] Viorel Mihalef, Dimitris Metaxas, and Mark Sussman. Animation and control of breaking waves. In SCA ’04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 315–324, 2004.

[MP96] Radomir Mech and Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. In SIGGRAPH ’96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 397–410, New York, NY, USA, 1996. ACM.

[NDN96] Tomoyuki Nishita, Yoshinori Dobashi, and Eihachiro Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. In SIGGRAPH ’96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 379–386, New York, NY, USA, 1996. ACM.

[NSCL08] Rahul Narain, Jason Sewall, Mark Carlson, and Ming C. Lin. Fast animation of turbulence using energy transport and procedural synthesis. ACM Trans. Graph., 27(5):1–8, 2008.

[ON94] Michael Oren and Shree K. Nayar. Generalization of lambert’s reflectance model. In Andrew Glassner, editor, Proceedings of SIGGRAPH ’94 (Orlando, Florida, July 24–29, 1994), Computer Graphics Proceedings, Annual Conference Series, pages 239–246. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[OS07] Christopher Oat and Pedro V. Sander. Ambient aperture lighting. In I3D ’07: Proceedings of the 2007 symposium on Interactive 3D graphics and games, pages 61–64, New York, NY, USA, 2007. ACM.

[Paj98] Renato B. Pajarola. Large scale terrain visualization using the restricted quadtree triangulation. In David Ebert, Hans Hagen, and Holly Rushmeier, editors, IEEE Visualization ’98, pages 19–26, 1998.

[Per85] Ken Perlin. An image synthesizer. In B. A. Barsky, editor, Computer Graphics (SIGGRAPH ’85 Proceedings), volume 19(3), pages 287–296, July 1985.

[PH93] Przemyslaw Prusinkiewicz and Mark Hammel. A fractal model of mountains with rivers. In Graphics Interface, pages 174–180. Canadian Information Processing Society, Canadian Human-Computer Communications Society, May 1993. ISBN.

[PT05] Mayur Patel and Noah Taylor. Simple divergence-free fields for artistic simulation. journal of graphics tools, 10(4):49–60, 2005.

[PTB+03] Simon Premoze, Tolga Tasdizen, James Bigler, Aaron Lefohn, and Ross T. Whitaker. Particle-based simulation of fluids. Computer Graphics Forum, 22(3):401–410, 2003.

[RB85] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In B. A. Barsky, editor, Computer Graphics (SIGGRAPH ’85 Proceedings), volume 19(3), pages 313–322, July 1985.

[Ree83] W. T. Reeves. Particle systems – a technique for modeling a class of fuzzy objects. ACM Trans. Graphics, 2:91–108, April 1983.

[REK+04] Kirk Riley, David S. Ebert, Martin Kraus, Jerry Tessendorf, and Charles D. Hansen. Efficient rendering of atmospheric phenomena. In Rendering Techniques, pages 374–386, 2004.

[RRC+03] Remolar Chover Ribelles, I. Remolar, M. Chover, J. Ribelles, O. Belmonte, and Dep Lenguajes Y Sistemas. View-dependent multiresolution model for foliage. In Journal of WSCG03, page 2003, 2003.

[SB08] H. Schechter and R. Bridson. Evolving sub-grid turbulence for smoke animation. In SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 1–7, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[Sch94] Christophe Schlick. A survey of shading and reflectance models. Computer Graphics Forum, 13(2):121–131, June 1994.

[SF92] Mikio Shinya and Alain Fournier. Stochastic motion - motion under the influence of wind. In A.C. Kilgour and L. Kjelldahl, editors, Computer Graphics Forum (Eurographics '92), volume 11(3), pages 119–128, September 1992.

[SF93] Jos Stam and Eugene Fiume. Turbulent wind fields for gaseous phenomena. In SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 369–376, New York, NY, USA, 1993. ACM.

[SKP05] Musawir A. Shah, Jaakko Kontinnen, and Sumanta Pattanaik. Realtime rendering of realistic-looking grass. In GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, pages 77–82, New York, NY, USA, 2005. ACM.

[SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pages 527–536, New York, NY, USA, 2002. ACM.

[SS93] Liang Shunlin and A. H. Strahler. An analytic BRDF model of canopy radiative transfer and its inversion. In IEEE transactions on geoscience and remote sensing. vol. 31, no 5, pp. 1081-1092, 1993.

[Sta97] Jos Stam. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. Computer Graphics Forum, 16(3):159–164, August 1997. Proceedings of Eurographics '97.

[Sta99] Jos Stam. Stable fluids. In SIGGRAPH '99, pages 121–128, 1999.

[Sta01] Jos Stam. A simple fluid solver based on the fft. J. Graphics Tools, 6(2):43–52, 2001.

[Sta03] Jos Stam. Real-time fluid dynamics for games, 2003.

[TLQ+05] Ping Tan, Stephen Lin, Long Quan, Baining Guo, and Heung-Yeung Shum. Multiresolution reflectance filtering. In Rendering Techniques, pages 111–116, 2005.

[TLQ+08] Ping Tan, Stephen Lin, Long Quan, Baining Guo, and Harry Shum. Filtering and rendering of resolution-dependent reflectance models. IEEE Transactions on Visualization and Computer Graphics, 14(2):412–425, 2008.

[TMF+07] Nils Thürey, Matthias Müller-Fischer, Simon Schirm, and Markus Gross. Real-time Breaking Waves for Shallow Water Simulations. Proceedings of the Pacific Conference on Computer Graphics and Applications 2007, page 8, October 2007.

[Tur92] Greg Turk. Re-tiling polygonal surfaces. In Edwin E. Catmull, editor, Computer Graphics (SIGGRAPH '92 Proceedings), volume 26, pages 55–64, July 1992.

[Vas02] Tim Vasquez. Weather Forecasting Handbook. Fifth Edition. ISBN 0-9706840-2-9.

[Wan03] Niniane Wang. Realistic and fast cloud rendering in computer games. In SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications, pages 1–1, New York, NY, USA, 2003. ACM.

[War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. SIGGRAPH 92, 26(2):265–272, 1992.

[WH91] Jakub Wejchert and David Haumann. Animation aerodynamics. In Thomas W. Sederberg, editor, Computer Graphics (SIGGRAPH '91 Proceedings), volume 25, pages 19–22, July 1991.

[WP95] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. In SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 119–128, New York, NY, USA, 1995. ACM.