REAL-TIME REALISTIC ILLUMINATION AND RENDERING OF CUMULUS CLOUDS

Sassi Abdessamed, Djedi Noureddine and Sassi Amina

Department of Computer Science,
Mohamed Khider University, 07000 Biskra, Algeria

ABSTRACT

Realistic simulation of natural phenomena such as clouds is one of the most challenging problems facing computer graphics. The complexity of cloud formation, dynamics and light interaction makes real time cloud rendering a difficult task. In addition, traditional computer graphics methods involve high amounts of memory and computing resources, which currently limits their realism and speed.

We propose an efficient and computationally inexpensive phenomenological approach for modelling and rendering cumulus clouds, by drawing on several approaches that we combine and extend.

This paper focuses on the modelling of the cloud's shape, rendering and sky model but is does not deal with the animation of the cloud.

KEYWORDS

Image Synthesis, Natural Phenomena, Clouds, Rendering & Procedural Modelling.

1. INTRODUCTION

Natural scenes are very large and have a very high visual complexity, including Clouds, which make their synthesis a challenge for those who are harnessed, and even worse in a real-time case. Clouds are an important elements for creating realistic images of natural scenes, outdoor scenes, flight simulators, space flight simulators, visualization of the weather information, creation of realistic clouds from satellite images, simulation of surveys of the earth, earth viewed from outer space, film, art and so on. Clouds remain one of the most significant challenges in the area of modelling natural phenomena for computer graphics. Synthesizing images of realistic clouds is a difficult task, whether one looks at their shape, their animation or their rendering. A wide variety of clouds exists, from haze to stratus, requiring different approaches. In this paper, we only deal with cumulus clouds, which are well-formed and much contrasted cloud. A phenomenological approach is used to avoid a complex simulation, by accounting directly for the useful appearance features. This paper is structured as follows: Section 2 reviews previous work on cloud modelling and rendering. Section 3 describes the important visual characteristics of cumulus clouds, in order to establish what really needs to be reproduced. Section 4 proposes our approach to simulate a cumulus cloud (modelling and shading). Section 5 presents our preliminary results, and discusses future works.

DOI : 10.5121/ijma.2013.5503

2. PREVIOUS WORKS

Many aspects related to the clouds are addressed, from the shape and dynamics (motion/movement) to global illumination, sky color and anisotropic scattering. Many kinds of clouds have been studied, and a lot of work has been published to date. This section is divided into two parts, Modelling the shape and rendering. However, some works focused on both aspects at a time.

2.1. Modelling the shape

Many methods for modelling the shape and motion of clouds have been proposed in computer graphics (we will not survey all of the published work).

Harris [9] has simulated in a relatively precise way the evolution of the clouds, like Stam [35], but with a spatial resolution very frustrates compared to the capacity of discrimination of the human eye. Dobashi [8] proposed an approach of numerical simulation of clouds for images synthesis applications using cellular automats in order to imitate the way in which the clouds grow and move. Miyazaki [36] extended this to use a Coupled Map Lattice model, a technique similar to the cellular automats but allowing to simulate the physics of the fluids, however always with a relatively coarse scale.

Other works based on a phenomenological approach, to obtain a plausible form with a lower cost. These approaches use sometimes elements of simulation in the form of system of particles [37]. While others prefer a purely phenomenological approach : Gardner [17], [18] represented the clouds like a cluster of ellipsoids. Voss [4] created realistic clouds with fractals. Perlin. [34] based on a fractal noise. Gardner [18] has produced realistic images of clouds by using Fourier synthesis. However, this does not create a true three-dimensional geometric model. Ebert [38] have used solid texture to generate clouds in three-dimensional spaces. Ebert [11] has also developed a method combining metaballs and a noise function to model clouds, he has used a functions of form like the implicit surfaces. Nishita [7] created clouds by generating metaballs using the idea of fractals, he has also combined arranged particles in hierarchical way with implicit surfaces. Kajiya [3] used a simple method based on Partial Differential Equations (PDE) to generate cloud data sets for their ray-tracing algorithm. Stam [39] has developed a simple method for modeling clouds. In their method, a user specifies density values at several points in three dimensional space. Then the density distribution of the clouds is obtained by interpolating the specified density values. Bouthors [27] has presented a method of creating the shape of the cloud from a hierarchy of particles placed on implicit surfaces.

Other methods recreating a cloud shapes from real 2D data (photographs type) and satellite images [40], [41], [42].

2.2. Rendering

Blinn [1] and Max [2] have proposed early models to represent the light reflection and diffusion through clouds. Kajiya [3] developed the works of Blinn and proposed a more general model of multiple diffusion and a method of ray tracing for volumetric rendering objects, he also presented an interesting physical model to simulate the formation of clouds. Voss [4] has used a similar method to produce clouds fractals in their scenes. Kaneda [5] also has simulated the anisotropic scattering in clouds and fog, including a case of multiple scattering. Max [6] extended the rendering to take into account the anisotropic scattering of light inside the cloud, and getting an illuminated corona when the light comes from the back. This method was later reused, using heuristics based on the principle of forward scattering typical of clouds Nishita [7]. In addition to the simulation of the multiple anisotropic diffusion by the particles of the cloud, it takes into account the luminous contribution of the whole of the sky, light reflected by the ground and also to the light scattering by the particles of the atmosphere before and after the crossing of the cloud. Then optimized by using the possibilities provided by the graphics hardware [8], [9].

Generally, a lot of work has been done on volumetric radiosity, starting with the zonal method by Rushmeier and Torrance [10]. These techniques and their variants have sometimes been used to render clouds, such as for the images generated by Ebert [11]. Stam [12], [13] has done much work on many aspect of fluids, in particular smoke and clouds, from representation to rendering and animation, including a rendering dedicated to stochastic densities [14]. In particular, he introduces an efficient method to compute light diffusion inside clouds [13].

Another aspect is the use of textures, fractals and shaders to represent the details. Musgrave [15] has extensively used them for cloudy skies.

Voss and Gardner [16], [18] have proposed empirical models based on textures fractals for the visualization of clouds. Gardner [17], [18] has proposed a very nice yet simple model of cumulus shape, based on ellipsoids whose transparency is textured and amplified near the silhouette. These models have also been used in combination with physical ones to enrich them with details.

The recent graphic material makes it possible to define pixels shaders, allowing to adapt their approaches in real time. The first attempt was made by Elinas [19].

Neyret [20] has effected a preliminary study to improve the rendering quality by taking into account the phenomenological observations, but keeping a rendering by ray-tracing. Since the cumulus are dense and often have a clear interface, they were also represented by volumes limited by surfaces, such as whole of ellipsoids [18], [19] or grids [21]. Premoze [22] introduced the idea of Most Probable Paths (MPP) in participating media. In addition, Premoze [23] speed up this technique and account for the spatial spreading of light around this mean path through an analytical formulation. Hegeman [24] has been brought this approach to real-time using graphics hardware for volume slicing in a manner similar to Harris [25], Riley [26]. Bouthors [27] has extended the study of Neyret [20], he described a phenomenological approach for realize a cumulus clouds model from observations of real scenes. Bouthors [28] focused on the rendering of stratiform clouds, it allows a convincing rendering of clouds (and very good performance). He simulated the light transport inside a gird, but do not consider any enrichment on the silhouettes, which appear polygonal and opaque. In this cloud the light is anisotropically scattered (Mie scattering) several times before leaving.

Bouthors [29] also proposed a new approach to calculating luminous transport, faster, which does not require slicing or browsing the volume (they treated the luminous paths of all orders). This is a rendering method for any type of cloud. It provides a photorealistic rendering, unfortunately it is very computationally expensive and is not suitable for a scene with a wider scale than a single cloud. Belcour [30] has extended the clouds model of Bouthors [28] (a local scale, the Earth can be approximated by a plane). The aim of the work of Belcour [30] is to make a real-time rendering of clouds on the scale of planet, for this they modified the model to pass from a Ground plane to a spherical Ground.

Elek [31] has presented an interactive algorithm for simulation of light transport in clouds. Exploiting the high temporal coherence of the typical illumination and morphology of clouds that build on volumetric photon mapping, which he modified to allow for interactive rendering speeds, instead of building a fresh irregular photon map for every scene state change we accumulate photon contributions in a regular grid structure.

3. CHARACTERISTICS OF CUMULUS CLOUDS

A common characteristic for all kinds of clouds is that the albedo is very close to 1 (there is almost no absorption), and that the droplets induce a high forward scattering.
The cumulus clouds are particularly dense and thick; this implies other unique and essential characteristics: the scattering at large scale is roughly isotropic, except in the thin external layer that is visible on the silhouette.

A consequence is that the cloud surface roughly re-emits the same amount of light that it has received in the neighborhood of each point, isotropically. We can therefore apply, in a first approximation the Lambert reflectance model.

This leads to another consequence: the radiosity plays an important role in the aspect of a cumulus. Indeed, the light reemitted by a surface element has almost the same energy than the received one (due to the important albedo). If this re-emitted light is received by another surface element, it significally increases the amount of energy that this one receives and re-emits. In this manner, les inter-reflections are almost not absorbed, and concavities on the cloud surface (e.g.between bubbles) behave like light traps, which can make some concavities be almost as bright as the sun. (see Fig. 1). This sometimes gives the impression that the cloud is illuminated from the inside.

This behavior can also arise, despite less intense, in areas that are not directly illuminated by the sun.

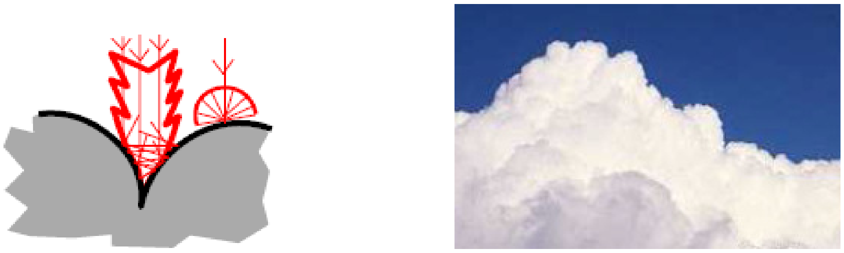

Figure 1. Left : amplified reflection in a concavity of the surface.
Right: Light amplification by concavities on real clouds.

Lastly, the cumulus corona has an aspect relatively different from that of the core of the clouds, as illustrated on Figure 2.

Indeed, being given the low density of the droplets at the surface of the cloud, the cumulated scattering remains narrow, forward and anisotropic (note that the diffuse clouds such as stratus, young cumulus or haze, have this property at any point).

This makes cumulus looking very contrasted: when illuminated from the front, the body is intensely white and the corona is almost transparent. When illuminated from the back, the core is extremely dark and homogeneous, while the corona is illuminated due to forward scattering (see Fig. 2).

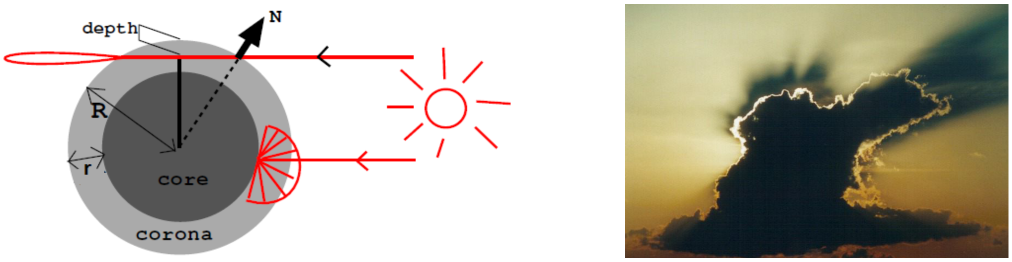

Figure 2. Left: result of cumulated scattering on the cloud core and on the corona.
Right: Enlighted corona of real clouds.

4. Our approach

Our idea is to represent the cloud as a hierarchy of agglutinated spheres or particles recursively defined on top of each other, some flattened at their base.

First, we need to make this hierarchy of particles, then we must find a model of illumination taking into account the characteristics of aspect described in Section 3 to reproduce the effects observed: generally reflective aspect, edges having a high forward scattering, furrows acting like a light traps , etc.

For this, we inspire from Bouthors's model [27], that we will extend by using the sky model makes by Nishita [32].

4.1. Modelling a Cumulus Clouds shape

If we observe a cumulus with more attention we can see that its global form is enough homogeneous and compact. This important characteristic allows to envisage the possibility of modelling a cloud at least on its shape.

Our idea is to represent the cloud as a set of hierarchical levels recursively defined on top of each other. Each level l defines by its own surface S_l and a finite number of particles. The particles of the level l is defined by their location on the surface S_{l-1} , their radius and their flattening (a randomly factor chosen between 0.1 and 1).

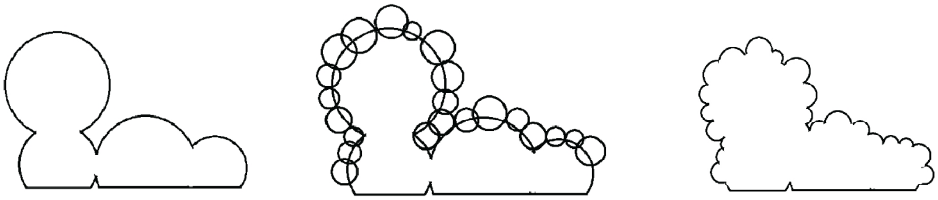

Figure 3. Left: Particles of level 0. Middle: Particles created on S_0. Right: surface S_1 of level 1

We take different radius for the particles of the same level, so that all the particles of a level do not have the same aspect and the layer look uneven.

Particles of level 0 (i.e., root) can be done in two cases:

Procedurally: in this case, we create a small number of particles randomly positioned on a horizontal plane or in a small subspace.

Controlled by the user: in the second case, an interface lets the user shape and place them by hand, allowing him to create realistic features.

When all of the particles of level 0 are created, we can generate a surface to place the particles of the following level, and so on.

The images presented in Figure 4 show some results of the modelling of our cloud, they show a clouds with a root level consisting of some particles and two supplementary levels generated on this one, resulting in a cumulus.

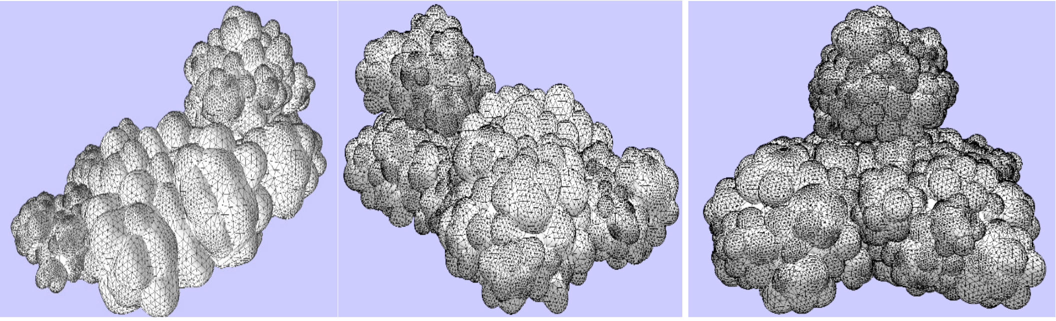

Figure 4. Two levels of hierarchy generated on the root level form a cloud.

This representation very figurative appears very simplistic. However, it gives a good approximation of the global geometry of a cloud. Moreover, it should be know that this geometry comes to be added a texture that will give to the cloud its cottony aspect, its behavior with the light (the transparency at the edges, light traps).

4.2. Rendering

We use the observations reported in the section 3 to create the rendering function inspired from that of Bouthors [27] that we extend by using the sky model makes by Nishita [32] to give a realistic and credible aspect to the cloud. Moreover, we implement the rendering function using the functionalities of recent graphics hardware (vertex and pixel shaders) to obtain a real-time rendering.

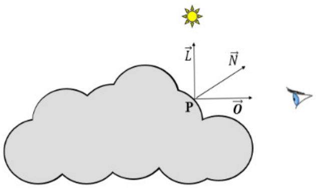

Figure 5. Elements used in the calculation of the illumination of a point.

$\vec{N}$: Vector perpendicular to the reflecting surface (the normal).
$\vec{L}$: Vector in the direction of the light source.
$\vec{O}$: Vector in the direction of the observer.

We also define the function saturate such as:

$$\text{saturate}(x) = \min(1, \max(0,x))$$

4.2.1. Basic Illumination

As indicated in the section 3, the particularities of cumulus encourage us in a first approximation to keep the **Lambert reflectance model** for the cloud outside the corona.
According to Lambert's law, the brightness of a surface at a point is proportional to the cosine of the angle between the normal to this surface $\vec{N}$ and the direction of the light $\vec{L}$.
Mathematically, the color I (P) at the point P is given by:

$$I\,(P) = k_d\, C_s \text{ saturate } (\vec{N} \cdot \vec{L}) \qquad (1)$$

C_s : the color of the sun.
k_d : the coefficient of diffuse reflection of the surface ($0 <= k_d <= 1$).

The equation (1) constitute our starting model to calculate the color of the cloud. We let therefore a shader to simulate this model that we apply to the surface of the cloud.

This model will be little by little modified and extended along the following sections in a way to simulate the particularities of cumulus.

After having applied the model of Lambert, we modify the smooth aspect of our surface in a way to add details to it without reduce the speed of cloud's rendering.

We prefer to simulate them during the rendering, by perturbing the surface during the execution of others shaders, by the bump mapping [33]. We use Perlin noise [34] on the surface to generate a height map. Then a normal map is generated for each point of this height map.

In the moment of rendering of the cloud's surface, the vector recorded on the normal map is used to perturb the normal of this surface at each point, this perturbation can be modulated by a coefficient k_b.

$$\vec{N}_b = \vec{N} + k_b\, \vec{B}(P) \qquad (2)$$

k_b : represents the amplitude of the perturbation, more k_b is high, more the perturbation will be important.

$\vec{B}(P)$: represents the perturbation vector read from the normal map and corresponding to the point P.
We put:

$$L\,(P) = \text{saturate } (\vec{N}_b . \vec{L}) \qquad (3)$$

We modify the equation (1) to introduce the perturbation:

$$I\,(P) = k_d\, C_s\, L\,(P) \qquad (4)$$

This effect of perturbation gives a granular aspect to our surface.

4.2.2. The ambient lighting

We add to our equation of illumination an ambient term, which represents the ambient light. It is considered that in the sky, ambient light has almost the color of the sky.

Our contribution is to modify the color of the sky used by Bouthors cloud model [27] that is a simple constant.

The color of the sky is not simply blue. The sky color around the zenith is different from the color around the horizon. Moreover, the color changes depending on the time of day (i.e., the sun position). For example, in the daytime the sky looks blue, changing to white near the horizon. However, at sunset or sunrise the sky becomes red near the horizon.

We will extend this model to pass from a simply blue color to a color that is not constant, and changes according to the direction of observation, the position of the sun, the atmospheric conditions and the light reflected by the ground.

For this, we choose to inspire the sky model of Nishita [32] to calculate the sky color.

The color variations of the sky are caused by the scattering and absorption of air molecules, aerosols and other small particles in the atmosphere.

The model of Nishita [32] allows to calculate the sky color considering both Rayleigh and Mie scattering, it has the following advantages:

1) Fast calculation of single scattering.
2) Efficient calculation of the sky color incorporating multiple scattering (considering the light reflected from the ground).

As shown in Figure 6, the scattered light at a point P on the viewing ray is attenuated through the atmosphere and arrives at the viewpoint P_v.

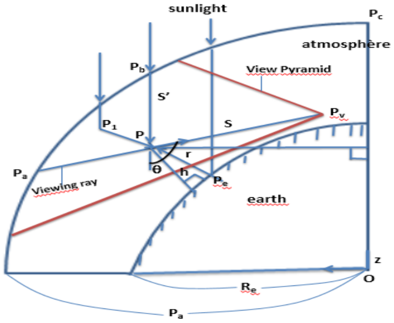

Figure 6. Optical paths for calculation of sky color.

The main optical paths to P are as follows:

(1) The sunlight arrives at point P through the atmosphere (path P_bP).
(2) The multiple-scattered light of the sunlight arrives at point P (path P_1P).
(3) The sunlight after traveling through the atmosphere and being reflected off the ground arrives at P (path P_eP).

The intensity of the light at the viewpoint can be obtained by integrating the intensity of the scattered light due to particles on the viewing ray.

4.2.2.1 Fast Calculation of Single Scattering

The intensity of light arriving at the viewpoint is the attenuated light that is scattered at the point P on the viewing ray (see Fig.6). The light arriving at the point P is also attenuated due to particles in the atmosphere. Taking into account both Rayleigh scattering and Mie scattering, the intensity at the viewpoint is calculated by the following equation.

$$I_v(\lambda) = \int_{p_v}^{p_a} I_s(\lambda)\ R(\lambda,s,\theta)\ \exp\ (-t(s,\lambda) - t(s',\lambda))\ ds \tag{5}$$

With :

$$R\ (\lambda,s,\theta) = K_r(\lambda)\ \rho_r(s)\ F_r(\theta) + K_m(\lambda)\ \rho_m(s)\ F_m(\theta)$$

λ : is the wavelength.

I_s : is the spectral distribution of extraterrestrial solar radiation.
s : is a variable for the integral (the distance between the point P and the viewpoint P_v) .
s': is the distance between the point P and the top of the atmosphere.
K_r and K_m : are coefficients for the scattering.
K_r is inversely proportional to $\lambda^4$and K_m is proportional to $\lambda^{-0.84}$: subscripts ''r'' et ''m'' stand for air molecules and aerosols respectively, and components of aerosols are omitted unless necessary in the following.
F_r : is the phase function (the function of θ shown in Fig.6) and is expressed as
$F_r(\theta) = ¾ (1+ \cos^2\theta)$.

$F_m(\theta)$ has strong forward scattering effects.

$t(s,\lambda)$: is the optical length between point P and the viewpoint.

$t(s',\lambda)$: is the optical length between the top of the atmosphere and point P.
t is expressed by the following equation.

$$t(s,\lambda) = \int_0^s (\ \beta_r(\lambda)\ \rho_r(l) + \beta_m(\lambda)\ \rho_m(l)\ )\ dl$$

$$= \beta_r(\lambda)\int_0^s \rho_r(l)\ dl + \beta_m(\lambda)\int_0^s \rho_m(l)\ dl \tag{6}$$

Where:
β_r is the scattering coeffecient (extinction coefficient). β_r is inversely proportional to λ^4.
ρ_r is the density at point P and is expressed as :

$$\rho_r = \rho_{r0}\ \exp(-h/H_r)\ ,$$

Where:
h : is the height from the ground,
ρ_{r0} : is the density at sea level
H_r : is the scale height.

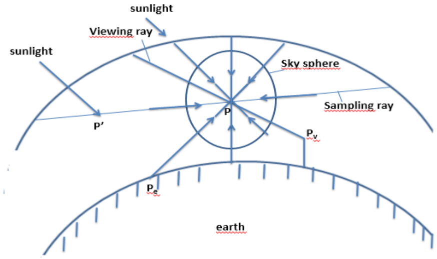

Figure 7. Optical paths for the 2nd order of scattering.

4.2.2.2 Calculation of Multiple Scattering

For the second order of scattering, the sunlight is scattered at P' incident to point P on the viewing ray, then the light scattered at P reaches viewpoint P_v (see Fig.7).

The light from various directions incident to point P. That is, the intensities from every direction w (called the sampling ray) should be gathered to P.

I_v : is the intensity reaching the viewpoint P_v, it is given by the following equation (see Fig.8).

$$I_v(\lambda) = I_s(\lambda) \int_{p_v}^{p_a} \Big\{ R(\lambda,s,\theta) \exp(- t(s',\lambda) + \frac{1}{4\pi} \int_{4\pi} dw\, F(\theta_\omega) \, (\int_{P}^{p_w} R(\lambda,s_w,\theta_{w'}) \exp(- t(s_w,\lambda) - t(s'_w,\lambda)) \, ds_w) \exp(- t(s,\lambda)) \Big\} \, ds \qquad (7)$$

Where the first term means single scattering, and the second term is the second order of scattering.

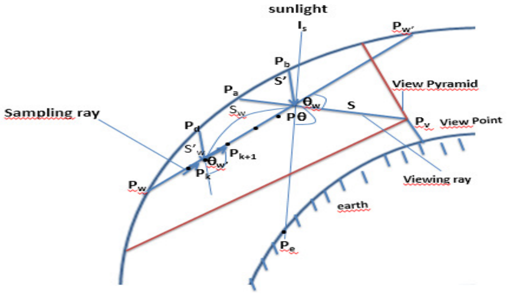

Figure 8. Sampling ray for the 2nd order of scattering

P_w : is the intersection point between the top of the atmosphere and the sampling ray .
$\theta_{w'}$: the phase angle between the sampling ray and the sunlight.
θ_ω : the phase angle between the sampling ray and viewing ray.
S_w : the distance between the sampling point on the sampling ray and P.
$t(s_w, \lambda)$: the optical depth from the sampling point on the sampling ray to point P.

The incident light to point P is assumed as the light from the sky sphere whose center is P (see Fig.7). And the sky sphere is subdivided into several directions called the sampling ray directions.

By discretizing the solid angle incident at P to M directions, equation (7) is expressed by the following equation:

$$I_v(\lambda) = K_r \; I_s(\lambda) \int_{p_v}^{p_a} F_r(\theta) \exp(- t(s', \lambda)) + \sum_{l=1}^{m} F_1 \, g_l(s, \lambda)) \;\; \rho_r(s) \;\; \exp(- t(s, \lambda)) \, ds \qquad (8)$$

where $g_l(s, \lambda)$: is the function of total transmittance of light incident to point P in sampling direction D_l, which is the total transmittance of the second order of scattering from every point on the sampling ray (PP_w), and is given by the following equation:

$$g_l(s, \lambda) = \int_{p_w}^{p} R(\lambda,s_w, \theta_t) \exp(- t(s_w, \lambda) - t(s_{w'}, \lambda)) \;\; ds_w \qquad (9)$$

$$\overline{F_l} = \frac{1}{4\pi} \int_{\Delta w} F(\theta_\omega) \, dw$$

By multiplying the intensity of sunlight I_s to equation (9) gives us the total intensity of the second order of scattering from every point on the sampling ray direction D_l. The above equation $g_l(s, \lambda)$

should be calculated at every sample point on the viewing ray. To calculate g at every sample point during the integration is a waste of time. To save the calculation time, g is precalculated and is stored in a table: the values of g in the viewing pyramid are stored.

Let's denote g at point P_k in direction D_l be $G_k^{(l)}$. $G_k^{(l)}$ is the total transmittance due to particles from P_w to P_k (see Fig.8).

After calculating the scattered intensity due to particles on path P_k P_{k+1} and the light attenuated due to the optical depth between it, $G_k^{(l)}$ at point $_{Pk}$ can be calculated by adding both of them to $G_{k-1}^{(l)}$; which is given by the following cumulative equation:

$$G_k^{(l)}(\lambda) = \mathbf{\Delta}\, G_k^{(l)}(\lambda) + G_{k-1}^{(l)}(\lambda) \exp(-\tau_\kappa(\lambda)) \tag{10}$$

$$\mathbf{\Delta}\, G_k^{(l)}(\lambda) = \int_{p_{k-1}}^{p_k} R(\lambda,,s,\theta) \exp(-\,t(s,\lambda) - t(s',\lambda))\, ds$$

τ_κ : is the optical depth of the path P_{k-1} P_k and is obtained when $\mathbf{\Delta}\, G_k^{(l)}$ is calculated.
S : is the distance from a sampling point to P_{k-1}
s' : is the distance from the top of the atmosphere to the sampling point.

We modify equation (4) to introduce the ambient factor:

$$I\,(P) = I_v + k_d\, C_s\, L\,(P) \tag{11}$$

4.2.3 The Corona

We modify then our model in order to simulate the effect of the reflections occurring in the corona of the cloud illuminated by the sun (forward scattering).

We will be able to know if the point that we draw is on the corona or not, due to the calculation of the density ρ (the low density of the water particles is at the corona).

We use a technique already studied by Antonie Bouthors [27]. This method consist to apply a transparency on the surface according to the density of the cloud being behind the point considered.

For this we assume that our blobs are spherical. Due to this approximation, it is very easy to determine locally what is the position of point P relative to the blob on which it is.

The normal $\vec{N}$ in the coordinates of camera inform us directly on the position of P on the sphere:

If this normal is directed toward the camera, P is in the middle of the circle representing the sphere on the screen. This means that the line D starting from the camera and passing by P, crosses the sphere by the possible longest path inside the sphere.

If this normal is orthogonal to the direction of the camera, P is on the contour of the circle representing the sphere on the screen, so the line D does not cross the sphere, but tangent with it. We can deduce the crossed length by D inside our sphere with radius R_i :

$$l_i(P) = 2\, R_i\, \vec{N}\, \vec{O} \tag{12}$$

This length will allow us to calculate analytically the integral of the density of the cloud along the radius. But for that, we need to know how is the density inside a cloud.
We define the maximum density attainable by a cloud as 1, and 0 as the nil density.

We simulate the distribution of density inside our sphere of cloud as follows:

All the points located at a distance less to $R_i - r$ from the center are considered to have a density 1. Between $R_i - r$ and R_i, the density decreases linearly until 0.

This gives us the density $\rho_i(P)$ of a point P:

$$\rho_i(P) = \text{saturate}\left(\frac{R_i - d_i}{r}\right) \qquad (13)$$

From equations (12) and (13) we have all the elements to calculate an analytical integral of the density crossed by D.
We assume that the density is constant throughout D inside the sphere.

The integration will be summarized then with a simple multiplication, at the price of a decreased in realism.

We can calculate the minimum distance between the line D and the center of the sphere:

$$d_i^{min}(P) = R_i\sqrt{1 - (\vec{N}\,\vec{O})^2} \qquad (14)$$

At this minimum distance d_i^{min} is associated the maximum density $\rho_i^{max}(P)$ crossed by D in the sphere:

$$\rho_i^{max}(P) = \text{saturate}\left(\frac{R_i - d_i^{min}}{r}\right) \qquad (15)$$

The amount $\widehat{\rho_\iota}(P)$ of droplets crossed by a ray is the integral of the density ρ along this ray. We approximate it by the product of the length *l* of the ray inside the cloud and the density □. We consider that all the points traversed by D in the sphere have this maximum density $\rho_i^{max}(P)$.

$$\widehat{\rho_\iota}(P) \approx \rho_i^{max}(P)\; l_i(P) \qquad (16)$$

$$\approx l_i(P)\ \text{saturate}\left(\frac{R_i}{r}\left(1 - \sqrt{1 - (\vec{N}\,\vec{O})^2}\right)\right) \qquad (17)$$

We add the factor k_ρ, to adjust the variation of density between the core of the cloud and the corona.

$$\rho = k_\rho\; l_i(P)\ \text{saturate}\left(\frac{R_i}{r}\left(1 - \sqrt{1 - (\vec{N}\,\vec{O})^2}\right)\right) \qquad (18)$$

Due to the density ρ , we can calculate the amount of light diffused by the sun in the direction of the observer through the cloud at the point P. Indeed, we assume that the phase function of the cloud is a Gaussian whose the standard derivation depends on the crossed density. We approximate this Gaussian by a power of cosine :

$$T_i(P) = e^{-\frac{(-\vec{O}.\vec{L})^2}{2\rho^2}} \qquad (19)$$

$$e^{-\frac{x^2}{2\sigma^2}} \approx \cos(x)^{\frac{1}{\sigma^2}} \quad (20)$$

From the equations (19) and (20) :

$$\mathrm{T_i(P)} \approx \cos(-\vec{O}.\vec{L})^{\frac{1}{\rho^2}} \quad (21)$$

The amount of light transmitted through the corona is :

$$\mathrm{T_i(P)} \approx \cos(-\vec{O}.\vec{L})^{\frac{k_g}{\rho^2}} \quad (22)$$

k_g : is a term allows us to adjust the width of the Gaussian used.

We modify equation (11) to take into account the illumination of the corona :

$$\mathrm{I\ (P)} = \mathrm{I_v} + \mathrm{C_s}\ (k_t\ \mathrm{T(P)} + k_d\ \mathrm{L\ (P)}\ ) \quad (23)$$

K_t : is a term allows us to adjust the intensity of the transmission term.

4.2.4 The Light Traps

The cumulus clouds are very structured (like a cauliflower, with large ellipsoidal heaps covered with bubbles, themselve covered with tiny bubbles), it is possible to estimate the location and geometry of the concavities (eg, between the bubbles), this concavities on the surface of the cloud act as light traps.

We modify our model to simulate the effect of multiple reflections occurring in the furrows of the cloud illuminated by the sun.

We know that the furrows correspond exactly to the local minima of d_{l-1} (the distance from point P to the surface S_{l-1}). We will increase the brightness of the points with a low d_{l-1}.

$$\mathrm{F\ (P)} = \mathrm{saturate}\ (1 - k_p d_{l-1}) \quad (24)$$

k_p :control the width of light traps, more k_p will be important and more the intensity added by F(P) decrease rapidly with d_{l-1}, and therefore the light provided by the multiple reflections will be limited to the hollow furrows, more k_p is low, the multiple reflections will have a larger area of effect.

The increasing of the brightness in all the furrows would have an unrealistic effect, only the furrows facing the sun are illuminated.

We can access to the normal $\vec{N}_{l-1}$ of the point of S_{l-1} the more nearly to P.

We can determine if the furrows faces the sun or not with the orientation information $\vec{N}_{l-1}$.

$$\mathrm{F\ (P)} = \mathrm{saturate}\ (1 - k_p\ d_{l-1})\ \vec{N}_{l-1}\ \mathrm{(P)}\ \vec{L} \quad (25)$$

As our model of form is also based on several levels, we can make use of it to take into account this phenomenon on all the scales.

$$\mathrm{F\ (P)} = \sum_{j=1}^{n} \mathrm{saturate}\ (1 - k_p\ d_{l-1})\ \vec{N}_{l-j}\ \mathrm{(P)}\ \vec{L} \quad (26)$$

Finally, We modify equation (23) to take into account the The Light Traps

$$I(P) = I_v + C_s (k_t \, T(P) + k_d \, L(P) + k_f \, F(P)) \quad (27)$$

5. RESULTS

We implemented the illumination method of clouds of Antoine Bouthors by applying the modifications that we developed during section 4. The images presented in the Figure 9 show a clouds implemented by our model. Indeed, on our implementation, the speed rendering is at least as faster than Bouthors's model, and the results obtained are more realistic than the previous models, particularly in terms of the richness of detail.

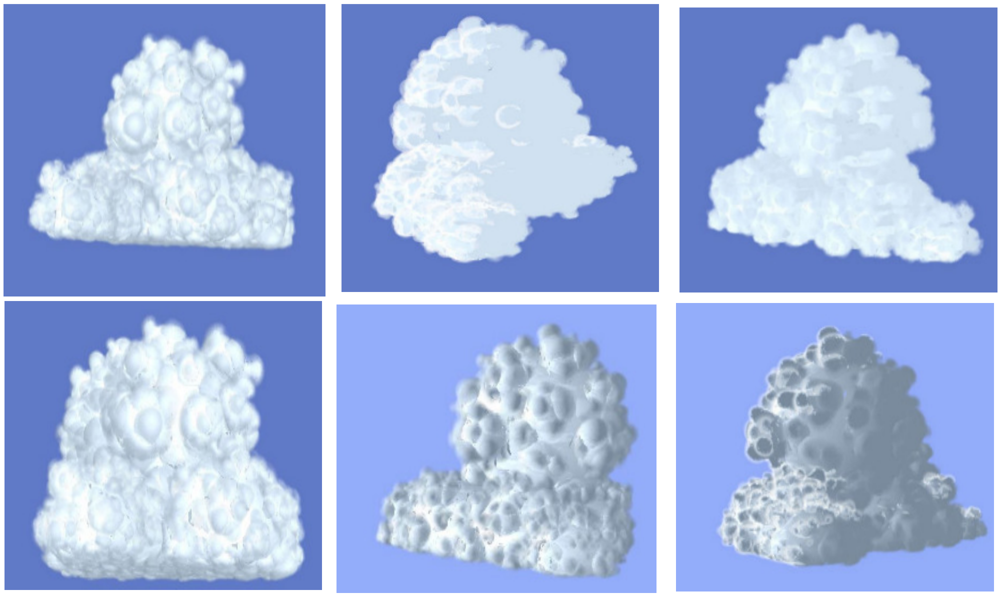

Figure 9. Resulting images of our model for various points of view and various sun positions.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a model (shape and visual aspect) based on a phenomenological method for modelling and rendering a cumulus cloud. Since our choice was to represent the cloud by a surface and do not rely on a volumetric representation, the rendering of this surface is really fast and gives results at least as realistic as the previous models, with an interactive speed. However, our model is limited only to the cumulus clouds.

Many improvements may be possible to our model:

The texture used for our surface has currently a bad lacunarity, which makes the result not very realistic, the plating of the height map and the normal map could also be improving.

An approximation which would also need to be improved is the calculation of ρ, which considers for the moment that the density on the path of D is the maximum density encountered and considers that our blobs are spherical, which is not true in reality.

REFERENCES

[1] James F. Blinn, (1982) “Light reflection functions for simulation of clouds and dusty surfaces”, Computer Graphics Vol. 16, No. 3, pp21–29.

[2] Nelson Max, (1986) “Light diffusion through clouds and haze”, Computer Vision, Graphics, and image Processing 33, pp280–292.

[3] James T. Kajiya & Brian P. Von Herzen , (july 1984) “Ray tracing volume densities ”, Computer Graphics (july 1984), pp168-174.

[4] Richard Voss, (1983) “Fourier synthesis of gaussian fractals: 1/f noises, landscapes, and flakes”, Tutorial on State of the Art Image Synthesis, ACM Siggraph Course Notes .

[5] Kazufumi Kaneda, Takashi Okamoto, Eihachiro Nakamae &Tomoyuki Nishita, (1990) “Highly Realistic Visual Simulation of Outdoor Scenes under Various Atmospheric Conditions”, Springer-Verlag, Tokyo pp117– 131.

[6] Nelson Max, Germany (June 1994) “Efficient light propagation for multiple anisotropic volume scattering”, In Fifth Eurographics Workshop on Rendering. Germany pp87–104.

[7] Tomoyuki Nishita, Eihachiro Nakamae & Yoshinori Dobashi (August 1996) “Display of clouds taking into account multiple anisotropic scattering and sky light ”, In Holly Rushmeier, editor, SIGGRAPH 96 Conference Proceedings.ACM SIGGRAPH, Addison Wesley pp379–386.

[8] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita & Tomoyuki Nishita, (July 2000) “A simple, efficient method for realistic animation of clouds”, In Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series, pp19–28.

[9] Mark J. Harris, William Baxter III, Thorsten Scheuermann and Anselmo Lastra, California (July 2003) “Simulation of cloud dynamics on graphics hardware,” in Graphics Hardware, pp. 92–101.

[10] Holly E. Rushmeier and Kenneth E. Torrance, New York (July 1987) “ The zonal method for calculating light intensities in the presence of a participating medium,” in Computer Graphics (Proceedings of SIGGRAPH 87), VOL. 21, pp293–302.

[11] David Ebert, (August 1997) “Volumetric procedural implicit functions: A cloud is born,” in SIGGRAPH 97 Technical Sketches Program.

[12] Jos Stam and Eugene Fiume, (August 1993) “Turbulent wind fields for gaseous phenomena,” in James T. Kajiya, editor, Computer Graphics (SIGGRAPH ’93 Proceedings), Vol. 27, pp369–376.

[13] Jos Stam and Eugene Fiume, (August 1995) “Depicting fire and other gaseous phenomena using diffusion processes,” in Robert Cook, editor, SIGGRAPH 95 Conference Proceedings, pp129–136.

[14] Jos Stam and Eugene Fiume, (June 1991) “A multiple-scale stochastic modelling primitive,” in Proceedings of Graphics Interface 91, 24–31.

[15] David Ebert, Kent Musgrave, Darwyn Peachey, Ken Perlin, and Worley, (October 1994), “Texturing and Modeling : A Procedural Approach, ” Academic Press.

[16] Richard Voss, (1985) “Fractal forgeries,” In R. A. Earnshaw, editor, Fundamental Algorithms for Computer Graphics, Springer-Verlag.

[17] Geoffrey Y. Gardner ,(July 1984) “Simulation of natural scenes using textured quadric surfaces,” In Hank Christiansen, editor, Computer Graphics (SIGGRAPH ’84), Vol. 18, pp11–20.

[18] Geoffrey Y. Gardner, (July 1985) “Visual simulation of clouds,” in B. A. Barsky, editor, Computer Graphics (SIGGRAPH ’85 Proceedings), Vol 19, pp297–303.

[19] Pantelis Elinas and Wolfgang Stuerzlinger, (2000) “Real-time rendering of 3D clouds,” Journal of Graphics Tools .

[20] Neyret Fabrice, (May 2000) “A phenomenological shader for the rendering of cumulus clouds,” Technical Report RR-3947, INRIA.

[21] Andrzej Trembilski and Andreas Broßler, (2002) “Surface-based efficient cloud visualisation for animation applications,” in WSCG pp453–460.

[22] Simon Premože, Michael Ashikhmin, and Peter Shirley, (2003) “Path integration for light transport in volumes,” in Eurographics Symposium on Rendering (EGSR) pp52–63.

[23] Simon Premože, Michael Ashikhmin, Jerry Tessendorf, Ravi Ramamoorthi, and Shree Nayar, (June 2004) “Practical rendering of multiple scattering effects in participating media,” in Eurographics Symposium on Rendering (EGSR) pp363–374.

[24] Kyle Hegeman, Michael Ashikhmin, and Simon Premože,(2005) “A lighting model for general participating media,” in ACM SIGGRAPH Symposium on Interactive 3D graphics pp117–124.

[25] Mark J. Harris and Anselmo Lastra, (2001) “Real-time cloud rendering,” Computer Graphics Forum 20, 3 .

[26] Kirk Riley, David S. Ebert, Martin Kraus, Jerry Tessendorf, and Charles Hansen, (2004) "Efficient Rendering of Atmospheric Phenomena," in Eurographics Symposium on Rendering pp375–386.

[27] Antoine Bouthors, (2004) "Rendu réaliste de nuages en temps reel".

[28] Antoine Bouthors, Fabrice Neyret, and Sylvain Lefebvre, (september 2006) "Real-time realistic illumination and shading of stratiform cloud," in Eurographics Workshop on Natural Phenomena.

[29] Antoine Bouthors, Fabrice Neyret, Nelson Max, Eric Bruneton and Cyril Crassin, (February 2008) "Interactive multiple anisotropic scattering in clouds," in ACM Symposium on Interactive 3D Graphics and Games, i3D, New York, NY, USA pp173–182.

[30] Laurent Belcour and Eric Bruneton (2009) "Rendu Temps-Réel de Nuages à l'Echelle Planétaire," Rapport de stage de Master de Recherche en Informatique Graphique, Vision et Robotique.

[31] Oskar Elek, Tobias Ritschel, Alexander Wilkie and Hans-Peter Seidel (May 2012) "Interactive Cloud Rendering Using Temporally- Coherent Photon Mapping," Graphics Interface Conference. Toronto, Ontario, Canada , pp141-148.

[32] Tomoyuki Nishita, Eihachiro Nakamae and Yoshinori Dobashi, (December 2000) " Display method of the sky color taking into account multiple scattering," in SIGGRAPH '96, pp117-132.

[33] James F. Blinn,, (August 1978) "Simulation of wrinkled surfaces," In Computer Graphics (Proceedings of SIGGRAPH 78), Vol. 12, pp286–292.

[34] Ken Perlin, (July 1985) "An image synthesizer," in B. A. Barsky, editor, Computer Graphics (SIGGRAPH '85 Proceedings) Vol. 19(3), pp287–296.

[35] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen, (Aug. 2001) "Visual simulation of smoke," In Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, pp15–22.

[36] Ryo Miyazaki, Satoru Yoshida, Tomoyuki Nishita and Yoshinori Dobashi, (October 2001) "A method for modeling clouds based on atmospheric fluid dynamics," in 9th Pacific Conference on Computer Graphics and Applications, pp363–372.

[37] Joshua Schpok, Joseph Simons, David S. Ebert and Charles Hansen, (july 2003) "A real-time cloud modeling, rendering, and animation system," Symposium on Computer Animation'03, pp 160–166.

[38] David S. Ebert and Parent R, (1990) "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-Buffer Techniques," Computer Graphics Vol. 24,No. 4. pp357–366.

[39] Jos Stam and Eugene Fiume, (June 1991) "A multiple-scale stochastic modelling primitive," in Proceedings of Graphics Interface '91, pp24–31.

[40] Yoshinori Dobashi, Tomoyuki Nishita, Hideo Yamashita and Tsuyoshi Okita, (October 1998) "Modeling of clouds from satellite images using metaballs," in Pacific Graphics '98,.

[41] Yoshinori Dobashi, Yusuke Shinzo and Tsuyoshi Yamamoto, (2010) "Modeling of clouds from a single photograph," Computer Graphics Forum 29, 7, 2083–2090.

[42] Yoshinori Dobashi, Wataru Iwasaki, Ayumi Ono, Tsuyoshi Yamamoto, Yonghao Yue and Tomoyuki Nishita, (2012) "An Inverse Problem Approach for Automatically Adjusting the Parameters for Rendering Clouds Using Photographs,"