INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Phenomenological Shader for the Rendering of Cumulus Clouds

Fabrice Neyret
iMAGIS - GRAVIR-IMAG/INRIA
Fabrice.Neyret@imag.fr
http://www-imagis.imag.fr/Membres/Fabrice.Neyret/

No 3947

mai 2000

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--3947--FR+ENG

A Phenomenological Shader for the Rendering of Cumulus Clouds

Fabrice Neyret
iMAGIS - GRAVIR-IMAG/INRIA

Fabrice.Neyret@imag.fr
http://www-imagis.imag.fr/Membres/Fabrice.Neyret/

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet iMAGIS

Rapport de recherche n°3947 — mai 2000 — 15 pages

Abstract: This paper presents a method for the efficient realistic shading of cumulus clouds. Our approach consists in taking advantage of all the *a priori* knowledge on the characteristics of such objects and on the light effects that are known to occur, thus avoiding a time consuming (blind) physical simulation of multiple scattering through volumes. This paper only focuses on cloud shading. It does not deal with the modeling and animation of the cloud's shape, or with sky models. The features we take into account are the illumination, including the effects of the inter-reflections and of the environment, and the cloud's corona behavior, especially when illuminated from the back. We show some results obtained with our preliminary implementation, illustrating these features. All the images are ray-traced in about $1\frac{1}{2}$ minute on a SGI $O2$.

Key-words: image synthesis, natural phenomena, rendering, clouds.

(Résumé : tsvp)

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International: +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International: +33 4 76 61 52 52

Un Shader Phénoménologique pour le Rendu des Nuages Convectifs

Résumé : Ce papier présente une méthode de rendu réaliste efficace du shading des nuages convectifs (i.e. les cumulus). Notre approche consiste à exploiter toute la connaissance a priori sur les caractéristiques de ces objets et des effets lumineux qui les concernent, afin d'éviter une simulation physique coûteuse en temps (car aveugle) des réflections multiples dans le volume. Ce papier ne concerne que le shading des nuages. Il ne traite pas de la modélisation et de l'animation de la forme des nuages, ni des modèles de ciel. Les caractéristiques que nous prenons en compte sont l'illumination, incluant les effets des inter-réflections et de l'environnement, et le comportement de la corolle du nuage, en particulier quand elle est rétro-éclairée. Nous montrons quelques résultats obtenus avec notre implémentation préliminaire, illustrant ces caractéristiques. Toutes les images sont ray-tracées en environ $1\frac{1}{2}$ minute sur une SGI $O2$.

Mots-clé : synthèse d'images, phénomènes naturels, rendu, nuages.

1 Introduction

Synthesizing images of realistic clouds is a difficult task, whether one looks at their shape, their animation or their rendering. Physical simulations imply processing volumetric rendering as well as radiosity and anisotropic scattering. This must be done in a wide and highly sampled area, thus yielding huge rendering cost. Among the various cloud types, cumulus are pleasant to observe, common in the summer sky and strongly structured. Thus, people have a strong intuition about what cumulus clouds should look like, intuiting phenomenological 'laws' about the shading, which can be validated according to the characteristics of this very cloud type. E.g. the cloud's silhouettes are enlighted when they are illuminated from the back, while fading to sky color when illuminated from the front; the main effect of radiosity is to amplify the illumination in the concavities of the surface; the shadowed areas are unsatured brown on the bottom of the cloud and slightly blue elsewhere, etc. This set of knowledge can be used to avoid a complex simulation, by accounting directly for the useful appearance features. We propose such a shading model (while leaving the issue on cumulus shape), which accounts efficiently for back and front illumination, radiosity and environment.

This paper is structured as follows: Section 2 reviews previous work on cloud rendering. Section 3 describes the structural and visual characteristics of cumulus clouds, in order to establish what really needs to be simulated. Section 4 lists and motivates our hypothesis and choices in order to model the various features of the reflected and transmitted illumination. Section 5 presents our preliminary results, and discusses future works.

2 Previous Works

A lot of work related to clouds has been published to date, so we will not survey all of it. First, many aspects are addressed, from shape and animation to global illumination, anisotropic scattering and sky color, including real-time adaptations. Moreover, many kind of clouds have been studied, from haze to stratus, in the scope of applications requiring either very large (satellite view) or very close scales (inside a single cloud). A majority of these techniques are, explicitly or not, targeted to fuzzy clouds, which cumulus are globally not.

Blinn [1] and Max [9] have proposed early models to account for light reflection and diffusion through clouds. Kajiya [8] introduced the idea of volumetric rendering, together with the first attempt of a numerical fluid simulation to produce the shape of CG clouds. Max [10] has extended the rendering simulation to take into account the anisotropic scattering inside the cloud, getting a nice enlighted corona when the light comes from the back. A lot of work has been done on volumetric radiosity in general, starting with the zonal method by Rushmeier and Torrance [16]. These techniques and their variants have sometimes been used to render cloud material, such as for the images by Ebert [3]. Nishita and Nakamae [13] have gathered anisotropic scattering, radiosity and sky illumination, getting very nice results (however their cumulonimbus are probably too transparent). Several papers propose models for sky illumination, but their review is out of the scope of this paper. Stam [18, 19] 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 [17]. In particular, he introduces an efficient method to compute light diffusion inside clouds [19].

Another aspect is the use of textures, fractals and shaders to represent the details. Musgrave [2] has extensively used them for cloudy skies. Gardner [4, 5] 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 physical models simulating multiple anisotropic scattering usually reproduce very nicely the light effects occuring for cumulus. But as we will see in the next section the precision of the process they simulate is over-sized in the case of cumulus clouds regarding the visible features real ones show, that are predictable. These simulations are thus uselessly costly in time, and often require the use of voxels having a low resolution compared to the scale of details that are perceptible on real clouds. Textural models are interesting for the details they provide, but do not address the illumination aspect that is essential for clouds that are highly reflective. They are cheap as long as they are sampled on a surface, but get costly when sampled in volume. That is our motivation to propose a model that engages means proportioned to needs, concentrated on visible features, which we list in the following section.

3 Characteristics of Cumulus Clouds

Cumulus clouds have very specific characteristics, since they are the product of unstable air. As such, wet and dry air are almost not mixed, which makes the mature clouds compact and looking almost concrete, with quite a well defined surface. The shape of this surface is formed by the rising thermals and the turbulence, which both induce this characteristic cauliflower pattern. A balance between the shearing (induced by the growth) and the mixing maintains a constant 'blurry' layer between the dry air and the core of the cloud that is as thin as approximately 10 meter [6, 7], laying on 'bubbles' that are roughly 100 meter large. A very young cumulus can be made of a single bubble (a thermal having risen up to the dew point altitude), thus not yet having the sharp properties mentioned above.This paper is rather targeted at rendering mature cumulus or cumulonimbus, which can be several kilometer large and high.

A common characteristic of every kind of cloud is that the albedo is very close to 1 (there is almost no absorption), and that the droplets induce a high forward scattering. In the particular case of cumulus clouds, that are dense and thick, this turns into several other unique and essential characteristics: the scattering at large scale is roughly isotropic (we motivate this in Appendix A), except in the thin external layer that is visible on the silhouette. As such, the diffusion through clouds can be approximated by models coming from thermal conduction, which yield fields with gradient values between the boundary conditions. In our case, boundary conditions consist of the illumination received at each location of the cloud surface, coming from the sun, the sky, the ground and the other clouds (or other visible parts of the same cloud). A consequence is that the cloud surface roughly re-emits the same amount of light it has received in the neighborhood of each location, isotropically, much like if is were simply reflecting the incoming light. Taking into account the apparent size of surface elements, this yields to... the Lambert reflectance model !

Figure 1: The illumination aspect of real clouds.

The next consequence is that radiosity plays an important role: the light re-emitted by a surface element of the surface has almost the same energy than the received one. 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. Thus inter-reflections are almost not absorbed, and concavities on the cloud surface (e.g. between bubbles) behave like 'light traps': all the energy received is re-emitted in a narrow solid angle instead of 2π, which can make some concavities be almost as bright as the sun ! (see figures 2 and 4,middle). This sometimes gives the impression than the cloud is illuminated from the inside, as the interior of 'faults' seems to be far brighter than the surface around. This multiplicative behavior can also arise, despite less intense, in areas that are not directly illuminated by the sun. Moreover, this occurs whatever the scale of the concavities, from very local to cloud-size.

Figure 2: Light amplification by concavities on real clouds.

Lastly, the cumulus corona behaves very differently from the core of the clouds, as figured on the scheme 4, left. Because of the light density, the cumulated scattering remains narrow and forward (oppositely to the case of diffuse clouds such as stratus, young cumulus or haze, for which the scattering remains highly directional everywhere). This makes cumulus looking very contrasted: when illuminated from the front, the body is intensely white and the corona is almost transparent[1]. When illuminated from the back, the core is dark (and homogeneously dark for thick cumulus) while the corona is illuminated (sometimes intensely) due to forward scattering, as illustrated on figure 3. As a remark, the light re-emitted in shadowed areas consists of a little remaining transmitted contribution, but mainly from reflection of the sky and ground, and more likely from the reflection of light by other parts of the cloud.

[1] Indeed it is not really: is has roughly the sky color. This appears clearly with an enhancement of the silhouette when another cloud is behind.

Figure 3: Enlighted corona of real clouds.

4 Our Model

Our key choice is to represent the cloud by a **surface**. The 'volumetric looking' aspects are figured by our shading model for the illumination features, and by a procedural texture model for the distribution of matter: in the spirit of [4, 5], we make the surface look more transparent at grazing angles, and we use Perlin noise [14] on the surface to modulate the density. Despite the rendered geometry is a surface, images in Gardner's papers have demonstrated that it gives a very convincing illusion of volumetric material. Our contribution is the addition of an illumination model to this approach.

4.1 Basic Illumination

The considerations above encourage us to keep the Lambert model for the cloud 'reflection' outside the corona, after having perturbated the surface normal with Perlin noise as stated above. We add to it an ambient term (that can be adjusted depending of the size of clouds) to take into account some remaining transmission. This ambient term also incorporates the light coming from the environment: the sky, the ground and other clouds. This component is modeled using a spherical 'lightmap' (which is 1D in our implementation). The color of the light map is brown near the ground, blue sky everywhere else, and white light is added near the horizon. A sky color model could easily be incorporated here. Together with color, the intensity in the map should be computed from an estimation of the environment: reflection of the ground decreases with the cloudiness of the sky (which makes the thunderstorm clouds so oppressive), and the amount of white light should follow the density of surrounding clouds in the same way.

4.2 The Corona

The amount q of droplets crossed by a ray is the integral of the density along this ray. We approximate it by the product of the length l of the section of the ray inside the cloud and the density ρ at the deeper location (roughly at the middle of the section).We assume that the density varies linearly from 0 to 1 across the corona layer that is 10 meter thick. Whatever the surface model is, we can make the hypothesis than the surface is locally spherical with a characteristic radius (the bubbles are typically 100 meter large). Then, from the (unperturbated) normal at the hit location we can estimate both the section length $l \sim 2R(N.O)$, and the deepest location reached in the layer (see figure 4, left). These estimations can be adjusted to increase the precision, depending of the model for the surface[2]. From the depth we get the maximum density ρ along the section, that is $\rho \sim \frac{R}{dR}(1-\sqrt{1-(N.O)^2})$ with N the normal to the bubble at the hit location, O the direction of the observer, $(A.B)$ the dot product of A and B, R the characteristic bubble radius[3] , dR the layer thickness. If this depth is less than the layer thickness, the ray crosses only the corona. If so, we estimate the global scattering according to Appendix A: the phase function is figured by a Gaussian whose standard deviation σ^2 is proportional to the amount of droplet crossed. It is known that Gaussians are very well approximated by power of cosines (e.g. this is used for Phong shading): $e^{-\frac{1}{2}(\frac{x}{\sigma})^2} \sim \cos(x)^{\frac{1}{\sigma^2}}$. Thus we approximate the scattering diagram by the cosine of the angle between $-L$ and O, with L the direction of the sun, powered by $\frac{K}{\sigma^2}$. The coefficient K is $\frac{2\sqrt{2R\ dR}}{\sigma^2_{max}}$, which allows us to adjust the effective scattering σ^2_{max} at the deepest part of the corona. The density ρ is modulated by Perlin noise as stated at the beginning of 4. The amount of scattered light is thus $I\rho\cos(-O.L)^{\frac{K}{\sigma^2}}$

The smooth transition between the corona and the core of the cloud is achieved by the linear interpolation between the basic Lambert illumination and the scattered illumination, according to the value of ρ (as it varies from 0 to 1 through the corona).

[2]In a first version of our implementation we were using implicit surfaces to model the cloud shapes. With this model it was easy to know the section length, but we failed at estimating both satisfactorily and cheaply the maximal depth. In our current version we use ellipsoids (whose shading is blended when two intersections are closer than the layer thickness), in which case the 'approximation' is exact.

[3]Even if the details of the surface are thinner, we consider here that the object of interest is the underlying sphere.

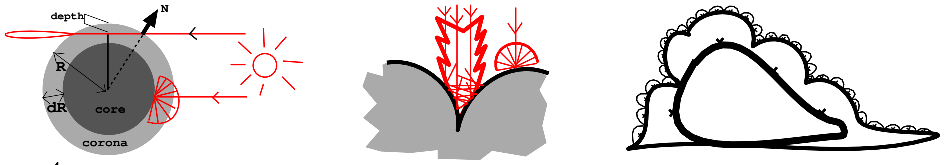

Figure 4: *Left:* result of cumulated scattering on the cloud core and on the corona. *Middle:* amplified reflection in a concavity of the surface. *Right:* 3 scales of shapes laying on each other.

4.3 The Radiosity 'Light-Traps'

As stated in section 3, radiosity between facing locations in the concave areas of the cumulus surface is very important, and can even represent the main contribution of the re-emission. As 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, and so to estimate the radiosity at these locations. We have chosen to associate an explicit surface for each scale, over which lay smaller shapes (that are roughly spheres), as illustrated on figure 4, right. The underlying surface is triangulated, in such a way that vertices correspond to the location of smaller shapes. This way, the concavity location corresponds to the center of the faces.

The local geometry is then almost symmetrical around the face center, so we estimate the radiosity from this location (see figure 5,left). If the concavity were hemispherical, assuming that each surface element can see the same solid angle Ω of sky, and can see all the other surface elements equally, one can easily show that the energy that is re-emitted by a surface element would be $\frac{2\pi-\Omega}{\Omega}I_{mean}+I_{loc}$ with I_{loc} the local (direct) illumination model, and I_{mean} the average value of I_{loc} in the concavity.

For other geometries, one has to estimate an analytical integral to get the multiplicative factor of I_{mean} (that generally depends on the location). We tried some complicated ones, however we finally found that an empirical law can do as well: we use $\frac{\arctan(k(1-h))}{\arctan(k)}$ with h the height of the location in local normalized coordinates, i.e. $(N_{sh}.\frac{(P-P_{sh})}{|P-P_{sh}|})$ with P_{sh} the center location of the small (spherical) shape, P the considered location on it, and N_{sh} the 'global' orientation of the shape, that corresponds to the normal of the underlying shape at the vertex P_{sh}. The coefficient k allows us to tune the expansion of the enlighted area, that is connected to the sharpness of the concavity (we use 4 in our current implementation).

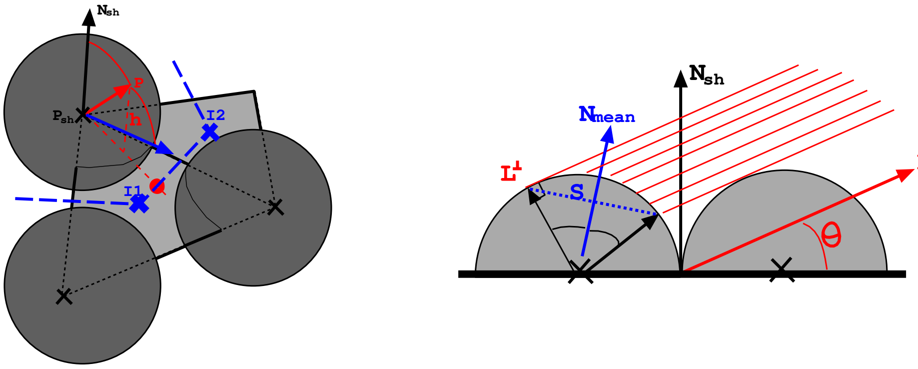

Figure 5: *Left:* 3 spherical shapes associated to the 3 vertices of a face of the underlying surface. The inter-reflection estimation on point P to be shaded is interpolated from $I1$ and $I2$. For clarity, the shapes are figured as small spheres, while they don't need to be (e.g. we use ellipsoids), and they should probably blend into each other without letting a gap between them. *Right:* the mean normal N_{mean} on the illuminated area, and the associated surface S. in 2D.

The mean direct illumination I_{mean} is equal to $amb + S * (N_{mean}.L)$, with N_{mean} the average of the surrounding normals that receive the sun light, and S the corresponding surface. We estimate it in 2D with circles instead of spheres (see figure 5,right), by computing the two radial vectors bounding the illuminated area on a circle (this computation has similarities with the model of anisotropy of [15]). N_{mean} is the halfway vector, and S is approximately the chord between them. If θ is the angle between L and the horizontal, i.e. $L = (\cos(\theta), \sin(\theta))$, we can see that $N_{mean} = \sqrt{\sin(\theta)}\ L + \sqrt{1-\sin(\theta)}\ L^{\perp}$ with $L^{\perp}$ the perpendicular of L, i.e. $(-\sin(\theta), \cos(\theta)$. Thus $(N_{mean}.L) = \sqrt{\sin(\theta)} = \sqrt{(N_{sh}.L)}$. The surface $S \sim 2\sqrt{1-(N_{sh}.L^{\perp})^2} = \sqrt{\sin(\theta)} = \sqrt{(N_{sh}.L)}$. This means that we have $S * (N_{mean}.L)$ approximatively equal to ... $(N_{sh}.L)$, the Lambert law !

Thus we proceed as follow: before rendering the scene, we compute for each face of the underlying surface the mean direct illumination I_{mean}, which is in fact equal to the classical Lambert law (including the ambient term). Then, when we need to compute the shading at a given hit point during the rendering, we have to sum the local illumination and the contribution of the inter-reflections. We estimate this last term by interpolating I_{mean} between the values at the centers of the two closest faces of the underlying surface, then multiplying it by the multiplicative function mentioned above (the arctangent in our implementation).

To find quickly the adequate pair of centers to use and the right interpolation coefficients, we rely on a data structure that lists the faces which are connected to a given vertex (which is the location P_{sh} of the center of the spherical shape, see scheme 5,left), and the edges (plus some precomputed values). The two faces to consider are the ones corresponding to the edge whose dot product with the direction $P - P_{sh}$ is maximal.

The same process can be used at the various scales. In our present implementation, we only manage one scale[4], i.e. one underlying surface covered by spherical shapes (that are blended ellipsoids).

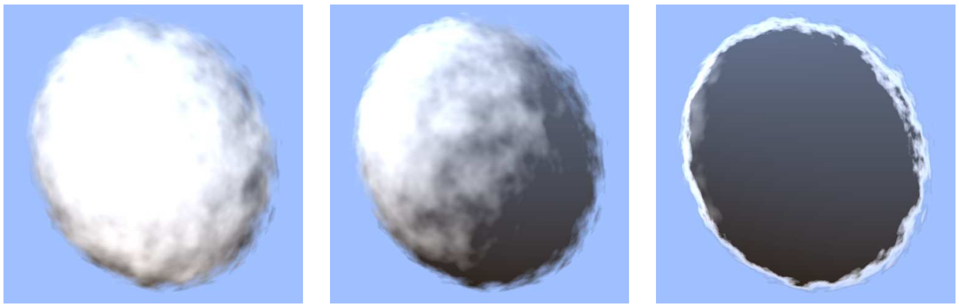

Figure 6: A single ellipsoid with various sun positions. Note the difference of shading between the core and the corona.

5 Results

Figure 6 presents the light effects on a single ellipsoid with various sun positions. For the images on Figure 7, 87 ellipsoids are used (so the underlying shape has 170 faces, that we have produced using the Turk algorithm [20] in order to get regular triangles). The images have been computed using a non-optimized ray-tracer, on a SGI $O2$ with a 180 Mhz R5000 processor, at resolution 600×600.

The rendering time is 100 seconds, with roughly 22% spent in the classical ray-tracing plus Lambert evaluation, 3% in the precalculations for the estimated radiosity, 8% in the shadow calculation, 12% in the density modulation for the fake volumetric layer, and 55% for the Perlin noise evaluation.

One can see that the overhead of our shading model is almost neglectible according to the basic rendering 'ala Gardner' without advanced shading.

[4]Managing also a larger scale would not increase significatively the rendering cost, but it would have to rely on a cloud geometrical model, which we have not in our current implementation!

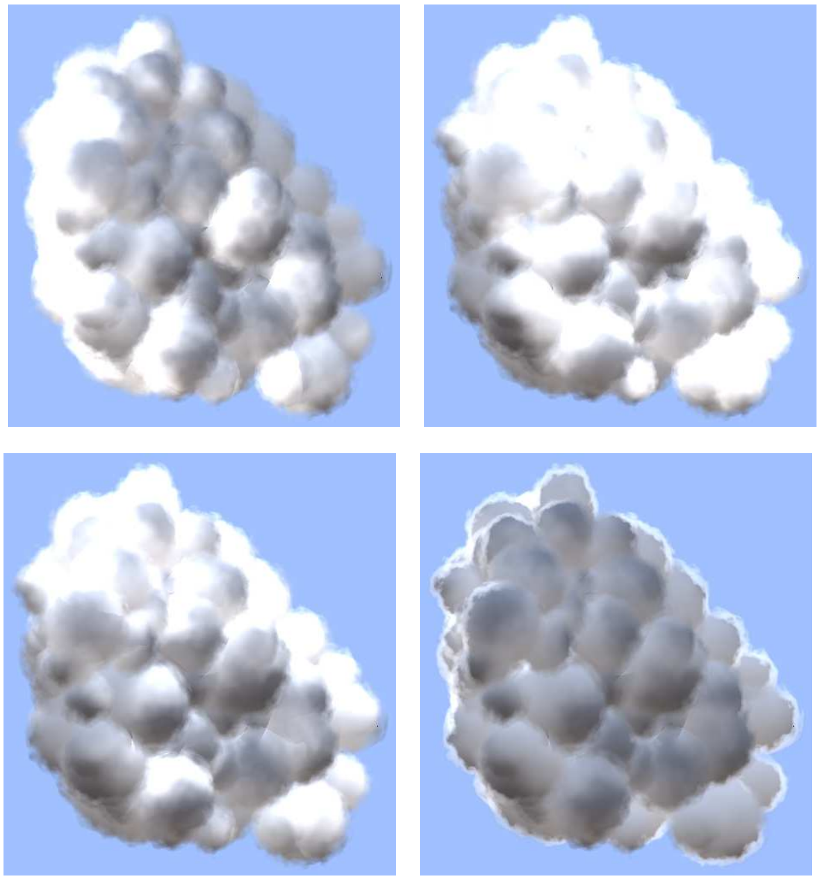

Figure 7: Resulting images for various sun positions. Only one level of surface is implemented. The surface is covered with 87 blended ellipsoids. The rendering time is 100 seconds.

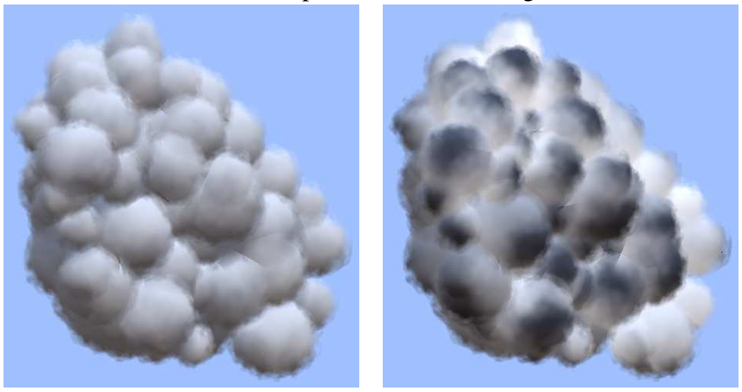

Figure 8: Components of the image 7, 2: *Left:* direct shading contribution. *Right:* inter-reflections contribution.

6 Future Work

Of course, the fact that we haven't paid much attention to the shape model of cumulus makes the quality of the rendering less easy to appreciate. We plan to investigate further on implicit surfaces to model the spherical details, as this allows for smooth shape and eases animation. Concerning animation, we plan to connect this to our multiscale bricks of cloud animation [12].

We absolutely have to implement the radiosity estimation also at a larger scale, in order to be able to model and correctly shade large clouds made of several large heaps of bubbles. This will probably be done for the final version of this paper.

Note that most of our model could be adapted to real-time rendering, as we used ray-tracing only for the visibility determination and the shadow calculation.

Finally, we currently investigate phenomenological and analytical approaches for other natural phenomena, such as the animation of brooks, and the shading of forests [11].

A Scattering at Large Scale

A narrow forward phase function can be figured by a Gaussian in polar coordinates with standard deviation σ^2. This value corresponds to a unit amount of droplet crossed. When a volume twice as large is crossed, each deviated ray exiting the unit section is scattered again in the following section. This exactly corresponds to the squared convolution of a Gaussian. More generally, considering a volume n times as large as the unit volume yields a cumulated scattering function whose standard deviation is $n \times \sigma^2$. As the polar coordinates are cyclical, the diagram finally turns isotropic for a given scale, that is probably about a few dozen of meter. The fact that the image of a paraglider totally disappears once he is a dozen of meters inside a cumulus, that these clouds are highly reflective, and that the core of mature cumulus illuminated from the back is quite homogeneously dark are illustrating this. Oppositely, other kind of clouds as well as young cumulus are very transparent and quite anisotropic, as they are both thin and light.

References

[1] 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.

[2] 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.

[3] David S. Ebert. Volumetric procedural implicit functions: A cloud is born. In Turner Whitted, editor, *SIGGRAPH 97 Technical Sketches Program*. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[4] Geoffrey Y. Gardner. Simulation of natural scenes using textured quadric surfaces. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 11–20, July 1984.

[5] Geoffrey Y. Gardner. Visual simulation of clouds. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 297–303, July 1985.

[6] W. W. Grabowski and T. L. Clark. Cloud-environment interface instability: Rising thermal calculations in two spatial dimensions. *Journal of the Atmospheric Sciences*, 48:527–546, 1991.

[7] W. W. Grabowski and T. L. Clark. Cloud-environment interface instability, part II: Extension to three spatial dimensions. *Journal of the Atmospheric Sciences*, 50:555–573, 1993.

[8] James T. Kajiya and Brian P. Von Herzen. Ray tracing volume densities. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 165–174, July 1984.

[9] N. L. Max. Light diffusion through clouds and haze. *Computer Vision, Graphics and Image Processing*, 33(3):280–292, March 1986.

[10] Nelson L. Max. Efficient light propagation for multiple anisotropic volume scattering. In *Fifth Eurographics Workshop on Rendering*, pages 87–104, Darmstadt, Germany, June 1994.

[11] Alexandre Meyer and Fabrice Neyret. Multiscale shaders for the efficient realistic rendering of pine-trees. In *Proceedings of Graphics Interface 2000*, May 2000.

[12] 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. Eurographics, Springer Wein, September 1997.

[13] Tomoyuki Nishita, Eihachiro Nakamae, and Yoshinori Dobashi. Display of clouds taking into account multiple anisotropic scattering and sky light. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, pages 379–386. ACM SIGGRAPH, Addison Wesley, August 1996.

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

[15] Pierre Poulin and Alain Fournier. A model for anisotropic reflection. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24(4), pages 273–282, August 1990.

[16] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21(4), pages 293–302, July 1987.

[17] Jos Stam and Eugene Fiume. A multiple-scale stochastic modelling primitive. In *Proceedings of Graphics Interface '91*, pages 24–31, June 1991.

[18] Jos Stam and Eugene Fiume. Turbulent wind fields for gaseous phenomena. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 369–376, August 1993.

[19] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, pages 129–136. ACM SIGGRAPH, Addison Wesley, August 1995.

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

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399