A New Approach for Multiple Scattering Modelling in Participating Media

Laurent Da Dalto, Mathias Paulin, Jean-Pierre Jessel, René Caubet
Institut de Recherche en Informatique de Toulouse,
Université Paul Sabatier,
118, Route de Narbonne, 31062 Toulouse Cedex, FRANCE
E-mail : {dadalto,paulin,jessel,caubet}@irit.fr

Abstract :

Many of the current rendering methods for participating media need a large storage space and a huge computations but do not simulate realistic media. Most of the time, it is due to the way the media are sampled. A fine sampling leads to a good representation of the optical and physical properties of media but requests often powerful computers and huge memory space.

That's why we present a new sampling solution. It is based on analytic shapes to model volume boundaries and variation functions to model volume properties. The sample is fitted to the ray while it is crossing the medium. An independent solution is proposed for the multiple scattering effects.

This new approach has numerous advantages. With a low memory and computation cost, the medium parameters are easy to change. The most important advantage is that all the physical and optical properties of volumes are designed, even the multiple scattering. Moreover, the density varies inside the medium using a non-linear function. Thus, we can represent a wide range of participating media without making important change in the implementation.

1. INTRODUCTION

During the last years, rendering methods became more and more close to reality. However the memory and computation cost of these methods strongly grew and required more powerful architectures.

With the possibility to make more computations in less time and the development of big storage units, new methods of rendering appeared, taking physical and optical properties of the scenes into account. In spite of this evolution, equations representing those properties are not easy to solve and their storage costs are very important.

The study of materials and light properties has been developed (in our research group as well) but the application of participating media theory seems to be more difficult.

The number and the complexity of equations required to represent such a medium is very important, so this theory is very complex to apply in a rendering application because. There are two kinds of existing methods. The first kind consists of analytic methods, in which the authors tried to simplify the mathematical model so they could solve those equations. It is necessary to find a method to solve the radiative transfer equation of the medium which is a very complex integro-differential equation. Therefore several restrictions are necessary in

order to make this equation analytically solvable. Blinn [Bli82] was the first who represented a gaseous phenomena (Saturn rings) but he made a lot of simplifications to succeed his simulation (low albedo medium, only simple scattering are taken into account). Max [Max86] also used the model presented by Blinn but added shading volumes to take light sources into account. At last, Nishita [Nis87] used also an analytic method to represent participating media, but with the restriction that the media must have a uniform density. The volumes are represented by curved patches. Analytic methods are not very accurate to represent gaseous phenomena because they imply some restrictions on the medium properties (densities, scattering, isotropic medium,...).
Therefore, other methods which do not modify the media properties have been presented.
First there are statistic methods. One of the most famous of this family is the Monte-Carlo method. Typically, this method allows to simulate radiative transfers into the medium and the scene using the "random walk". Which means that a probability distribution is associated with each property. This method requires high computation time to be close to reality and the errors generated are not easily controlled. Pattanaik [Pat93] and Blasi [Bla93] have developed this method.
Rushmeier [Rus87] uses the zonal method which is close to the radiosity method to compute the interaction between volumes and objects in the scene. Thus, she computes three kinds of form factors: volume-volume, volume-object and object-object (classical form factors). However, the media must be isotropic and this method takes only simple scattering into account. It is possible to apply this method to anisotropic media or media with high albedo but the computation times are very important and it is very hard to implement. Other methods, more accurate than the zonal method, are the flow methods. The aim of these methods is not to solve an integro-differential equation (the radiative transfer equation for the medium) but to transform this equation in an equation system with partial derivatives. In order to this, it is necessary to separate the spatial dependence and the angular dependence of the luminance. Two important flow methods have been presented: the spherical harmonics and the discrete ordinates. The first one is very suitable for isotropic media but needs a lot of computation time and memory for anisotropic media. It has been used by Kajiya [Kaj84]. The second one samples the volume into "voxels" and the angular space into solid angles where the luminance is supposed uniform. Each voxel has a constant density. The method computes the propagation of the light energy through the voxels of the medium for each direction representing a solid angle. With this algorithm, it is possible to represent media with various albedo and to take multiple scattering into account. This method has been used by Patmore [Patm93] and Languénou [Lan94].
Finally, Stam [Sta95] proposed an accurate model to represent density distribution of a medium like smoke or clouds but there are still important assumptions concerning the physical properties of media and particularly the diffusion process.
The aim of our works is to represent participating media as accurately as possible. This paper describes a new method allowing to represent media without any constraint (of size or number) and with all their optical and physical properties.
Our modelling is closed to Ebert A-buffer solution [Ebe90] since we sample each ray in a regular way. Unfortunately Ebert uses only solid textures and made some important assumptions (like simple scattering and low albedo).
First, we will introduce our representation of the participating media. Then we will show how it was included in our global illumination application. Finally, we will try to discuss the interest of such a model and present some results.

2. The medium modelling

To model participating media distribution, we represent the spatial distribution of the density linked to one medium. All the optical and physical properties depend on this distribution. We can use three different kinds of structure to express the density distribution.

The basic one is defined by selecting a point as the centre point of an ellipsoid. For this point, we set a density value and an "extinction function". So, the density value decrease depending on the extinction function from the centre point to the border of the ellipsoid. This is the easiest way to model the density distribution.

An other solution we implement, is to use implicit function. The value given by the function at a selected point represents the punctual value of the density. This solution is good to represent media like clouds but the computation of the density value is more expensive than the basic solution.

Finally, we can use three dimensional functions to define the density distribution. These functions represent solid textures. We can update the parameters (noise for example) to create any kind of medium. This solution is well fitted to the smoke modelling.

Now, if we consider a point in the volume, it can be associated with particles in the medium (or molecules depending on their size and their properties). The light energy entering the point is partly absorbed, partly scattered, partly re-emitted. The point himself can emit light energy in a given direction (see Fig. 1). All these phenomena are density dependant.

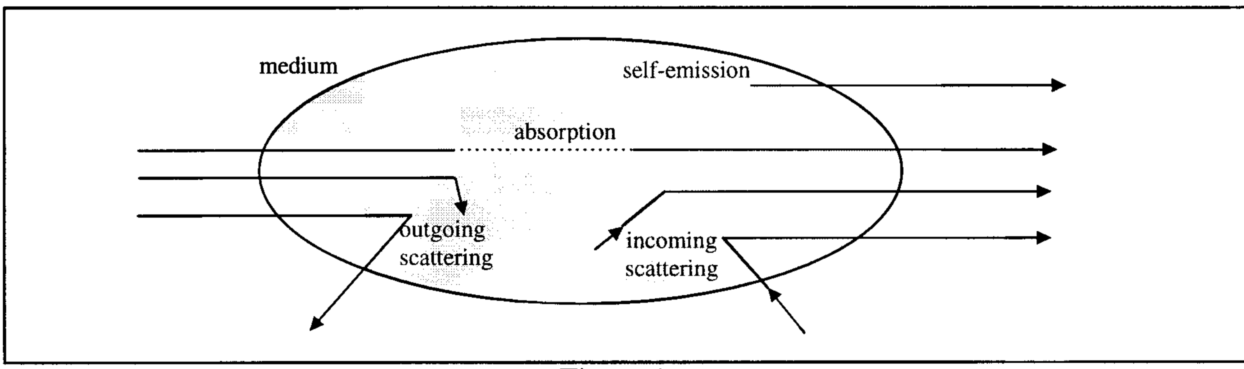


Figure 1 :
Optical phenomena associated to a participating medium

For a small travel S of a ray into the medium at a temperature T, we can represent the luminance variation by the radiative transfer equation :

$$\frac{dL_\lambda}{dS} = \underbrace{-a_\lambda L_\lambda(S)}_{\text{absorption}} + \underbrace{a_\lambda L_{\lambda b}(T)}_{\text{self-emission}} - \underbrace{\sigma_\lambda L_\lambda(S)}_{\text{outgoing scattering}} + \underbrace{\frac{\sigma_\lambda}{4\Pi}\int_{\omega_i=4\Pi} L_\lambda(S,\theta,\phi)\Phi(\theta,\phi)d\omega_i}_{\text{incoming scattering}} \quad (1)$$

where **a** is the absorption coefficient, σ is the diffusion coefficient, **L** is the amount of energy (luminance) incoming from direction **S**, L_b is the black body luminance, **T** the local temperature, Φ the phase function. All the terms are wavelength depend (λ).

The first three terms of the equation are easy to compute as all the coefficients are known. The last term is the most complex to evaluate and needs, at least, to be approximated as we can not make an analytic integration over three dimensional space.

Moreover, the coefficients (a_λ and σ_λ) are very hard to value because they can vary from one element to an other in the same volume. Most of the latest proposed models require that, at least one parameter must be constant or uniform upon the medium. Moreover, a large number of small basic volumes is requested to approach a realistic medium [Lan94]. It is memory expensive and needs more computation time to render.

Therefore we propose a new way of representing parameter variations and a solution to compute the incoming scattering in a realistic fashion with few assumptions.

2.1 Absorption and Self-emission :

In our application, elements in a selected medium have the same size. As we can generate as many media as we need, we can also cross media to create more complex one. So, for one medium, we define the value of the absorption and self-emission for one element of this medium. Depending on the density, we can compute the value of the absorption, emission and outgoing scattering at any location in the medium.

The absorption and the outgoing scattering can be linked to create the loss term represented by

$$K = a + \sigma \qquad (2)$$

The self-emission and the energy loss are the easiest parameters of the volume to compute because they are not incoming directions dependent. The last parameter, the multiple scattering, needs a more complex definition, we will discuss now.

2.2 Multiple scattering :

To model the multiple scattering, we must define the **phase function**. This function $\Phi(\theta,\phi)$ expresses the ratio of energy outgoing in the direction $D(\theta,\phi)$ compared to the energy incoming from a selected direction (see fig. 2).

In fact, (θ,ϕ) represents the angle between D and the incoming direction of the ray so we can rotate the phase function without changing their values. Therefore a set of theoretical functions have been defined [Bla93] to be used in analytic computations. We implement all those presented in Blasi paper.

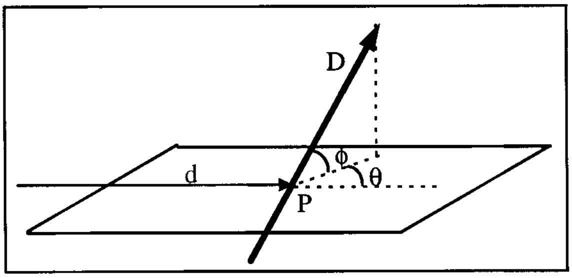


Figure 2 :

D receives energy from d scattered on particle at point P. Angles (θ,ϕ) define the ratio of energy scattered.

In our case, we supposed the phase function is the same for all the particles (or the molecules) inside a volume. Actually, realistic media are composed with a lot of different phase functions. But it is possible to approximate this phenomena by crossing two or more media with different phase functions. In this case, the other parameters (absorption, self-emission and outgoing scattering) would be taken many times into account therefore it is necessary to adapt them.

We are able to compute the energy gain from the incoming scattering using the phase function and the incoming luminance of one element. As we can not integrate the energy reaching the element analytically, we must sample the space into solid angles. Although this sampling solve the problem of incoming scattering integration for a particle, we still need to compute the scattering between all the particles of the medium and between the medium and the other objects of the scene.

We solve this problem by computing the incoming scattering for only one point of the medium (making a little assumption). The value for the other medium elements will be deduced from this point. We will see in detail how this can be implement during the global illumination processing.

3. Illumination process

The model presented before, has been implemented in a progressive radiosity method. We use a voxel grid to speed up the ray shooting. Moreover, the illumination phase generate adaptive subdivision of objects depending on the created shadows [Pau94]. This application can be executed in parallel using an heterogeneous set of machines managed by PVM.
During the illumination process, media are considered like objects in the scene but are illuminated using a different algorithm.
In fact, for one selected source, we first compute the media illumination and then illuminate the solid objects. This sequence is not an arbitrary selection. It is important because the illumination of a medium create a new multiple scattering function linked to the current light source. When we illuminate the other objects, there may be media between light and selected objects. This sequence allows to consider the multiple scattering factor due to the considered light source when the visibility ray crosses participating media before reaching an object.

3.1 Volume illumination

In a classical global illumination application, we compute the form factor between the selected light emitter and all the objects in the scene. We use the progressive radiosity process to compute these illuminations.
So, first of all, a light source is selected as the most emissive patch in the scene. Now, we have to compute the form factor between this patch (represented by a triangular face) and the volumes.
We first compute the distance between the light source and the volume representing the medium. This distance allows us to compute the energy loss due to the air traversal. Then, if the amount of energy is important enough, we will compute the incoming scattering for the first particle the ray hits in the medium. For the moment, we do not consider the objects or other media existing between the light source and the selected medium. This will be take into account during the rendering phase.
Here, we assume that all the particles next to the selected particle have the same size and properties that the central particle. This will help us to solve the incoming scattering for the hit particle. We will compute the incoming scattering using an iterative convergent algorithm. It must converges to the real value of the incoming scattering. As, for the moment, we do not know the outgoing direction, we will store the energy incoming in all the direction sampled before. We now will introduce the way the algorithm converges to the solution.

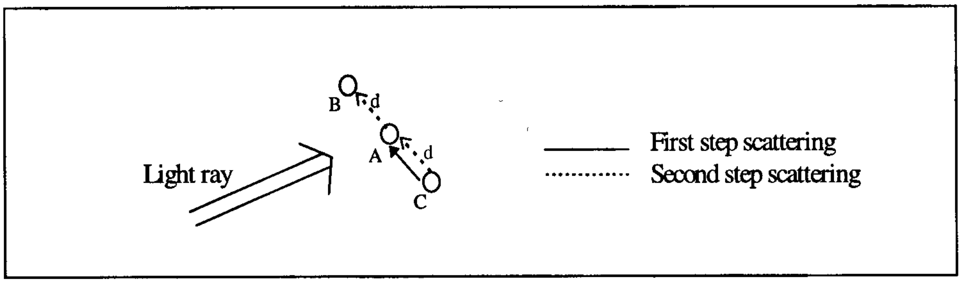


Figure 3 :
The light ray creates no incoming scattering during the first step along the direction d. Whereas, during the second step, A scatters energy along d to B, as C will scatter the same amount of energy to A along d.

At the first step (step 0), only the energy incoming from the light source is stored. It is a basic representation of a simple light scattering solution. If we stop here, the algorithm will be very quick but we will make a big approximation.
The second step (step 1) models the first scattering. The light energy incoming the particle will be scattered in all the sampled directions to the neighbouring particles. Therefore these particles will update their multiple scattering function. But, as the particles are very closed to each other and have the same size, we could translate this variation to the first hit particle as we show on figure 3.
And so on, we compute each step by adding new incoming scattering values to the values stored before.
We stop our computation in two cases : either the sum of all incoming scattering added is below a minimal value (convergence) or we reached the maximal step the user defined.
Finally, the incoming scattering in all the sampled directions is stored and associated to the light source who create it. So, a multiple scattering function is computed and stored for each important light source illuminating the volume (remember that for the moment this function represents the punctual value of the multiple scattering for one point in the medium).

3.2 Solid object illumination

The illumination process concerning the solid objects is not really different from the classical global illumination process.
As before, we compute the form factor between the light source patch and the selected object patch to be illuminated. But now, we consider also the media existing between the two patches. Those volumes may and would update the light energy exchanged between the two patches. The medium may contain the light emitter or the patch to be illuminated or may just be on the way of the ray between the two objects. This step create the shadows of the media upon the solid objects.

4. Rendering process

During this step, rays are shot from the observer into the scene.
Now, we must consider the participating media contribution and render them. So, we compute the intersection between rays and the media in the scene. Classically, we only need to know the first intersection with solid objects and, thus, compute the pixel colour. Now, we still compute the first object intersection but before returning the pixel colour, we need to determine if the ray intersects some media before reaching the object. In this case the colour returned by the ray is modified by media traversal. Moreover, if the ray leaves the scene without hitting anything, it is also very important to determine if it does not cross any media before. Otherwise the scattering effects will update the pixel colour of the scene background.

4.1 Volume traversal

If a ray crosses a medium, we must compute its energy variation due to the volume. For this, we sample the ray into the volume by taking regular spaced point along its path. Only the latest sample is not regular because it depends on the position of the leaving point along the ray path.

We compute the value of each property of the medium for each sampling point (self-emission, absorption, incoming scattering and outgoing scattering). The two first properties can be

considered as punctual values. The energy loss must be computed along the distance between two sampling points.
This is make using Bouguer's law which express the attenuation A between two points P and P' :

$$A(P, P') = e^{-\tau(P, P')} \quad (3)$$

where τ is the optical depth between P and P'. The optical depth can be defined by an absorption coefficient α and a scattering coefficient σ :

$$\tau(P,P') = \int_{P}^{P'} \alpha(P'')dP'' + \int_{P}^{P'} \sigma(P'')dP'' \quad (4)$$

In our case, we have defined K as the loss coefficient (see equation 2) for one point and corresponding to the ratio of energy absorbed and diffused along a unit path. So the equation (4) can be rewritten :

$$\tau(P,P') = \int_{P}^{P'} K(P'')dP'' \quad (5)$$

Therefore, we must compute the most complex property of the component of the media : the incoming scattering. For one medium, we have a set of multiple scattering functions computed for one particle and for one selected light source. We need to interpolate the value of each function for the point selected along the ray path and add each scattering effects.
For each sampling point, we shoot a ray to each light source (stored during the illumination process). If the ray does not hit the source then we do not add its associated incoming scattering function to the scattering at the sampling point. Else, we compute the real value of the multiple scattering on the selected point.

4.2 Computing real multiple scattering :

To do this, we compute the energy loss and the self-emitted energy from the point the ray shot to the light source enter the volume to the selected point. Then, we update the current multiple scattering function to take these modifications into account. In fact, we have computed the multiple scattering function and stored its real value for a point on the border of the volume. So, for this point, the light ray did not travel into the medium whereas, for the sampling point, we need to consider the part of the ray included into the volume. That is why we need to make such a computation.
Now, we know the real value of the incoming scattering function for the sampling point and for a selected light source, we can do the same computation with each light source stored for the medium. Having all the incoming energy in all the sampled directions at the point, we can compute the energy gained by the incoming scattering using the phase function. Therefore, we compute the last part of the Radiative Transfer equation on the selected point:

$$\text{Scattering}_{\text{Incoming}} = \frac{\sigma_\lambda}{4\Pi} \int_{\omega_i = 4\Pi} L_\lambda(S,\theta,\phi)\Phi(\theta,\phi)d\omega_i \quad (6)$$

This computation is not really expensive, as we already compute the incoming scattering effects. Most of the time is spent to shoot rays to light sources. But, it is only visibility rays without any energy computations needed, so, with our acceleration voxels grid the shot time is reduced.

5. Results

The model we just present, has been implemented on graphic workstations (SGI R4400). We have made several computations to see how it reacts. With our method, we are able to stop the multiple scattering process whenever we want to see the visual result. So we computed the

first scattering (simple scattering from light sources) and compared this result to a "double scattering" (simple plus second level scattering) and so on.
Picture 1 shows a simulation of a cloud using three dimensional texture as in Ebert model. We can see the shadow cast by the occluder on the medium. In this case, the medium is not self emissive. Its density varies from a point to an other. On this picture, only simple scattering has been represented. There is only one light source near and above the planes. We used Rayleight phase function (the energy is scattered principally forward and backward).
Picture 2 is the same cloud that the one on picture 1 but now we compute all scattering effects (convergence). The light diffusion is more important and can be seen on visual results. As the observer is near the light source, the medium scattered to him an important amount of energy (backward). So the medium is very bright. Illumination lasted 6 seconds (300 ms for picture 1) and rendering 15 minutes. Rendering time is the same for both pictures as all multiple scattering computations are made during illumination.
This picture is more expensive to compute than the previous one but the difference is not really important. In fact, we compute only once the real value of the multiple scattering (for one point in the medium for each light source). Then during the rendering, the time spent to compute the real incoming energy for one point stays the same. This is a great advantage of this representation.
Picture 3 is a simulation of a planet with a visible atmosphere. The planet is completely included in the volume. The medium emits blue light and has a little absorption coefficient. We can see the planet side which is illuminated by the primary light source (a big sun) as well as its dark side.
Therefore, using the same model, we can simulate little participating media (gaseous phenomena like smoke, fog or clouds) as well as very large media like planet atmosphere.
And finally, picture 4 is a representation of a room with fog.

6. Conclusion and future directions

With this new approach for modelling participating media, we have reduced strongly the memory cost needed to represent media. All optical and physical properties are represented and can vary inside the medium. Our model have a lot of parameters so we can use implicit functions, three dimensional textures, analytic functions or simple values to represent the density distribution. From this density, the other properties of the medium varies.
All shadows are computed, even those created by volumes or objects on a medium. This phenomenon, which is very difficult to consider, is very easy to take into account using our method. Furthermore, it is possible to have objects into the media without modifying the algorithm. Objects may be partly or completely included in a volume. The algorithm is the same in both cases.
Most of all, we are now able to represent the multiple scattering without sampling the space in our media. We compute real multiple scattering for one point and approximate the value of the scattering for the other points. For the moment, we store one scattering function for each source illuminating the medium but it is less expensive than store this same function for each voxel for example (if we compare our method to [Lan94] because it seems there will be more voxels in a medium than sources selected in a scene.
We plan to further develop this method. We will try to compute the errors due to the scattering approximation and to the spherical angles sampling and try to reduce them.
Our aim is to create good modelling of the participating media without creating spatial sampling (generating aliasing and important assumptions).

We have also begun the implementation of realistic materials (with bi-directional reflectance) and realistic lights (with spectral and spatial distribution). This will allow us to create scenes based on realistic physical and optical properties.
Finally, we started to develop a new application generating Infrared pictures and integrating participating media. This application will be of the greatest importance for several domains. Those works is developed in association with CESR (a remote sensing laboratory) who gives us theoretical tools and realistic data. The aim of the association is to represent radiative transfers into a forest. Then we will be able to study the way the forest grows depending on the light energy it receives.

7. Literature

[Bla93] - P. Blasi, B. Le Saëc, C. Schlick, "A Rendering Algorithm for Discrete Volume Density Objects". Proceeding of Eurographics 93, Vol. 12, N° 3, Barcelona, September 1993, pp. 201-210.

[Bli82] - J. Blinn, " Light Reflection Functions for Simulation of Clouds and Dusty Surfaces ". Proceeding of SIGGRAPH'82 (Boston, Massachusetts, July 26-30, 1982). In Computer Graphics 16,3 (July 1982), pp. 21-29.

[Ebe90] - D.S. Ebert, R.E. Parent, "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques ". Proceeding of SIGGRAPH'90 (Dallas, Texas, August 6-10, 1990). In Computer Graphics 24,4 (August 1990), pp. 357- 363

[Kaj84] - J.T. Kajiya, B.P. Von Herzen, " Ray Tracing Volume Densities ". Proceeding of SIGGRAPH'84 (Minneapolis, Minnesota, July 23-27, 1984). In Computer Graphics 18,3 (July 1984), pp. 165-174.

[Lan94] - E. Languénou, K. Bouatouch, M. Chelle, " Global Illumination in Presence of Participating Media with General Properties ". 5th Eurographics Workshop on Rendering, Darmstadt, June 13-15, 1994, pp. 69-85.

[Max86] - N. Max, " Light Diffusion Through Clouds and Haze ". Computer Vision, Graphics and Image Processing 33 (1986), pp. 280-292.

[Nis87] - T. Nishita, Y. Miyawaki, E. Nakamae, " A shading model for atmospheric scattering considering luminous intensity distribution of light sources ". Proceeding of SIGGRAPH'87 (Anaheim, California, July 27-31,1987). In Computer Graphics 21,4 (July 1987), pp. 303-310.

[Pat93] - S.N. Pattanaik, "Computational Methods for Global Illumination and Visualisation of Complex 3D Environments". PhD thesis, Birla Institute of Technology and Science, INDIA, February 1993.

[Patm93] - C. Patmore, " Simulated multiple scattering for cloud rendering ". ICCG 93 Conference Proceeding, IFIP, February 1993, pp. 59-71.

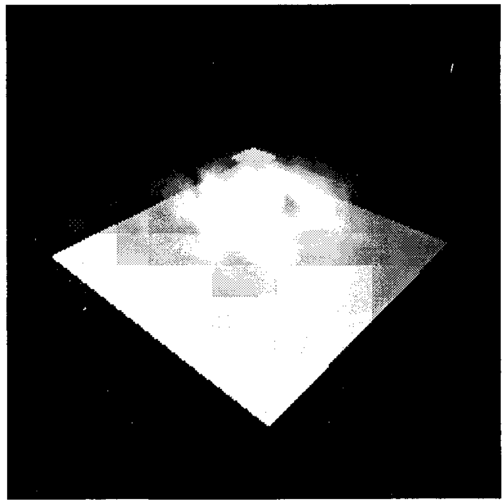
[Pau94] - M. Paulin, J.P. Jessel, " Adaptative Mesh Generation for Progressive Radiosity : a Ray-Tracing based Algorithm ". Proceeding of Eurographics 94, Oslo, September 1994, pp. 421-432.

[Sta95] - J. Stam, E. Fiume, " Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes ". Proceeding of SIGGRAPH'95 (Los Angeles, California, 6-11 Août, 1995), Computer Graphics, pp. 129-136.

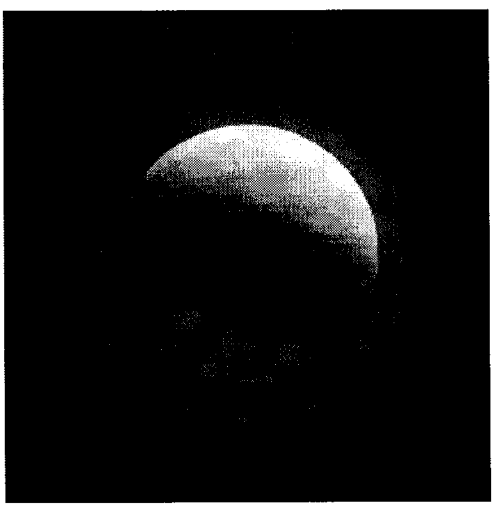
[Rus87] - H. Rushmeier, K. Torrance, " The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium ". Proceeding of SIGGRAPH'87 (Anaheim, California, July 27-31,1987). In Computer Graphics 21,4 (July 1987), pp. 293-302.

8. Pictures

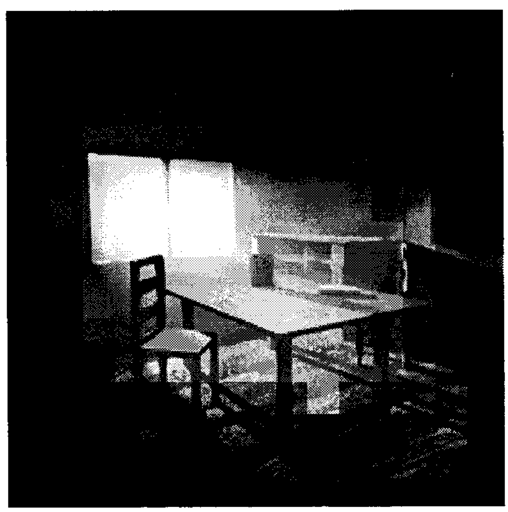
Picture 1



Picture 2



Picture 3



Picture 4