Real-time realistic illumination and shading of stratiform clouds

Antoine Bouthors   Fabrice Neyret   Sylvain Lefebvre

EVASION-GRAVIR/IMAG-INRIA

Abstract
Realistic rendering of clouds involves solving the complex interaction of light within the cloud and with its environment. Interactive methods achieve efficient cloud rendering by ignoring several lighting effects. However, these effects are visually important, and removing them strongly reduces realism.
We present a novel approach for capturing the important effects of multiple anisotropic Mie scattering within cloud layers (i.e., stratiform clouds), and the inter-reflections between the ground and the cloud base under sun and sky illumination. Our model maps well to graphics hardware, enabling the real-time rendering of animated cloud skies over landscapes.

1. Introduction

The visual aspect of a cloud depends on complex light interactions: individual cloud droplets scatter light according to a very irregular diagram, the Mie function [BH98], which is responsible for glories and fogbows around the antisolar direction. Inside the cloud, light is anisotropically scattered multiple times before leaving. Despite this accumulated diffusion, anisotropy remains strong at many places such as along the silhouette (thus the "silver lining") or in the direction of the "pseudo-specular" reflection. The ground-clouds interaction introduces further lighting effects. Since a cloud's albedo is high, the cloud's bottom reflects the ground. Inuits can find a channel in ice shelf or find the land from the sea by observing "water sky" and "ice blink" on the cloud bottom [Art].

The complete simulation of a cloud appearance including all these effects has never been done. Several interactive cloud simulations have been proposed, but they increase performance by ignoring most light interaction modes. Our model revisits the cloud illumination problem and accounts for new important lighting effects (see Fig. 2, right). It enables the real-time rendering of animated cloudy layers over a landscape. In this paper, we focus on stratiform clouds such as stratocumulus, i.e., not vertically developed like cumulus or cumulonimbus (see Fig. 1). Also, we do not consider clouds made of ice. Our goal is to render such clouds in real time from a viewpoint located either under the clouds (*e.g.,* from the ground) or above them (*e.g.,* from a plane). We are not considering flying trough or inside the clouds.

Our contributions are:
- An improved representation of the scattering model, based on the exact Mie function (whereas previous approaches use coarse approximations) and droplet size distributions.
- Reflectance and transmittance shaders to be applied to the cloud surface. These shaders account for local scattering and estimate the effects of multiple light scattering through the cloud.
- A GPU-friendly algorithm to efficiently compute inter-reflections between the clouds base and the ground (accounting for shadows, sky, and landscape texture).

Our approach consists in obtaining functions giving the intensity and direction distribution of light on each side of a homogeneous plane-parallel slab. To do so, we separately treat the different orders of scattering within this slab (see Fig. 2, left). We use analytical solutions for orders 0 and 1, and approximate solutions for orders 2 and ≥ 3. We propose a modified Mie model allowing us to treat only meaningful scattering events, *i.e.,* events that cause a significant change a the photon's path direction. We apply the resulting functions as a shader applied on the cloud surface, locally considered as a slab.

We account for the inter-reflection between the cloud base and the ground (considered as parallel planes) by introducing an efficient GPU-compliant algorithm performing radiosity iterations: density and color variations on the two planes are stored in textures, as well as their dynamic light-
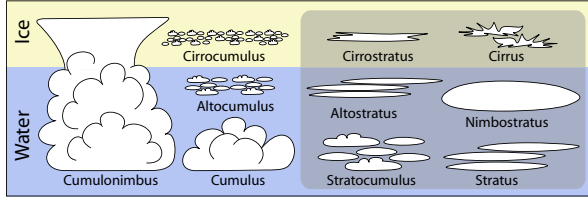
Figure 1: *Main types of clouds. Gray area: stratiform clouds. Since these clouds have a layer shape with slightly varying thickness, we locally approximate them by a slab.*

Figure 2: Left: *All light path categories we simulate:* ≥ 3 *orders of scattering* (I^r_{3+}, I^t_{3+})*, double scattering* (I^r_2, I^t_2)*, single scattering* (I^r_1, I^t_1)*, transparency* (I^t_0) *ground-clouds radiosity* (*Rad*)*.* Right: *All effects we reproduce:* a: *diffuse reflectance.* b: *glory.* c: *fogbow.* d: *pseudo-specular reflectance.* e: *diffuse transmittance.* f: *ground-clouds inter-reflection.* g: *forward scattering.*

ing. Form factor integration is approximated using weighted rings in the MIP-map pyramid (see section 5).

We validate our model by comparing it to Monte-Carlo simulations. Our GPU-enhanced implementation runs at 18 to 40 FPS on animated 512×512 cloud height fields.

2. Previous Work

An accurate way of simulating the multiple anisotropic Mie scattering of light in a cloud is to rely photon tracing, i.e., Monte Carlo integration. However, this would require an unacceptable time to converge. Various simplifications have been proposed in the literature:

- Low albedo / low density hypothesis: one single scattering is considered [Kv84, Bli82].
- Simpler scattering functions: Rayleigh [HL01, Kv84], Gaussian [PAT*04], Henyey-Greenstein [Max94, NND96], which are easier and faster to compute and whose smooth variations present directional artifacts.
- Simplification into simple diffusion solvers: isotropic scattering [DKY*00], diffusion approximation [Sta95, JMLH01].
- Forward-dominant hypothesis: single-pass algorithms [HL01], very convenient for GPU-enhanced interactive rendering.

Only a few papers actually make use of Mie scattering to render atmospheric effects [REK*04, JW97]. Only [REK*04] use it for clouds, and only for local effects such as the glory (the light transport is assumed to be forward-dominant).

The problem is that visual features of clouds (see Fig. 5) are due to different lighting modes directly or indirectly connected to the particular shape of the Mie scattering diagram (see Section 3.3): Glories and fogbows are essentially due to single scattering and strongly depend on the backward peaks of the phase function. "Pseudo-specular" reflection and bright silhouettes are caused by the forward anisotropy and are the effect of a few scattering events (2 to 10). The overall brightness of clouds comes from diffuse illumination (high orders of scattering cancels directionalities in the diagram). Thus, considering only diffusion, or only forward scattering, or smooth scattering functions, forbids realism. And simply adding a Mie single scattering on top of a forward [REK*04] or isotropic diffusion is not sufficient either.
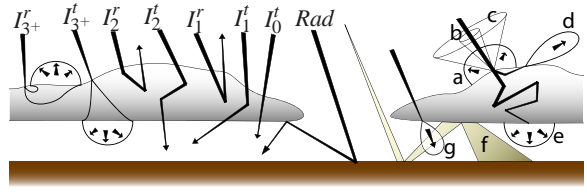
The study of most probable paths [PAT*04] is an interesting way to focus on important features. This accounts for the low-order scatterings, but it misses the diffusion effect. Moreover, [PAT*04] relies on Gaussian phase function.

Cloud optics has been widely studied in other fields (meteorology, applied optics, atmospheric sciences etc.). In particular, the study of cloud droplet phase functions [Dei64] coupled with cloud droplet size distribution [Cla74, MVC00] are of interest to us.

3. Cloud optics

3.1. Multiple scattering in clouds

Multiple scattering has been extensively described in the literature (see for instance [JC98]). Let us recall the main principles of interest to us (the following notation will be used throughout the paper).

The material of a homogeneous cloud can be described by the *effective radius* of its droplets r_e and the *density of droplets* per cube meter N_0. These define the *extinction cross-section* $\sigma = \pi r_e^2$, the *extinction coefficient* $\kappa = N_0\sigma$, and the *extinction function* (*i.e.,* the cloud transparency) $\tau(x) = e^{-\kappa x}$ which is the probability to traverse the cloud along a path of length x without hitting a droplet. Thus, the probability of hit within x meters is $1 - \tau(x)$, and its derivative $s(x) = \kappa e^{-\kappa x}$ is the probability density of a hit exactly at the end of a free path of length x. This also yields the *mean free path* $l_0 = 1/\kappa$.

When hitting a particle, light can either be absorbed or scattered. The albedo of cloud droplets is very close to 1 for visible light, *i.e.,* light is not absorbed in clouds: all the light that hits is distributed elsewhere. The probability that light hitting a droplet from direction ω is scattered in direction ω' is determined by the *phase function* $P(\omega, \omega')$. Since water droplets are spherical, this function is axisymmetric and can be expressed in spherical coordinates $P(\theta)$ with $\theta = \widehat{\omega, \omega'}$.

Stratiform clouds usually hold between $N_0 = 10^8$ and 10^9 droplets per m^3 having a radius $r_e = 2\,\mu\text{m}$ to $15\,\mu\text{m}$. This yields a mean free path $1/\kappa$ of several meters (20 m for $N_0 = 3.10^8\,\text{m}^{-3}$ and $r_e = 7\,\mu\text{m}$). Since a cloud spans hundreds to thousands of meters in each direction, most rays are

scattered several times before exiting the volume (for the example above, the no-hit probability is 10^{-2} for 100 m and 10^{-10} for 500 m).

3.2. Droplet size distributions in clouds

The size of water droplets has a huge visual impact: for a given amount of water, the smaller the droplets the more opaque the cloud (half the size doubles the extinction coefficient). The Mie function also varies with respect to the droplets size. Low- and mid-level clouds are a mixture of water droplets of different sizes. Literature in applied optics and atmospheric sciences rely on *droplet size distribution* (DSD) models. The modified Gamma distribution [Lev58, Dei69, Cla74] is well suited for cloud droplets. It is defined by:
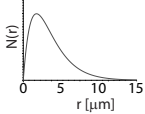
$$N(r) = \frac{N_0}{\Gamma(\gamma) r_n} \left(\frac{r}{r_n} \right)^{\gamma-1} e^{-\frac{r}{r_n}}$$

Figure 3: *Example DSD.*

This function describes the density $N(r)$ of droplets of radius r, with r_n the characteristic radius of the distribution, γ representing its broadness, and N_0 the total density of droplets. Γ is the Gamma function. A cloud distribution is thus only described by the three parameters N_0, r_n and γ. In terms of optical properties, the effective radius corresponding to this distribution is $r_e = (\gamma + 2) r_n$.

Measured parameters for all kinds of clouds can be found in the literature. For stratiform clouds, we refer to [MVC00]. In the examples of this paper, we used values of $N_0 = 300 \text{cm}^{-3}$, $\gamma = 2$ and $r_e = 7 \mu m$. The resulting DSD is shown on figure 3. We consider this distribution to be constant in space and time.
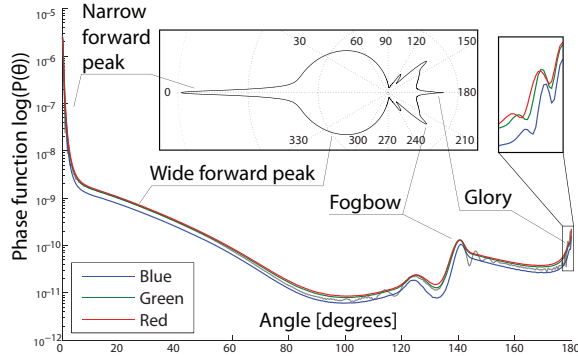

Figure 4: RGB: *efficient Mie phase function (log).* Gray: *pure Mie* ($r = 20 \mu m$). *Note the undesired high frequency ripples.* Inset: *Polar plot of the function (log).*

3.3. Scattering behavior

The phase function $P_{r,\lambda}(\theta)$ characterizing the scattering caused by a water droplet of size r at wavelength λ is obtained by the Mie theory [BH98]. It is quite expensive to

Figure 5: *Some features of stratiform clouds (real photographs).* Left: *glory and fogbow.* Middle: *pseudo-specular effect.* Right: *Strong forward scattering.*

compute, highly dependent on r and λ, and its intensity oscillates strongly when θ varies. This is what makes the Mie function so impractical, and probably why it is avoided in CG cloud simulations.

The *effective phase function* $P(\theta)$ corresponding to a DSD can be obtained by weighting and summing the phase functions of the various droplet sizes [Dei64]. We used Philip Laven's MiePlot software [Lav] to pre-compute the effective phase function for our droplet distributions (see Section 3.2).

Besides the more accurate physics—which shows especially on the glory [MSPS04]— considering the DSD has a very fortunate consequence: the high frequency ripples in $P_{r,\lambda}(\theta)$ vanish in $P(\theta)$ (and, indeed, they are not observed in real clouds). So, the effective phase function is a lot more practical than the pure one.

Each feature of this function directly corresponds to an important visual phenomenon [Min54, LL95, Cow] in the cloud:
• *Narrow forward*: there is a strong peak in the forward direction. It corresponds to 5° concentrating 51% of the function. It is responsible for the *silver lining* on the silhouette when the sun is behind the cloud, since a few scattering events occur for these light paths (see fig. 5, right).
• *Wide forward*: This large lobe represents 48% of the scattering. This part produces a fairly isotropic distribution when several scattering events are accumulated. It also causes the "*pseudo-specular*" reflection in the antisolar azimuthal direction on the top of the cloud layer, since less scattering events are required for light paths in this direction to get out of the layer (see fig. 5, middle).
• 39° *backward peak*: This peak is responsible for the *fogbow* (or *cloudbow*), a bow where the contrast of the cloud layer is enhanced. It is visually similar to the rainbow with poor or no colors. This effect is caused by single scattering event occurring close to the cloud surface and represents very little energy. It is more visible on mist and thin layers (see fig. 5, left).
• *Backward peak*: This 2°-wide peak is responsible for the *glory*, a colored circle around the antisolar point which is often seen from plane, and the *antisolar spot*, which is generally masked by the shadow of the observer. These features are caused by single scattering events occurring close to the cloud surface (see fig. 5, left).

4. Our model

As stated in the overview, we consider different categories of light paths. First, we separate the strong forward scattering *events F* from other scatterings *events S*. We then approximate light paths $\sum_i (F^*S)^i F^*$ as $F^* + \sum_i (S')^i$, where S' corresponds to a modified Mie model: the forward peak is embedded into the extinction function (see Section 4.1).

Furthermore we study the contributions of $S'^1, S'^2, S'^{\geq 3}$ paths for a given outgoing direction of reflection or transmission. The study of these modes is done on plane-parallel layers of thickness H with a homogeneous droplet distribution. Clouds are assumed to behave locally like slabs of such thickness. We are looking for the intensity I outgoing in direction $\overrightarrow{V}$ when a slab is lit from the sun direction $\overrightarrow{L}$.

The 0-scatter corresponds to the transparency and strong forward scattering (Section 4.2). The single scattering contribution can be obtained analytically (Section 4.3). The double scattering is integrated thanks to an approximation (Section 4.4). Note that integrating the energy of these 3 modes with our modified Mie model is equivalent on average to integrating paths with up to 5 scatters with the standard Mie model. This is due to the fact that the strong forward scattering event are embedded in the extinction function. We treat the remainder (the 3^+ mode) as diffuse, and we obtain its importance via a simple 1D model (Section 4.5). The total reflectance and transmittance are used to obtain the amount of reflected/transmitted environment light (coming from the sky and from the ground). The computation of inter-reflections between the cloud base and the ground is treated in Section 5.

Notation: Let $\overrightarrow{V}, \overrightarrow{L}$ and $\overrightarrow{N}$ be the view direction, the light direction, and the normal to the slab (*i.e.,* the up or down direction) at the considered illuminated location. $\theta_v, \theta_l, \theta_{vl}$ are the angles between $\overrightarrow{V}$ and $\overrightarrow{N}$, $\overrightarrow{L}$ and $\overrightarrow{N}$, and $\overrightarrow{V}$ and $\overrightarrow{L}$. $\mu_v = \cos\theta_v$, $\mu_l = \cos\theta_l$. $H_l = H/\mu_l$ and $H_v = H/\mu_v$ are the thickness of the slab along the light and view direction, respectively. The subscripts 0, 1, 2, 3^+ and *ms* refer to light paths undergoing no scattering, single scattering, double scattering, three scattering events or more, or multiple scattering (*i.e.,* $0 + 1 + 2 + 3^+$), respectively. The superscripts r and t indicate that we are computing the light reflected by the slab or transmitted through the slab. Typically, we will consider I^r when the viewer and the light are on the same side of the layer, and I^t otherwise (see Figure 2).

In the following sections, we estimate the contributions $I_0^t, I_1^t, I_1^r, I_2^t, I_2^r, I_{3^+}^t, I_{3^+}^r$ leading to $I^t = I_0^t + I_1^t + I_2^t + I_{3^+}^t$ and $I^r = I_1^r + I_2^r + I_{3^+}^r$. If $\overrightarrow{V} \cdot \overrightarrow{L} \geq 0$ then $I = I_r$ otherwise $I = I_t$.

4.1. Modified-Mie: forward scattering embedding

In order to treat F events separately from S events, we define a modified Mie model embedding the forward scattering. Let $P_f = \frac{1}{4\pi}\int_0^{\theta_f} 2\pi\sin(\theta)P(\theta)d\theta$ be the weight of the
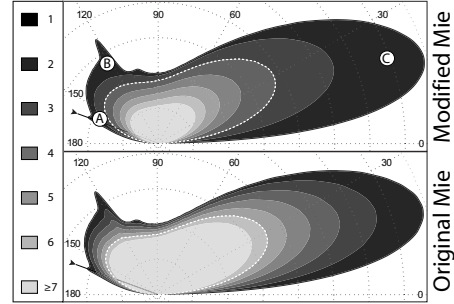

Figure 6: *Reflectance BRDF of a slab with our modified-Mie model* (Top) *and with the original one* (Bottom)*, both integrated through Monte Carlo simulation.* $H = 100\,\text{m}, \theta_l = 70°$. *Colors represent the contribution of each order of scattering. Results with our model closely match the original. Note how 1+2 scattering in the modified-Mie model accounts for up to the fifth order in the original Mie (dashed lines).* A: *glory.* B: *fogbow.* C: *pseudo-specular reflection.*

forward peak for $\theta < \theta_f$. For typical cloud values, $P_f = 51\%$ for $\theta_f = 5°$. In paths containing at least one S event we neglect the direction change for F events, *i.e.,* scattering events with $\theta < \theta_f$. This is equivalent to decreasing the extinction cross section according to P_f and canceling the peak of the phase function. The phase function and extinction parameters presented in 3.1 are thus modified as follows:

$P_S(\theta) = \frac{P(\theta)}{(1-P_f)}$ if $\theta < \theta_f$ otherwise 0

$\sigma_S = (1-P_f)\pi r_e^2$, $\kappa_S = \sigma_S N_0$, $\tau_S(x) = e^{-\kappa_S x}$, $s_S(x) = \kappa_S e^{-\kappa_S x}$

For $P_f = 51\%$, half of the events are implicitly accounted for in the modified extinction, so we only have to treat explicitly the other half. This doubles the mean free path and halves the number of scattering events per light ray.

The consequence of this approximation is a $5°$ error at each extinction event. But since multiple scattering tends to smooth out directionalities, this error is negligible for high orders of scattering. Figure 6 shows a comparison between a Monte Carlo simulation done with the original and the modified Mie scheme. Note that we do not apply this approximation for paths containing no S event, since the deviation would not be negligible. F^* paths (called "*0-scatter*") are instead incorporated into the transparency, see Section 4.2.

In the following section we use our modified-Mie model in place of the original one. That is, notation $P(\theta)$, σ, κ, $\tau(x), s(x)$ represents $P_S, \sigma_S, \kappa_S, \tau_S, s_S$. P_F stands for the strong forward peak function, with $P_F(\theta) = \frac{P(\theta)}{P_f}$ if $\theta < \theta_f$ otherwise 0.

4.2. Transparency and 0-scattering

Transparency is obtained using the extinction function. Using the original $\tau(x)$ function provides the direct transmittance

$$I_{0d}^t = \tau(H_v)\delta_{(\overrightarrow{V} - \overrightarrow{L})}$$

with $\delta_{(x)}$ the Kronecker function. Using $\tau_S(x)$ adds the effect of strong forward scattering. We prefer to provide this
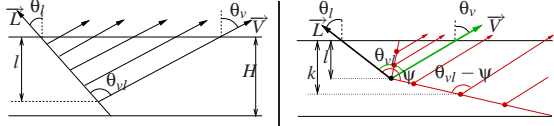
Figure 7: Left: *Single scattering.* Right: *Double scattering. Integral over all red paths is approximated by the value of the green path and a convolution.*

term separately, taking into account the $5°$ deviation of the forward peak:

$$I_{0F}^t = \tau_S(H_v)P_F(\theta_{vl})^{\kappa H_v}$$

Note that $P_F(\theta_{vl})^{\kappa H_v}$ is the strong forward peak convolved by the mean number of scattering, which equals κH_v. We have $I_0^t = I_{0d}^t + I_{0F}^t$.

4.3. Single scattering

The amount of light getting out of the slab in direction $\overrightarrow{V}$ after one scatter can be integrated analytically along the ray as in [Bli82] (see Fig. 7, left):

$$I_1^r = \int_0^H s(\frac{l}{\mu_l})P(\theta_{vl})\tau(\frac{l}{\mu_v})\frac{dl}{\mu_v} = \frac{\kappa P(\theta_{vl})\mu_l}{\mu_v+\mu_l}\left(1-\tau(H_l+H_v)\right)$$

$$I_1^t = \int_0^H s(\frac{l}{\mu_l})P(\theta_{vl})\tau(\frac{H-l}{\mu_v})\frac{dl}{\mu_v} = \frac{\kappa P(\theta_{vl})\mu_l}{\mu_v-\mu_l}\left(\tau(H_v)-\tau(H_l)\right)$$

4.4. Double scattering

In double scattering, photons are first scattered with an angle ψ, then follow a path $L_{1,2}$ before being re-scattered with an angle $\theta_{vl}-\psi$ toward the viewer along a path $L_{2,exit}$: (see fig. 7, right)

$$I_2^r = \int_{l=0}^H s(\frac{l}{\mu_l})\int_\psi\int_k P(\psi)s(L_{1,2})P(\theta_{vl}-\psi)\tau(L_{2,exit})dk\,d\psi\,\frac{dl}{\mu_v}$$

In order to approximate the two last integrals, we use a common approximation [REK*04]: the weight of all the paths scattered by ψ is considered to be constant over ψ. The double integral becomes $\tau(L_{1,exit})\int_\psi P(\psi)P(\theta_{vl}-\psi)d\psi$, which yields simply the convolution $P*P$ denoted $P^2(\theta_{vl})$. Therefore, we can rewrite I_2 as

$$I_2^r = \frac{\kappa^2 P^2(\theta_{vl})\mu_l}{\mu_v+\mu_l}\left(1-\tau(H_l+H_v)\right)$$

$$I_2^t = \frac{\kappa^2 P^2(\theta_{vl})\mu_l}{\mu_v-\mu_l}\left(\tau(H_v)-\tau(H_l)\right)$$

4.5. 3^+ scattering

Since the cumulative phase function tends to become diffuse for high orders of scattering, we assume a diffuse function for I_{3^+}. Thus, we only have to determine the amount of total reflected and transmitted energy. For this we consider a simple 1D model through the slab which gives us the estimated transmittance and reflectance T_i and R_i for each order $i = 0, 1, 2$ and in total ($i = ms$). The details of our 1D model are described in appendix A and validated on figure 8. This model gives us the expressions for $T_{ms}, T_0, T_1, T_2, R_{ms}, R_1$ and R_2 using the same input as for I_1 and I_2. We can thus
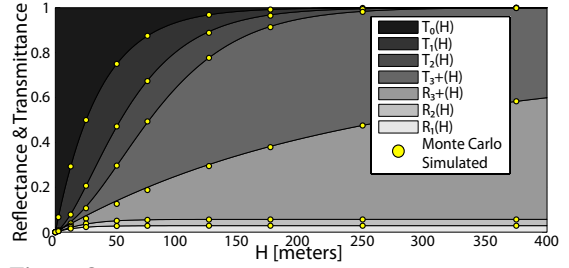
Figure 8: Dots: *values obtained with a Monte Carlo simulation for various H values and* $\theta_l = 30°$. *Areas:* T_i *and* R_i *functions for modes* $i = 0, 1, 2, 3^+$ *(see appendix A) fitted against the simulations. The fitting is almost perfect.*

compute $T_{3^+} = T_{ms} - T_0 - T_1 - T_2$ and $R_{3^+} = R_{ms} - R_1 - R_2$. Then, since I_{3^+} is considered diffuse, the solution for it is simply

$$I_{3^+}^r = R_{3^+}\frac{\mu_l}{4\pi\mu_v}, \qquad I_{3^+}^t = T_{3^+}\frac{\mu_l}{4\pi\mu_v}$$

4.6. Sky and ground contribution

Sky and ground are secondary sources which contributes to the illumination of the cloud. We take then into account in several ways:

• Due to transparency, an observer can see the sky or the ground through the cloud. Rendering is done with standard blending. The alpha value of the cloud is given by $1-\tau(H_v)$.
• The illumination of these wide secondary sources is multiply scattered inside the cloud layer. Due to the size of the sources we consider this transport as diffuse, using transmittance T_{ms} and reflectance R_{ms}.
• Inter-reflections occur between the ground and the cloud bottom, as detailed in the next section. We solve these interactions in section 5 and use the result to obtain the ground illumination. For the sky illumination we use a standard sky model [O'N05].

5. Radiosity simulation

To account for the effects of ground-clouds interaction, we propose a efficient GPU-compliant radiosity algorithm between the ground and the clouds plane. The following sections describe our approach.

5.1. Light and reflectance data

We consider that the clouds bottom and the ground are two parallel planes at distance h. The input parameters are:

• Sun and sky colors (*i.e.,* R,G,B intensities), and sun location. Sky color might be defined by a sky shader (view and sun dependent).
• The ground texture, and a coefficient I_G estimating its original illumination in order to interpret the texture as a color reflectance field $\rho_G(u,v)$.
• The texture of cloud bottom diffuse reflectance $\rho_C(u,v)$. It is obtained from the cloud height field using the diffuse reflectance shader R_{ms} described in the previous section.

- The texture of clouds bottom directional transmittance (*i.e.,* transparency) $T_C(u,v)$. It is obtained from the cloud height field using the function T_{0+1+2} defined in appendix A (1+2 orders are considered mainly directional due to their strong forward scattering).

- The diffuse source $S_C(u,v)$ at the clouds bottom. It is obtained from the cloud height field using the diffuse transmittance shader T_{3+} (described in appendix A) multiplied by the sun color, plus the overall transmittance T_{ms} multiplied by the sky color.

- The diffuse source $S_G(u,v)$ on the ground, which is the ground illumination due to the sun and the sky (before modulating by the ground reflectance).
    - The sun contribution $S_G^d(u,v)$ corresponds to direct shadows obtained by multiplying the sun color by $T_C(u,v)$ (offset by the $h.sin(\theta_l)$ parallax) and multiplied by the Lambert reflectance $cos(\theta_l)$.
    - The sky contribution $S_G^{sky}(u,v)$ requires integrating the incoming light on the hemisphere at every point on the ground. This is explained in Section 5.3.

As an initialization, our algorithm starts by computing:
- The direct lighting on the ground (the $S_G^d(u,v)$ above).
- The indirect lighting on the ground $S_G^i(u,v)$, which requires the integration of the sky (the $S_G^{sky}(u,v)$ above) and the integration of the clouds bottom diffuse source $S_C(u,v)$.

Once the initialization is completed, our inter-reflection algorithm iterates multiple times to simulate light exchanges between the ground and the clouds. We now detail this algorithm.

5.2. Inter-reflections iterative algorithm

To compute the inter-reflections, we associate two 2D texture buffers to both the ground and the clouds plane: R_C and R_G, which accumulate received light ; and dR_C, dR_G which contains the light bouncing from every point after an iteration (light to be emitted at the next iteration). These buffers are low resolution as inter-reflections are low frequency. At initialization, $S_G^i(u,v)$ is directly stored into R_G, and dR_G is set to the initial illuminated ground color $\rho_G(u,v)(S_G^d(u,v) + S_G^i(u,v))$. We do not accumulate the direct illumination (*i.e.,* the shadow map) $S_G^d(u,v)$ into R_G, since cloud shadows from direct sun illumination are at high resolution: at rendering time, the ground texture $\rho_G(u,v)$ will be multiplied by $(S_G^d(u,v) + R_G)$.

At each pass (alternately from ground to clouds and clouds to ground) and at each point (u,v) of the receiver, we have to integrate over the hemisphere the light dR emitted by the other plane. The resulting amount E is added to $R(u,v)$, and $E\rho(u,v)$ is stored into $dR(u,v)$. It will be used as the emitted light at the next iteration. The integration is explained in Section 5.3.

These computations are done on the GPU using a pixel shader. R and dR are updated simultaneously using multiple render targets. MIP-mapped pyramids are automatically generated by the GPU as well.

Note that all these textures are low-resolution, and only a few iterations are necessary: at the ith iteration the amount of energy exchanged is roughly ρ^i, where ρ is the average reflectance of the ground (assuming the cloud albedo is 1). Typical country grounds have reflectance ranging from 0.1 to 0.2, so 2 iterations are often enough. Even with a snow landscape requiring many inter-reflection passes, the radiosity computation remains fast enough for interactive cloud animation (we used 8 for Figure 11.E).

5.3. Integration of light over the hemisphere

For each point $p = (u_0, v_0)$ of the receiver (*i.e.,* each pixel of the buffer), we have to sum the emitted radiosity of all the pixels (u,v) of the other buffer dE weighted by the pixel form factor $F(u_0, v_0, u, v)$. Of course, this 4-fold loop is too costly. Fortunately, integrating textures is a common problem for which numerous solutions are available, such as MIP-mapping.

In the spirit of [SP89], we consider areas of similar contribution on which we assume the weight is constant. We improve their method by choosing 3 square rings S_1, S_2, S_3 of exterior size $h, 2h, 8h$ centered on p. The form factors for these areas are $F(p,S_1) = 0.5, F(p,S_2) = 0.3, F(p,S_3) = 0.18$. In fact we will use weights $f_1 = 0.5, f_2 = 0.3, f_3 = 0.2$ to avoid loosing energy. Let A_1, A_2, A_3 be squares of size $h, 2h, 8h$. We have $S_1 = A_1, S_2 = A_2 - A_1, S_3 = A_3 - A_2$. The average $dE(A_1), dE(A_2), dE(A_3)$ of the texture dE over the squares is obtained in one texture fetch $dE_i(u0, v0)$ using the appropriate level i in the MIP-map pyramid. Note that using buffers with pixels of size h is sufficient. Then, the levels $0, 1$ and 3 will be used. The total integral is thus $E = f_1 dE(A_1) + f_2(\frac{4}{3}dE(A_2) - \frac{1}{3}dE(A_1)) + f_3(\frac{64}{60}dE(A_3) - \frac{4}{60}dE(A_2))$,
i.e., $E = \nu_1 dE_0(u0, v0) + \nu_2 dE_1(u0, v0) + \nu_3 dE_3(u0, v0)$ with $\nu_1 = 0.4, \nu_2 = 0.387, \nu_3 = .213$

6. Implementation and Results

Figure 9 validates our analytical reflectance model on two different scenes. Note how the combination of our 3 modes closely approximates the Monte-Carlo simulation. The error mostly comes from assuming that the 3^+ scattering is isotropic and from the double scattering approximation. Figure 8 validates our 1D reflectance/transmittance model (the fitting of our parameters is near perfect). We validated our radiosity algorithm against an analytical case (square cloud).

For the cloud field, we rely on 512×512 animated textures precomputed on the CPU using advected textures [Ney03]. We display the cloud surface as a height field
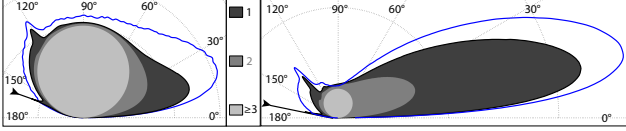
Figure 9: *Comparison of our backward multiple scattering approximated model against Monte-Carlo photon tracing.* Left: $H = 100\,\text{m}$, $\theta_l = 70°$. Right: $H = 5\,\text{m}$, $\theta_l = 80°$. Areas: *Contribution of each order of scattering.* Blue line: *Monte-Carlo result.*

Figure 10: *Close-up views on reproduced cloud features.* Left: *fogbow.* Right: *glory (6-fold zoom of the left image).*

(*i.e.,* a geometric patch) that is build on the fly on the CPU from the texture. The clouds height vary from 0 to 500m and they span several km. When the observer is on the ground and see the clouds from below, we display the cloud bottom using a textured quad. When viewed from below, the shader used at each pixel of this mesh renders

$$I^t C_{sun} + T_{ms} C_{sky} + R_{ms} C_{ground}$$

When viewed from above, the shader renders

$$I^r C_{sun} + R_{ms} C_{sky} + T_{ms} C_{ground}$$

I^t sums I_0^t (section 4.2), I_1^t (section 4.3), I_2^t (section 4.4) and I_{3+}^t (section 4.5). The latter depends on T_{3+}, which uses T_{ms}, T_0, T_1 and T_2 (appendix A). The same dependencies apply for the calculation of I^r. The radiosity algorithm makes use of R_{ms}, T_0, T_1, T_2, T_{3+} and T_{ms} (appendix A). The parameters sent to the shaders in order to compute these values are:

• $\overrightarrow{V}$, $\overrightarrow{L}$, and $\overrightarrow{N}$. Since single and double scattering are local events, we use the local normal of the surface as $\overrightarrow{N}$ for I_1 and I_2. Because multiple scattering is a global process, we use $\overrightarrow{N} = \overrightarrow{Z}$ for I_{3+}.

• The texture $h(u,v)$ used to estimate H, plus scaling factors in the 3 directions.

• Sun, Sky colors C_{sun}, C_{sky} and ground illumination texture C_{ground}.

• Phase functions $P_S(\theta)$, $P_F(\theta)$ and $P^2(\theta)$ as textures.

• The whole set of θ_l-dependent parameters needed for T_i ans R_i, passed as a 2D texture (we used the ones listed in appendix B).

At each frame we construct and send the cloud scene to the GPU, we run on GPU the radiosity simulation described in section 5, we render the background (sky or ground), then the cloud layer using our color+transparency shaders.

The implementation is done on a PC with an NVIDIA Quadro FX 1400. At resolution 1024×768, an image in which 50% of the pixels are covered by clouds (intermediates between the two images of Fig. 11.C) runs at 18 FPS. One major bottleneck during clouds animation is the update of the 512x512 height field geometry. When clouds are not animated, the frame rate increases to 40 FPS.

Our results are illustrated in Figure 11 and in the companion video. They show the various expected features including "pseudo-specular" reflection (Fig. 11.C), "silver lining" (Fig. 11.A, 11.D), fogbow (fig. 10, left) and glory (fig. 10, right) , and the radiosity effects including water sky and ice blink(Fig. 11.A,right, 11.E). See figure 5 for comparison with real photographs.

7. Conclusion and future work

We presented a method to render cloud layers in real-time providing all expected optical features, as well as inter-reflections with the ground, which was not feasible with existing methods. However, our model is restricted to stratiform clouds. We would like to carry on this approach to all kinds of animated cloudy skies (*e.g.,* cumulonimbus storms or congestus fields), and from any point of view. This method should be easily transposable to ice-content stratiform clouds, where only the phase function differs.

We assumed that the rays always pass through the cloud bottom. This neglect interesting behaviors at the cloud sides. We plan to explore better estimation for these configurations.

Relying on geometry for clouds is very efficient as compared to the massive overdraw caused by methods with sliced volumes. Still, it lacks the blurry effect on the silhouette. We aim at adapting Gardner's technique [Gar85] for that.

The accuracy of our model could be further improved: the pseudo-specular part is slightly underestimated (see fig. 9) and yields overdarkened parts on top of the cloud. This is because the diffuse mode I_{3+} does not fully match the real behavior well in thin areas, where it keeps some anisotropy.

Radiative transfer can be improved as well. We did not consider the inter-reflections between different parts of the cloud (which usually brightens creases), and between different clouds. Aerial perspective should be taken into account for distant interactions. Also, the interaction with a non-flat ground has yet to be treated.

Still, we were able to achieve real-time rendering on detailed scenes while reproducing the main features of clouds aspect. Such complex lighting effects where not possible with previous methods, and we believe our approach opens promising avenues of future work toward realistic real-time cloud rendering.

8. Acknowledgments

A. Top: *Final rendering of clouds top, I_3^+ contribution, sky contribution, $I_1 + I_2$ contribution, ground contribution.* **Bottom:** *Final rendering of clouds bottom, I_3^+, sky, $I_0 + I_1 + I_2$ and ground contributions. (see Figure 2)*

B. *View from under the clouds without (*top*) and with radiosity (*bottom*).*

C. **Left:** *Clouds in full day-light.* **Right:** *Clouds at sun-set. Notice the strong contribution of the scatter terms (pseudo-specular).*
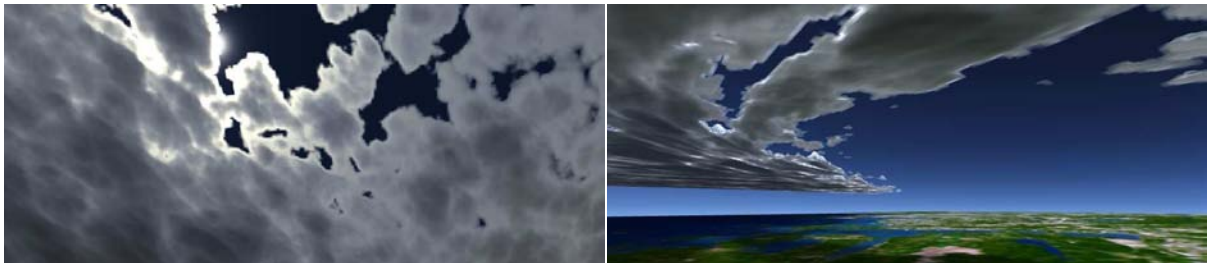
D.Left: *Sun through the clouds, seen from the ground.* **Right:** *Clouds in full day light, seen from the ground.*
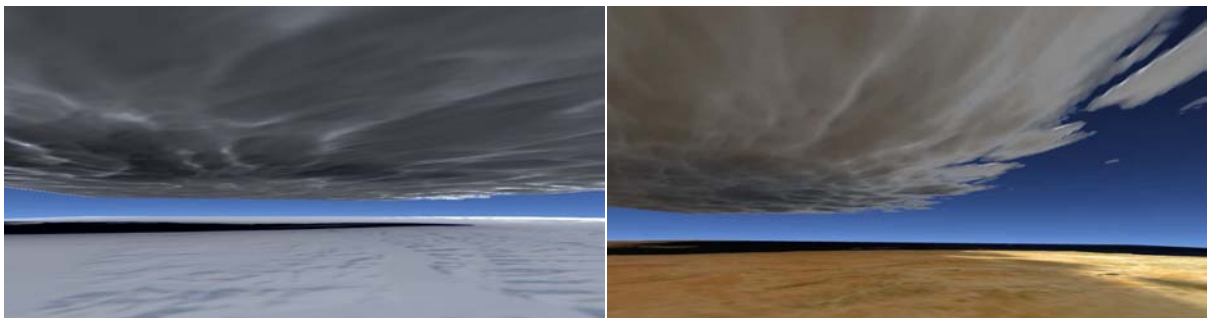
E.Left and right: *Two instances of the* water sky / ice blink *effect. Clouds located just above the water appear darker than above ice or sand.*

Figure 11: *Results obtained at 18 to 40 fps with 512×512 clouds animated heightfields and 3 to 10 pairs of radiosity passes.*

References

[Art] ARTIC CLIMATOLOGY AND METEOROLOGY: Water sky and ice blink.
http://nsidc.org/arcticmet/basics/phenomena/water_sky.html.

[BH98] BOHREN C. F., HUFFMAN D. R.: *Absorption and Scattering of Light by Small Particles*. Wiley-Interscience, 1998. ISBN: 0471293407.

[Bli82] BLINN J. F.: Light reflection functions for simulation of clouds and dusty surfaces. In *SIGGRAPH'82* (1982), pp. 21–29.

[Cla74] CLARK T. L.: A Study in Cloud Phase Parameterization Using the Gamma Distribution. *Journal of the Atmospheric Sciences 31*, 1 (1974), 142–155.

[Cow] COWLEY L.: Atmospheric optics.
http://www.atoptics.co.uk/ .

[Dei64] DEIRMENDJIAN D.: Scattering and polarization properties of water clouds and hazes in the visible and infrared. *Applied Optics IP 3* (Feb. 1964), 187–196.

[Dei69] DEIRMENDJIAN D.: *Electromagnetic Scattering on Spherical Polydispersions*. American Elsevier Publishing Company, New York, 1969. ISBN 0444000380.

[DKY*00] DOBASHI Y., KANEDA K., YAMASHITA H., OKITA T., NISHITA T.: A simple, efficient method for realistic animation of clouds. In *SIGGRAPH'00* (2000), pp. 19–28.

[Gar85] GARDNER G. Y.: Visual simulation of clouds. In *SIGGRAPH'85* (1985), ACM Press, pp. 297–304.

[HL01] HARRIS M. J., LASTRA A.: Real-time cloud rendering. *Computer Graphics Forum 20*, 3 (2001), 76–84.

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH'98* (1998), ACM Press, pp. 311–320.

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *SIGGRAPH '01* (2001), pp. 511–518.

[JW97] JACKEL D., WALTER B.: Modeling and rendering of the atmosphere using mie-scattering. *Computer Graphics Forum 16*, 4 (1997), 201–210.

[Kv84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. In *Computer Graphics (Proceedings of SIGGRAPH 84)* (July 1984), vol. 18, pp. 165–174.

[Lav] LAVEN P.: Mieplot.
http://www.philiplaven.com/mieplot.htm.

[Lev58] LEVIN L. M.: Functions to represent drop size distribution in clouds. the optical density of clouds. *Izv. Akad. Nauk. SSSR, Ser. Geofiz. 10* (1958), 198–702.

[LL95] LYNCH D. K., LIVINGSTON W.: *Color and Light in Nature*. Cambridge University Press, 1995. ISBN: 0521775043.

[Max94] MAX N. L.: Efficient light propagation for multiple anisotropic volume scattering. In *Fifth Eurographics Workshop on Rendering* (June 1994), pp. 87–104.

[Min54] MINNAERT M.: *The nature of Light & Color in the open air*. Dover Publications, 1954.

[MSPS04] MAYER B., SCHRÖDER M., PREUSKER R., SCHÜLLER L.: Remote sensing of water cloud droplet size distributions using the backscatter glory: a case study. *Atmospheric Chemistry and Physics 4* (Aug 2004), 1255–1263.

[MVC00] MILES N. L., VERLINDE J., CLOTHIAUX E. E.: Cloud Droplet Size Distributions in Low-Level Stratiform Clouds. *Journal of the Atmospheric Sciences 57*, 2 (2000), 295–311.

[Ney03] NEYRET F.: Advected textures. In *ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA)* (july 2003).

[NND96] NISHITA T., NAKAMAE E., DOBASHI Y.: Display of clouds taking into account multiple anisotropic scattering and sky light. In *SIGGRAPH 96* (Aug. 1996), Computer Graphics Proceedings, pp. 379–386.

[O'N05] O'NEIL S.: *GPUGems 2 : Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Addison-Wesley, 2005, ch. 16 A GPU Framework for Solving Systems of Linear Equations, pp. 253–268.

[PAT*04] PREMOZE S., ASHIKHMIN M., TESSENDORF J., RAMAMOORTHI R., NAYAR S.: Practical rendering of multiple scattering effects in participating media. In *Rendering Techniques: Proceedings of EGSR'04* (June 2004), pp. 363–374.

[REK*04] RILEY K., EBERT D. S., KRAUS M., TESSENDORF J., HANSEN C.: Efficient rendering of atmospheric phenomena. In *Rendering Techniques: Proceedings of EGSR'04* (June 2004), pp. 375–386.

[SP89] SILLION F., PUECH C.: A general two-pass method integrating specular and diffuse reflection. In *Proceedings of SIGGRAPH '89* (1989), ACM Press, pp. 335–344.

[Sta95] STAM J.: Multiple Scattering as a Diffusion Process. In *Rendering Techniques: Proceedings of EGWR'95* (1995), Springer-Verlag, pp. 41–50.
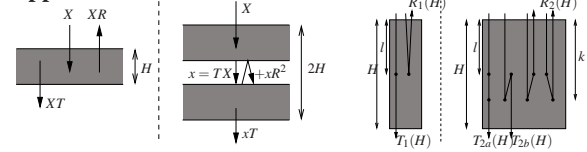
Appendix A: 1D models

Figure 12: Left*: 1D multiple scattering model, where X is the incoming intensity.* Right*: 1D single scattering model, 1D double scattering model.*

Multiple scattering Let us assume we know the transmittance $T_{ms}(H)$ of a slab of height H through multiple scattering. The reflectance is $R_{ms}(H) = 1 - T_{ms}(H)$. A $2H$ slab can be seen as two slabs of height H stacked together. If x is the amount of light traversing the interface between the two layers in the transmissive direction, we have $x = T_{ms}(H) + xR_{ms}^2(H)$ whose solution is $x = \frac{1}{2 - T_{ms}(H)}$ (see Fig. 12, left). The amount of light transmitted by the $2H$ layer is what is left from x after traversing the second layer. Thus $T_{ms}(2H) = \frac{T_{ms}(H)}{2 - T_{ms}(H)}$. By induction, we get $T_{ms}(H) = \frac{\beta}{H - (H-1)\beta}$ with $\beta = T_{ms}(1m)$. We find β by fitting against a Monte Carlo simulation (see appendix B).

But isotropic approximation is valid for a large thickness H only. Below a certain critical thickness (several hundred meters in our case), $T_{ms}(H)$ does not fit the real behavior which becomes strongly dependent on θ_l. We found experimentally that this error can be well corrected by an exponential factor. We use
$$T_{ms}(H) = (b + (1-b)e^{-cH})\frac{\beta}{H - (H-1)\beta}$$
where b and c are correction parameters depending on θ_l that we fit against a Monte Carlo simulation (see appendix B and Fig. 8).

Single and double scattering For the single and double scattering approximation, we consider a 1D column holding particles with an extinction function $\tau_c(x) = e^{-\kappa_c x}$ and $s_c(x) = \kappa_c e^{-\kappa_c x}$ (see Fig. 12, right). In the 1D case, light can only be scattered in 2 directions: reflective (coefficient r) or transmissive (coefficient t). With this model, the 1D single

scattering solution is

$$T_1(H) = \int_0^H s_c(l)\alpha^t \tau_c(H-l)dl = tHs_c(H)$$

$$R_1(H) = \int_0^H s_c(l)\alpha^r \tau_c(l)dl = \frac{1}{2}r(1-\tau_c(2H))$$

Transmittance from double scattering can happen either by double-backward or double-forward scattering.
$T_2(H) = T_{2a}(H) + T_{2b}(H)$ with

$$T_{2a}(H) = \int_0^H s_c(l)t \int_0^{H-l} s_c(k)t\tau_c(H-k)dk\,dl$$
$$= \frac{1}{2}(\kappa_c tH)^2\tau_c(H)$$

$$T_{2b}(H) = \int_0^H s_c(k)r \int_0^k s_c(k-l)r\tau_c(H-l)dl\,dk$$
$$= \frac{1}{4}r^2(1+(2H\kappa_c-1)\tau_c(2H))\tau_c(3H)$$

Reflectance from double scattering can happen when being scattered backward then forward, or vice-versa.

$$R_2(H) = 2\int_0^H s_c(l)r \int_0^l s_c(l-k)t\tau_c(k)dk\,dl$$
$$= \frac{1}{2}rt\left(1-(2H\kappa_c+1)\tau_c(2H)\right)$$

Here again, the θ_l-dependent parameters κ_c, t and r are fit against Monte Carlo simulation (see appendix B and Fig. 8). As for b and c, these parameters account for the bias resulting from modeling a 3D phenomenon with a 1D model. That is why they vary with θ_l, and why $r \neq 1-t$.

Transparency is directly given by the extinction function:
$$T_0(H) = \tau(H_l)$$

Appendix B: Used values

Here is an example set of parameters used in appendix A for the DSD used in the examples, *i.e.,* $N_0 = 300\text{cm}^{-3}$, $\gamma = 2$ and $r_e = 7\mu\text{m}$. We obtained them by running several photon tracing simulations at relevant values of H and θ_l and fitting the parameters by optimization. The fitting is very good, as displayed on figure 8.

$\beta = 0.9961$ (constant).

θ_l	b	c	κ_c	t	r
0°	1.1796	0.0138	0.0265	0.8389	0.0547
10°	1.1293	0.0154	0.0262	0.8412	0.0547
20°	1.1382	0.0131	0.0272	0.8334	0.0552
30°	1.0953	0.0049	0.0294	0.8208	0.0564
40°	0.9808	0.0012	0.0326	0.8010	0.0603
50°	0.9077	0.0047	0.0379	0.7774	0.0705
60°	0.7987	0.0207	0.0471	0.7506	0.0984
70°	0.6629	0.0133	0.0616	0.7165	0.1700
80°	0.5043	0.0280	0.0700	0.7149	0.3554
90°	0.3021	0.0783	0.0700	0.1000	0.9500