Obscurant representation for realistic IR simulation

Patrick Gozard[(a)], Alain Le Goff[(b)], Jean Latger[(c)], Thierry Cathala[(c)] ,
[(a)] DGA/DCE/ETBS, BP 712, 18015 Bourges, France
[(b)]DGA/DCE/CELAR, BP 7419, 35174 Bruz cedex, France
[(c)]OKTAL Synthetic Environment, 2 rue de Boudeville, 31100 Toulouse, France : www.oktal-se.fr

ABSTRACT

Obscurant representation is a key component of ground battlefield simulation, especially in the infrared domain. Obscurant are special counter measures (clouds) classically used to hide armored vehicles and deceive threatens.
Obscurants are very difficult to represent especially because of multi diffusion effects of hot particles and smoke, but this representation is very important to quantify the efficiency of the decoy.
This article describes a new model being involved in the simulation workshop CHORALE of the French DGA [1], [2] & [3]. This new model enables to simulate the emitted radiance and the transmission of any pre computed obscurant cloud in the virtual battlefield.
In the modeling step, the cloud is defined by a set of "voxels" (elementary volume elements). Each voxel contains the spectral extinction coefficient and the spectral scattering coefficients. The shape, i.e. the voxels content, is changing with time to convey the dynamic evolution of the obscurant.
In the Non Real Time rendering step, primary rays are traced inside the clouds. For each voxel, scattering rays are then traced to their neighboring voxels and the local hot sources. Actually, ray tracing is used to solve the Radiative Transfer Equation. The main advantage is to be able to solve it taking into account the synthetic environment: the local terrain, the target hidden in the cloud, the atmospheric and weather conditions. The main originality is the multithreading ray tracing which enables to tackle huge quantities of rays in complex geometric environment.

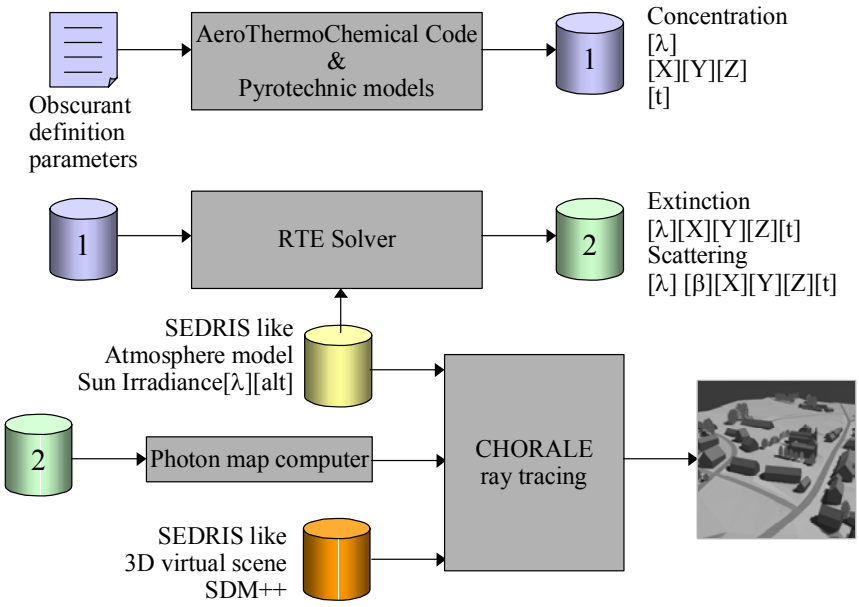
Keywords: Simulation, ray tracing, infrared, obscurant, cloud, scattering, diffusion, radiative transfer, multithreading.

1. INTRODUCTION

Obscurant are special clouds generated using specific military pyrotechnic devices manufactured in France by specialized company such as GIAT or Lacroix. Obscurants are used in the ground or marine battlefield to hide from hostile means for detection, mainly in the infrared domain. New concepts of use studies, efficiency assessment, ejection modeling are basic topics in the scope of simulation. The DGA CHORALE workshop, based on OKTAL Synthetic Environment company COTS, is being upgraded with a special software module enabling to simulate multi diffusion effects of hot particles and smoke due to obscurant. The main originality is that the IR obscurant rendering is part of a more global rendering including atmosphere, thermal environment, terrain, vegetation, buildings and vehicles.

2. GENERAL ARCHITECTURE

The multi diffusion rendering is part of a more general tool, under development in the DGA/ETBS. This tool is divided into 3 parts. The third part is detailed in this paper.
The first stage consists in modeling obscurants using AeroThermoChemical codes and pyrotechnic models. The second step is a Radiative Transfer Equation solver, using finite difference method, which transforms concentrations into optical parameters such as scattering and extinction coefficients. The third step aims at importing the obscurant in a 3D virtual battlefield and rendering it using ray tracing IR imagery, inheriting the sophisticated facilities of the operational simulation workshop called CHORALE.



3. CHORALE

CHORALE is devoted to 3D virtual scene generation. This scene reproduces a realistic environment of an operational theater for one or several weapon systems with sensors. That for, it includes:

- the objects, that can be targets or active sources such as luminous fires and counter measures,
- the background (sky, terrain, sea), on or before which the objects are inserted,
- the atmosphere, that constitutes the optic radiation propagation canal,
- the sky, comprising heavenly bodies (sun, moon, stars) and clouds.

The other part of CHORALE concerns computation of the physical signal, as perfect as possible, received by a sensor that watches the 3D scene.

CHORALE is based on a generic kernel consisting of efficient set of ray tracing functionality. This kernel possesses original capabilities: computation time is nearly independent on the scene complexity especially the number of polygons, databases are enhanced by precise physical and thermal data, the ray casting is linked off line with specific software simulating meteo and environment effects (LOWTRAN, MODTRAN, MOSART, EOSAEL), special mechanisms of antialiasing have been developed that enable to consider very accurate details in the field of view, a generalization of texture definition allows to simulate directional dependence of the emissivity and reflection factors, specific categories of objects are characterized such as 3D clouds, obscurants and flares (IR decoys). The thermal realism is achieved by taking notably into account radiant flux, spread shadows, material permeability and specific phenomena due to vegetation.

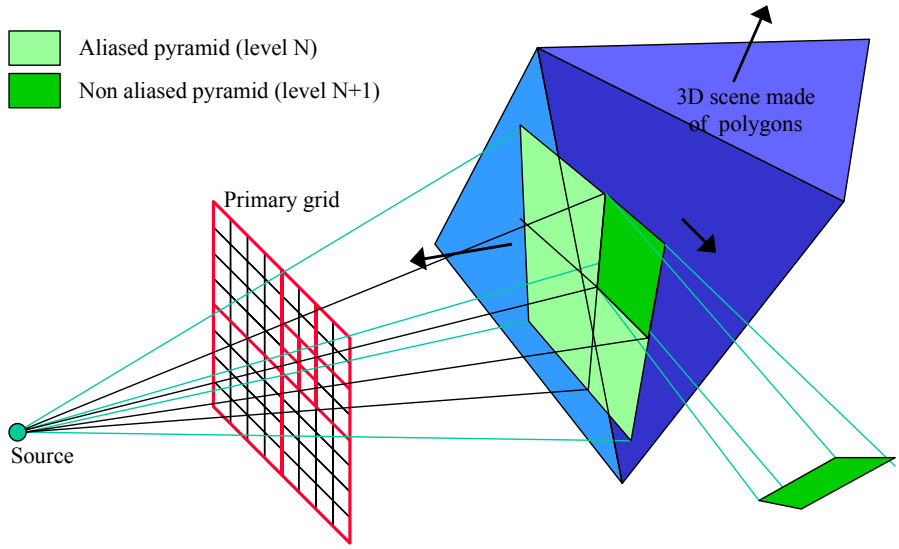
4. SPECRAY

The ray-tracing core of CHORALE is called SPECRAY.
The time consumption is very optimized using SPECRAY. Actually performances are nearly independent on scene complexity. To do this, SPECRAY uses a spatial subdivision method that enables to get a perfect knowledge of the scene topology before computing the first image. Except for moving objects, which possess a special treatment, this topology is static and available till the database doesn't change. Scene space is decomposed in a hierarchy of volume elements that both

contain the list of inner objects and topological relations with the other elements. Space scene is then turned into a recursive space of volume elements, which efficiently improves the intersection computations.

SPECRAY implements a very original method for “sparing rays” called antialiasing.. The aliasing artifact classically occurs when a polygon becomes smaller than the ray spot, for instance due to the distance. This is the more important artifact concerning the physical aspects. In any case the solution to improve quality mainly consists in over sampling by tracing more rays. The best method, regard to physical requirements, is the adaptive one. The idea is that the density of rays is proportional to the local 3D complexity. For instance a few rays are traced to a uniform huge polygon when many rays are traced to a very complex small target. The most important antialiasing criteria are the number of different polygons in the ray spot, the number of different materials, and the normal vector variation within the ray spot.
The user can modify the sensor basic resolution (number of pixels), the maximum oversampling factor necessary for adaptive antialiasing, the reflection maximum depth for specular materials, the scattering maximum depth. A set of parameters can be modified that are used by SPECRAY to characterize the adaptive antialiasing algorithms. For instance, the user can introduce a minimum angle for the normal vector of a surface between two basic rays so that a new ray can be traced by SPECRAY in case of high variations of the normal vector.

Adaptive sampling regards to normal vector variation for reflection process

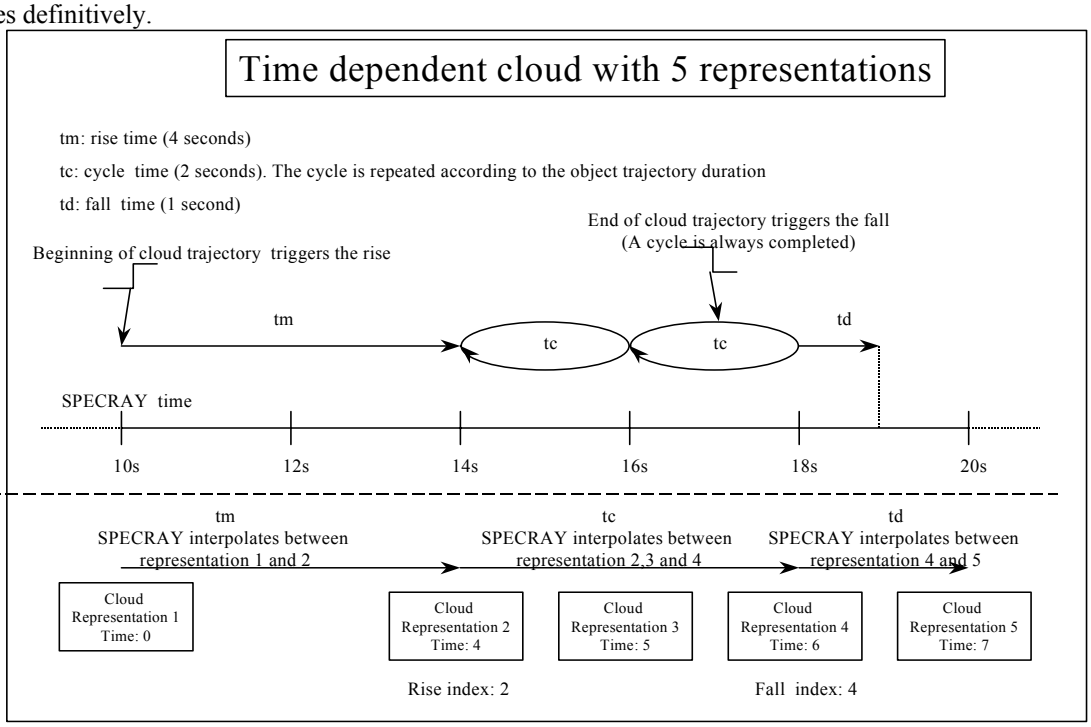


5. CLOUD REPRESENTATION USING VOXELS

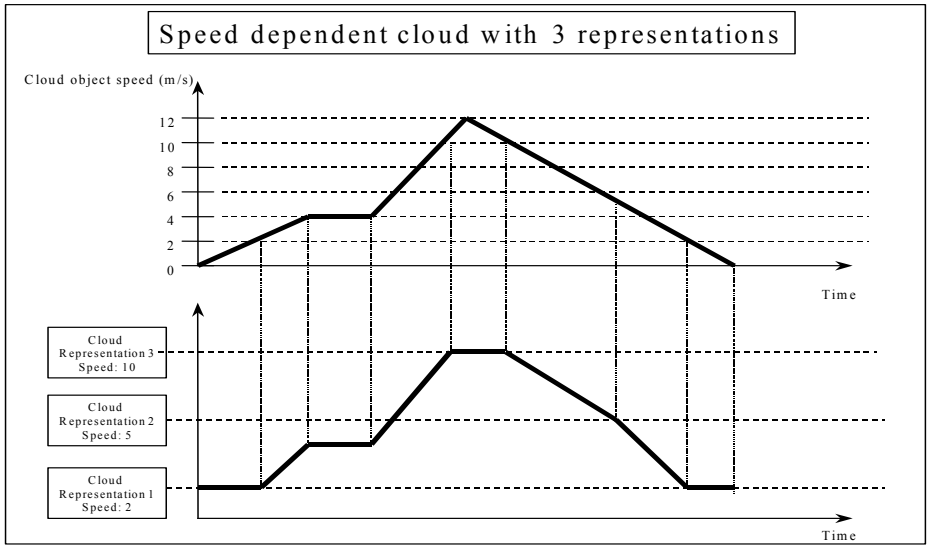
A cloud is a complex 3D model defined by a set of regular voxels. Each voxel is determined for each date. In the previous version of SPECRAY, one voxel included equivalent spectral emissivity and spectral transmission. In the current version of SPECRAY, spectral emissivity corresponding to the retro diffusion is replaced by a general scattering function. These data are spectral depending and time depending to simulate the cloud’s expansion. Indeed, cloud representation contains time and/or speed information in such a way that SPECRAY can animate a cloud object, interpolating, for each voxel (according to time and/or speed), the radiance and attenuation values between the different representations. The cloud object must have a trajectory to be animated.

A cloud can evolve according to two kind of parameters:

The Time: to represent smoke-producing phenomena or clouds that change their morphological characteristics (shape, molecular distribution, etc.) in time. In this case, the cloud has an initialisation phase, representing the firing of a smoke-producing phenomena or the cloud formation; a cyclical animation phase; and an extinction phase. The end of the initialisation and the beginning of the extinction phases are identified by cloud representation indexes associated each one to a time, serving as boundaries between the evolution phases. The initialisation phase starts with the **activation** of the cloud and ends at the time supplied for its termination. After this time, the evolution phase starts and the animation loops between the cloud representation indexes explained above. When the cloud **deactivation** is triggered, the evolution continues with the first representation associated with the extinction phase, going on until the last representation, after which the cloud extinguishes definitively.

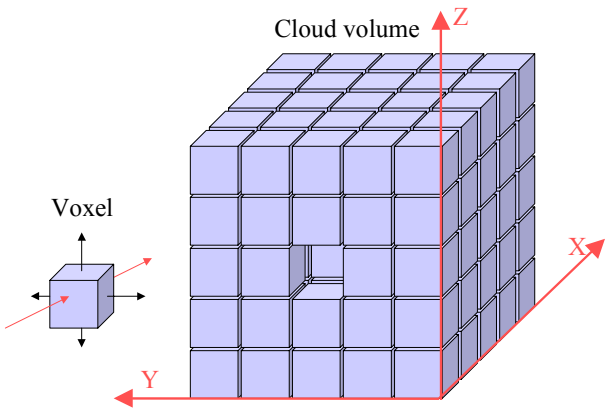


The Speed: to represent dust or exhaustion gas clouds coming from a moving object. The choice of the cloud representation only depends on the speed associated to the cloud. There is no **activation** or **deactivation** for this kind of cloud.



Actually, this model is very powerful because it is a real 3D set of voxels. Typically it means that the interaction with a target within the cloud is accurately simulated. Cloud model can be applied to counter measure smoke or obscurant, dust cloud, aircraft or helicopter plume or ground vehicle plume and atmospheric clouds.

3D voxels clutters called clouds can be defined using external data file. Each voxel is characterized by a scattering component and a transmission component. When a ray crosses a voxel, scattering and transmission of the voxel is computed taking into account the optical range within the voxel.



6. WHAT IS PHOTON MAPPING

The photon map algorithm was developed in 1993–1994 and the first papers on the method were published in 1995. It is a versatile algorithm capable of simulating global illumination including caustics, diffuse inter reflections in very complex scenes. It provides the same flexibility as general Monte Carlo ray tracing methods using only a fraction of the computation time. The global illumination algorithm based on photon maps is a two-pass method.
The first pass builds the photon map by emitting photons from the light sources into the scene and storing them in a *photon map* when they hit non-specular objects.
The second pass, the rendering pass, uses statistical techniques on the photon map to extract information about incoming flux and reflected radiance at any point in the scene. The photon map is decoupled from the geometric representation of the scene. This is a key feature of the algorithm, making it capable of simulating global illumination in complex scenes containing millions of triangles, instanced geometry, 3D volumes and complex procedurally defined objects.

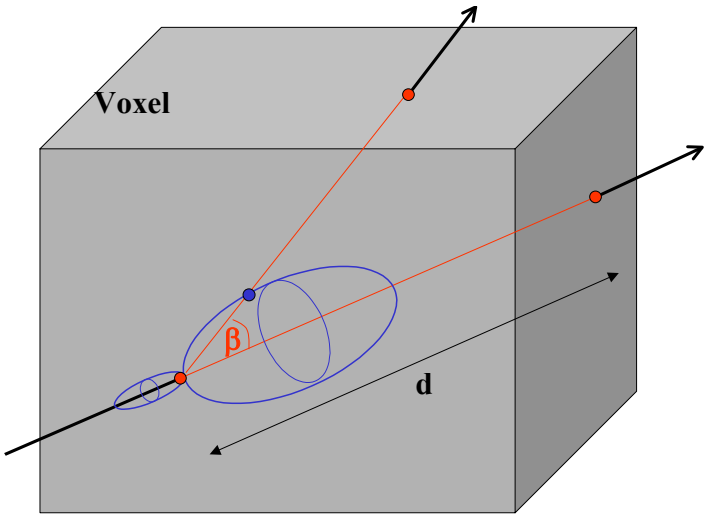
Compared with finite element radiosity, photon maps have the advantage that no meshing is required. The radiosity algorithm is faster for simple diffuse scenes but as the complexity of the scene increases, photon maps tend to scale better. The main benefit of the photon map compare with classical methods is efficiency, and the price paid is the extra memory used to store the photons. For most scenes the photon map algorithm is significantly faster, and the result looks better since the error in the photon map method is of low frequency which is less noticeable than the high frequency noise of general Monte Carlo methods. This is the key advantage for clouds representation which is both complex in term of voxels amount and smooth or filtered in term of spatial frequencies.

6.1. Photon Mapping applied to surfaces

OKTAL SE has already applied photon maps techniques to solve multi diffusion problems inside very hot objects rendered in the infrared spectrum.
Results are very good so the technique has been assessed for participant media, also rendered in the infrared spectrum, such as obscurant clouds for military applications with French DGA. Currently, this technique finds two applications with French DGA: one for ground battlefield counter measures using CHORALE workshop, and one for marine application in partnership with MBDA.

6.2. Photon Mapping applied to 3D voxels representing obscurant clouds

Obscurant clouds are simulated using a 3D box of voxels in the X, Y and Z direction. Currently, the spacing is regular in each direction. It is planned to replace this regular model by an "octree" representation. In the octree structure, each elementary cube can be recursively divided into 8 cubes, which enables to fit to non-homogeneous 3D repartition in the cloud. Each voxel is first characterized by an extinction coefficient, k, depending on wavelength. If d is the ray path within the voxel, the radiance attenuation is computed as: $\exp(-k(\lambda)\times d)$.
Moreover, each voxel contains a scattering coefficient, μ, theoretically dependent on incident and scattered angles: $\mu(\lambda,\theta_{in},\theta_{out},\phi_{in},\phi_{out})$, which can be approached by $\mu(\lambda,\beta)$, where β is the angle between the ray and the scattered direction.



6.3. Photon management

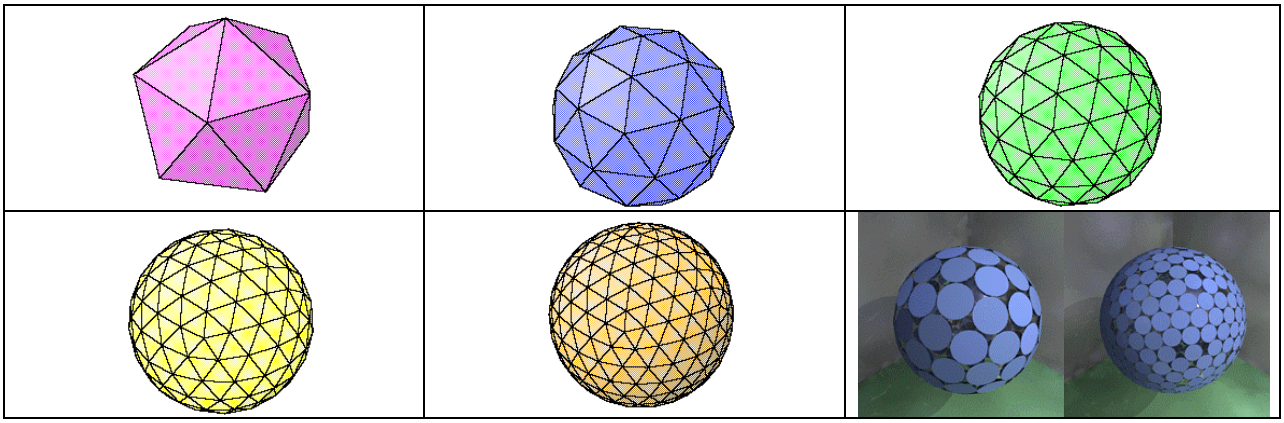
The purpose of the photon-tracing pass is to compute indirect illumination on diffuse parts of the scene. This is done by emitting photons from the light sources, tracing them through the scene, and storing them at diffuse voxels.

→ **Photon emission**

The scene i.e. the volume made of aggregated voxels, contains multiple light sources. As a consequence, photons should be emitted from each light source. More photons should be emitted from brighter lights i.e. hotter particles than from dim lights i.e. warmer particles, to make the power of all emitted photons correct. One might worry that scenes with many light sources would require many more photons to be emitted than scenes with a single light source. Fortunately, it is not so. In a scene with many light sources, each light contributes less to the overall illumination, and typically fewer photons can be emitted from each source.

→ **Photon tracing**

Once a photon has been emitted, it is traced through the set of voxels using photon tracing (also known as "light ray tracing", "backward ray tracing", "forward ray tracing", and "backward path tracing"). Photon tracing works in exactly the same way as ray tracing except for the fact that photons propagate flux whereas rays gather radiance. This is an important distinction since the interaction of a photon with a material can be different than the interaction of a ray.
The first step consists in defining the discrete direction in free space to trace photons. Two different approaches exist: tracing photons either in isotropic directions, which can be done using a regular meshing of a 360° sphere centered on the photon, or in random directions based on stochastic sampling. The method actually implemented is a combination of these two approaches i.e. a regular sampling modulated by a random law.



Then, when a photon hits a neighbor voxel, it can either be reflected, transmitted, or absorbed. Actually it can be either absorbed (to simulated attenuation) or scattered in the β direction. Whether it is scattered, or absorbed is decided probabilistically based on the material parameters of the intersected voxel. The technique used to decide the type of interaction is known as "Russian roulette" basically we roll a dice and decide whether the photon should survive and be allowed to perform an-other photon-tracing step.
Why do we go through this effort to decide what to do with a photon? Why not just trace new photons in the scattered directions and scale the photon energy accordingly? There are two main reasons why the use of « Russian roulette » is a very good idea. Firstly, we prefer photons with similar significant power in the photon map. This makes the radiance estimate much better using only a few photons. Secondly, if we generate, say, 26 photons per interaction then we will have 26x26 photons after 2 interactions. This means 676 photons after 2 interactions compared to 1 photon coming directly from the light source! The use of Russian roulette is therefore very important in photon tracing.

→ **Photon storing**

Photons are so stored for each voxels. For each photon-voxel interaction, the position, incoming photon power, and incident direction are stored. A photon is effectively stored if the power attached to the photon is greater than a pre defined threshold.

→ **Photon rendering**

Given the photon map and the ability to compute a radiance estimate from it, we can proceed with the rendering pass. The photon map is view independent, and therefore a single photon map constructed for an environment can be utilized to render the scene from any desired view. There are several different ways in which the photon map can be visualized.
Actually, a filter is to be defined to sum the different contribution of discrete photon and interpolate these contributions for every point in the 3D scene.

7. CONCEPTION KEY POINTS

The first technical difficulty concerns the effective strategy to cast photons in 4π steradians. Correlated works made by OKTAL SE in the field of antialiasing technique, implemented in the CHORALE ray-tracing kernel, SPECRAY, have been reused.
The second key point is the physical model associated with the "Russian roulette" approach. The problem is to link the probabilistic laws applied, in term of amount of absorbed and scattered photons, to the physical values of extinction and scattering distribution functions.
The final point is the algorithm of filtering of the discrete photons which has to fit to physics.

8. DUAL USE OF THE MODEL

There are two modes of obscurant rendering. The first mode consists in storing the 3D photons map in the voxels for each step of time. Let us remind that the dependence on time is due to the expansion of the obscurant cloud, the shape of which is continuously changing. This mode is much more efficient in term of rendering since global illumination is already computed. In this case, a simple voxels traversal using ray tracing is sufficient. The problem is that the pre computed global illumination supposes no masking inside the cloud. As a consequence, it supposes that there is neither intersection of the cloud with the terrain, nor any vehicle hidden inside the obscurant. This is the reason why the second mode is available. In this mode, the photon map construction and the rendering is done in the same process.

9. TECHNICAL CHALLENGE

The technical challenge is both the time of computation, mainly to create the photon map which supposes multi scattering through a great amount of voxels, and the memory space, mainly to store the photon map. In term of storing, let us say that the photon map is multi dimensional: wavelength (dependence of scattering function), time or date (due to cloud expansion), voxel amount (typically 100 x 100 x 100).

10. VALIDATION PROCESS

The general idea is to compare a finite element method used to solve the Radiative Transfer Equation and the ray tracing method. The two different ways are planned to be used to reach the same result. Special target cases are to be considered such as uniform and homogeneous obscurants, by night to be independent to sun illumination. Moreover, confrontation to real images is to be performed by French DGA, in order to validate the simulation model.

11. TO BE CONTINUED

A lot of optimizations are to be done, especially in term of time computation and memory allocation. Currently the ray tracing can take advantage of a cluster of heterogeneous computer (Windows, UNIX, Linux) using CORBA for multithreading, which reduces noticeably the elementary time of computation of an image. Besides, a simplified real time implementation is planned for the next generation tools, based on Open GL.

REFERENCES

1. Alain Le Goff, , Jean Latger, "Realistic multi spectral simulation including IR simulation", SPIE Proceedings, Vol. 3694, April 1999

2. Alain Le Goff, , Thierry Cathala, "Automatic temperature computation for realistic IR simulation ", SPIE Proceedings, Vol. 4029, April 2000

3. Henri Mametsa, Frédéric Rouas, André Bergès, Jean Latger, "Imaging radar simulation in realistic environment using shooting and bouncing rays", SPIE Proceedings, Vol. 4543, September 2001