A Realistic Rendering Algorithm for an Intuitive Interpretation of 3D Scalar Fields

Philippe Blasi, Charles Wuethrich

LaBRI [1]
351 cours de la Libération
33405 Talence (FRANCE)

Faculty of Media [2]
Bauhaus University Weimar
Coudraystr. 13
D-99423 Weimar (GERMANY)

blasi@labri.u-bordeaux.fr, caw@informatik.hab-weimar.de

Abstract : We present in this paper a method for rendering three dimentionnal scalar fields. We make an accurate bijection between scalar fields and a participating media (haze) in order to get an intuitive interpretation of the field. We propose a physic based illumination model, with no restrictive assumptions about the characteristics of the medium (arbitrary phase functions) and about the physical phenomena included in the rendering process (multiple scattering). The algorithm allows to vary continually from surfacic to volumic visualization and to visualize only a part of the field. Examples on a 3D image of a sampled magnetic field illustrate the different possibilities of the method.

Keywords : 3D Image, Participating Media, Monte-Carlo Ray-Tracing, Multiple Light Scattering.

1 Introduction

Scientific visualization relies heavily on computer graphics. There is a wide range of domains of applications : medical imaging, meteorology, interpretation of experimental data ... A difficult problem is rendering the notion of depth on a two dimensional image. Accurate image synthesis, with good lightning models and shadowing, is necessary to render it. In the present paper, we study the visualization of three dimensional scalar fields. They are functions giving for each point of space, identified by three coordinates x, y and z, a single value related to the displayed phenomenon : intensity of a magnetic field, geological data ... This value can be numerically computed or empirically measured. The scalar field is often represented under a discrete form, as a finite set of sample points. The spacial distribution of these points depends on the chosen data structure : 3D image [Sabe88], mesh nodes [Inak91], octrees [Meag82], ... Since they are defined in three dimensions and they have no surfaces, visualizing scalar fields is problematic . A common approach consists in defining within the scalar field a surface and displaying it. The surface can be simply defined by cross-sectionnal planes or composed of polygonal surfaces generated by processing the field cells [LoCl87]. But these technics only display a part of the scalar field, ignoring much of data contained in it. The other approach consists in considering the whole scalar field and displaying it as a surfaceless three dimensional object. Sabella [Sabe88] proposed a technique for visualizing 3D scalar field in which the field is rendered as a varying density emitter object. His technic uses some elements of the physic of light in cloud-like objects.

---

[1]Laboratoire Bordelais de Recherche en Informatique. Université Bordeaux I et CNRS URA 1304. The present work is also granted by the *Conseil Régional d'Aquitaine*.

[2]This work is also supported by Silicon Graphics Germany.

In this paper, we chose to go farther in this way. We make bijection between the 3D scalar field and a well known object, haze density, in order to allow an intuitive interpretation of the resultant image. We propose a new visualization algorithm, based on an accurate and realistic rendering algorithm [BLS93], accounting for multiple scatterings and phase functions. We use both 3D images and mesh nodes to represent the scalar field.

2 Rendering Model

In order to visualize a 3D scalar field as a whole, without any loss of data, we must represent it with a translucent volumic object whose main characteristic corresponds to the value of the field. A well known object in every day life meeting these requierements is haze. We make a bijection between values of the 3D scalar field and haze density. The perception of density comes from the amount of light absorbed and scattered by the medium. Thus, we decided to use an accurate realistic rendering algorithm for the haze, considering that the more realistic the image is, the more intuitive its interpretation becomes. Therefore, this algorithm does not make restrictive assumptions for the medium and accounts for multiple scatterings of light and phase functions.

2.1 Characterisation of the Medium

Haze is a participating medium. It is composed of small drops of water that interact with light according to two characteristics : the absorption and scattering efficient section of the particle, expressed in square meter. The density of the medium is expressed as the number of particles by cubic meter : $N_{particles}.m^{-3}$. The most important phenomena which are involved in physics of participating media are absorption and scattering. Absorption corresponds to the transformation of energy from the visible spectrum into warmth. Scattering corresponds to a distribution of energy in the whole space when a ray of light intercepts a particle — such a distribution is usually described by a phase function (Rayleigh or Mie phase function). Absorption efficient section, scattering efficient section and density defined as above are not very convinient to manipulate. Because absorption and scattering depend linearily on these three values, we prefer to use the following coefficients :

- absorption and scattering coefficients, expressed in m^{-1}, characterize the medium
- relative density is a dimentionless value, ratio of the volume occupied by particles to the total volume.

These coefficients characterize the whole volume object, instead of the particles making it up.

2.2 Rendering Equations

To get realistic pictures of participating media, we need to solve two equations : the *image rendering equation* (Equation 1) and the *scene rendering equation* (Equation 2). The first one defines the illumination of a given pixel in the image. It is essentially a mean over several integration dimensions (pixel width, pixel height, lens radius, lens angle, shutter time). The second function defines the radiance of each couple (P, V) where P is a point and V a direction of the scene. It is a Fredholm equation that expresses the transport of light in the environment. See [BLS93] for a more complete explanation of these equations.

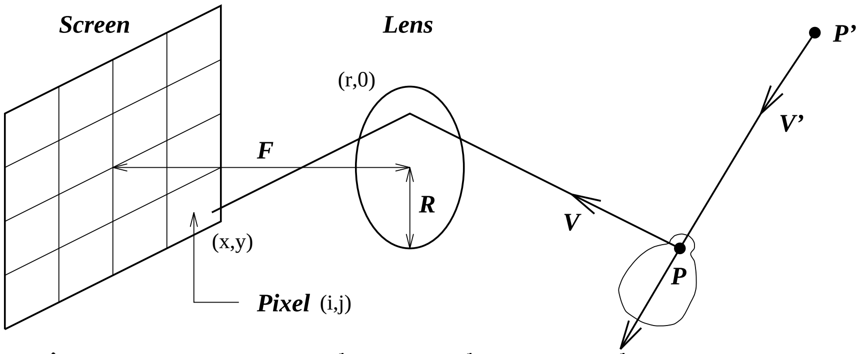


Figure 1 : Image rendering and scene rendering equations

$$I(i,j) = \frac{1}{T\pi R^2} \int_x \int_y \int_r \int_\theta \int_t K_{\mathrm{V}}(M,P)\, L(P,V)\, dx\, dy\, r\, dr\, d\theta\, dt \tag{1}$$

$$L(P,V) = L_{\mathrm{E}}(P,V) + \int_{V' \in \mathcal{V}} K_{\mathrm{S}}(P,V,V')\, K_{\mathrm{V}}(P,P')\, L(P',V')\, dV' \tag{2}$$

- $I(i,j)$: Illumination of pixel (i,j)
- $L(P,V)$: Radiance leaving P in direction V
- $L_{\mathrm{E}}(P,V)$: Emitted radiance leaving point P in direction V
- $K_{\mathrm{S}}(P,V,V')$: Surface scattering factor at point P between directions V and V'
- $K_{\mathrm{V}}(P,P')$: Volume attenuation factor between points P and P'
- $\mathcal{V}$: Set of directions for incident light (solid angle 4π)
- $T\pi R^2$: Normalization factor (T is the shutter time and R is the lens radius of a virtual camera)
- dV' : Differential solid angle element surrounding direction V'

We consider these two equations on a somewhat particular point of view. The points P and P' are scattering points. They can be either on the surface of a solid object or inside a volume object. The $K_{\mathrm{S}}(P,V,V')$ — surface scattering factor — coefficient expresses the distribution of energy in space at a scattering point P. It is the ratio of radiances between two directions V and V' at P. The $K_V(P,P')$ coefficient — volume attenuation factor — expresses the attenuation of light traveling through a participating medium. It is the ratio of radiances between two scattering points P and P' (in the vaccum, K_V yields 1). P and P' are supposed to be two successive scattering points, therefore the only physical phenomenum that can occur between these points is absorption. Thus K_V is given by Bouguer's law restricted to absorption :

$$K_{\mathrm{V}}(P,P') = e^{-\int_P^{P'} \alpha\ \rho(P'')\, dP''} \tag{3}$$

where α is the absorption coefficient and $\rho(P)$ is the density at the point P. The surface reflection factor $K_S(P,V,V')$ expresses the scattering at a point P where a ray intercepts the surface of a particle :

$$K_{\mathrm{S}}(P,V,V') = \frac{1}{4\pi}\ \gamma\ \varphi(V,V') \qquad \text{and} \qquad \gamma = \frac{\sigma}{\alpha+\sigma} \tag{4}$$

where $\varphi(V,V')$ is the phase function of the object, γ the albedo, σ is the scattering coefficient and α the absorption coefficient.

The phase function $\varphi(V,V')$ expresses the ratio of energy propagated in direction V compared to the energy coming from direction V'. In physic litterature, phase functions are very complex and approximations are usualy used (see [BLS93] for more details).

We propose the following approximation :

$$\varphi_{r,k,k'}(t) = r\ \varphi_k(t)\ +\ (1-r)\ \varphi_{k'}(t) \quad \text{where} \quad r \in [0,1] \ \ k \in \,]-1,1[\ \ \ k' \in \,]-1,1[\tag{5}$$

This phase function presents numerous advantages. The most important is that the function is inversible and integrable, two properties necessary for our method of resolution of the Rendering Equations.

Volume objects are defined as a set of particles whose position in space is random, but whose distribution is known. Thus locations of scattering points are computed in a probabilistic way, using a probability of interception of a ray by a particle. The probability of interception increases as a function of the density, the scattering coefficient and the distance covered by the ray in the object since the last tested point. In Bouguer's law, the scattering coefficient expresses the fraction of incoming energy which is lost by scattering. Therefore it is natural to define the probability of interception $\omega(P, P')$ as :

$$\omega(P, P') = 1\ -\ e^{-\int_P^{P'} \sigma\ \rho(P'')\ dP''} \tag{6}$$

3 Monte-Carlo Solution of the Rendering Equations

3.1 Principle of Monte-Carlo methods

Rendering equations are multidimensional integral equations that cannot be solved analytically. Numerous numerical techniques have been proposed to evaluate them, including several Monte-Carlo based methods [Cook84, Kaji86]. Monte-Carlo methods are well adapted to such problems. They enable to solve multidimensional integral equations as if they were monodimensional, by using separated Markov chains [Hamm64, Kaji86]. The principle of the proposed techniques is the use of a stochastic ray-tracing scheme to evaluate statistically the scene rendering equation. But a rough statistical evaluation of this equation can result in a great variance and a very slow convergence speed.

Several sampling techniques have been developed for Monte-Carlo methods to overcome these stumbling blocks [Hamm64]. Importance sampling (or weighted sampling) allows to reduce the variance and speed-up the convergence by oversampling the function when it is high and by undersampling it when it is low. For a given distribution function f, an *optimal importance sampling* can be obtained from a uniform sampling by using the inverse function F^{-1} of the repartition function F associated with f. So, any evaluation of $t = F^{-1}(u)$, where u is a uniform random variable, provides a stochastic weighted value of t (see [ShWa92] for some applications of this result in computer graphics). Consequently, it becomes very interesting to use a inversible and integrable distribution phase function for our surface scattering factor.

In our context, the role of the distribution function is played by the surface scattering factor. Several expressions for this factor can be found (given by different theories in physics) but, unfortunately, none of them is integrable and inversible, and thus cannot provide optimal sampling.

The solution we propose to circumvent this limitation is to approximate theoretical expressions with more simpler functions which are integrable and inversible. We have developped an approximation technique which enables such simplifications while preserving in good accuracy with the original functions. This scheme has been applied both for light scattering on surface objects and on volume objects [BLS93, BLS94].

3.2 Our Rendering Method

3.2.1 Overview

The two pass algorithm we propose is a Monte-Carlo simulation of a particle model of light in participating media. It simulates the interaction of light with the particles of the participating medium and visualize it. This medium is modelized as a 3D image, i.e a tridimensional grid of voxels. The density of the medium is assumed to be constant within each voxel.

3.2.2 Energetization Pass

During this pass, rays carrying energy are sent stochastically from each emitting source, according to its goniometrical distribution. Each ray progresses incrementally throughout the medium, step by step. During each step, the energy carried by the ray is scaled by the attenuation factor. The length of the steps varies between zero and the size of a geometrical voxel, according to the density and the scattering coefficient. The length is computed in order to have, as far as possible, the same probability of interception at each step. The probability of interception of the ray is computed using the length of the step and the density of the medium at the sampling point. If no interception occurs, the ray is simply propagated a step further in the same direction. Otherwise, the energy of the ray is stored with its direction in the storage structure and we perform an importance sampling of the phase function to cast a new ray propagating the scattered energy from the scattering point.

At the end of the illumination pass, for each optical voxel, the stored incident energies on the sampled sphere are converted into scattered radiances using the phase function of the medium.

3.2.3 Rendering Pass

During the rendering pass of the algorithm, which is view-dependent, visualization rays are dispached from the viewer toward each pixel of the screen with an attenuation factor initialized to one. These rays travel straight foward through the medium, progressing incrementally voxel by voxel (voxel sampling). Other sampling strategies can be used to reduce possible aliasing (distance sampling, volume sampling) [SaGe92]. At each step, the ray accumulates the radiance stored in the encountered voxels during the first pass, scaled by the current attenuation of the ray. Then, the attenuation factor is scaled at each by the volume attenuation factor using the density of the current voxel and the distance covered in it. When a visualization ray leaves the participating medium, the radiance of the background is attenuated and added to the radiance accumulated by the ray. This radiance becomes the color of the pixel.

3.2.4 Storage Consideration

The storage of radiances with their directions is an expensive memory consuming problem. Several solutions have been proposed (sampled spheres [LSSC90] or spherical harmonics [SAWG91], ...). When each voxel has its own storage structure, the memory cost becomes prohibitive for large 3D images. Thus, it is necessary to lower the number of storage structures. One solution consists to store radiances not inside the volume itself, but on its surface [BLS94] (the surface of a medium is the boundary between null density voxels and the others voxels). But this storage is only interesting for high density media. For low density media, it requires an increased numbers of rays and a high sampling rate of directions for the storage structures. A new storage technic, independant of the resolution of the 3D image of densities, is used. It is a 3D image of optical voxels containing storage structures with a lower resolution than the 3D image of geometrical voxels containing densities. Consequently, its memory cost becomes reasonnable, even for large data sets.

In order to keep the geometric details of the medium and an accurate shadowing, the contribution of a geometrical voxel to the energy of the corresponding optical voxel is chosen proportional to its probability of interception. We perform the following calculations :

- For each optical voxel, we sum the probabilities of interception of the corresponding geometrical voxels and store this sum.
- At the end of the first pass, we divide the energy stored in the optical voxel by the sum.
- During the second pass, we calculate the radiance of a geometrical voxel by multiplying its probability of interception by the radiance stored into the curent optical voxel.

We use to illustrate our method a 3D scalar field representing a magnetic field inducted by an electrical dipole. Its dimensions are 64x64x64 geometrical voxels. Figure 4 shows the visualization of this scalar field with 64x64x64 optical voxels. Figure 5 show the same field, but with only 16x16x16 optical voxels. We can see there is little differences though the storage structure is 64 times smaller.

In our implementation, energies and radiances are stored around a sphere sampled by meridians and parallels. The resolution of the sampling depends on the shape of the phase function. For the isotropic phase function, we only need one sample whereas we have to increase the number of samples for narrower functions.

4 Utilization of the Method

In order to visualize a 3D scalar field, we must represent it under the discrete form of a 3D image. Each voxel of the image contains a sampled value of the field. The rendering algorithm considers this value as the density of a participating medium in order to compute the illumination of the voxel.

The scalar field used to illustrate our method is the intensity of a magnetic field generated by a dipole. It has been discretized into a 3D image of 64x64x64 voxels.

Our visualization of a 3D scalar field depends on four parameters :

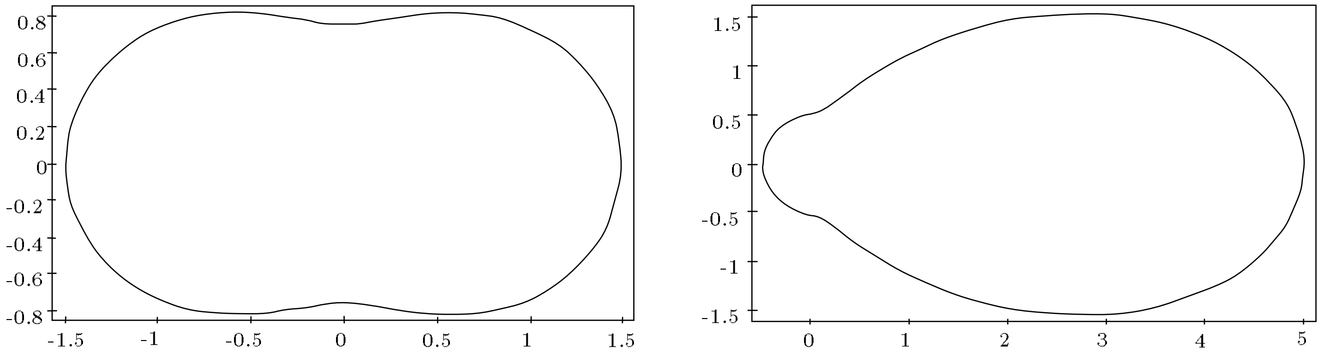
- The phase function of the participating medium.
- The caracteristics of the emitting sources : intensity, position, goniometrical distribution.
- The absorption coefficient α.
- The scattering coefficient σ.

Each of them is studied in order to evaluate their influence on the visualization. Then, we deduce how they can control the visualization.

4.1 Phase function

The phase function describes the directionnal distribution of light scattered by a particle. For spherical particles, its shape depends on the radius of the particle. When the particles are very small compared to the wavelength, there is an equal quantity of forward and backward scattering (Figure 2 (a)). For large particles with a radius of same order than the wavelength, the scatterings are strongly forward(Figure 3 (b)).

The particles of water forming haze are spherical particles with a radius of the same order than the visible light wavelength. Consequently, we have quite forward scatterings. In order to get a suitable function, the parameter k,k' and r of the phase function must be set respectively to -0.5, 0.7 and 0.12.



(a) (b)

Figure 2 : Phase functions for small particles and large particles

4.2 Emitting sources

In order to get a easy interpretation of the image, we use only white monochromatic light sources.

We use two kinds of light sources :

- Global Light Sources. They are parallel sources with constant intensity, illuminating the whole field. They are placed at a constant distance from the field. Usualy, one source is sufficient to get a good visualization.
- Local Light Sources. These sources have a goniometrical distribution and an intensity chosed by the user. They are placed manually inside or outside of the field in order to highlight a chosen part of the field. Figure 8 shows the visualization of a part of the magnetic field with parallel local light source.

The quality of the rendering depends on the number of illumination rays sent by the sources and on the dimension of the storage structure. It is necessary to send more rays for high dimensions, but the result will be more precise.

4.3 Absorption coefficient

The volume attenuation factor K_{V} is an exponential function of three variables : the length of travel through the medium, the density ρ, the absorption coefficient α. The density depends on the geometry of the medium, but we can control the absorption coefficient α to tune the visualization.

As we can see on Figure 3, K_{V} varies smoothly according to α from the constant function 1 when $\alpha = 0$ (no absorption) to a function rapidily decreasing to the asymptotic 0 (great absorption).

So, the absorption coefficient α controls the depth of the illumination and visualisation of the field. A different absorption coefficient can be used for the energetization pass and an other one for the rendering pass. When α is high, the illumination rays are rapidly absorbed and thus cannot deeply travel through the field. Consequently, only its surface is illuminated. In the same way, visualization rays can only retrieve surface illumination. The field looks like a solid object. When the absorption decease, illumination and visualization rays can travel freely through the field. It is entirely illuminated and the whole field is visible.

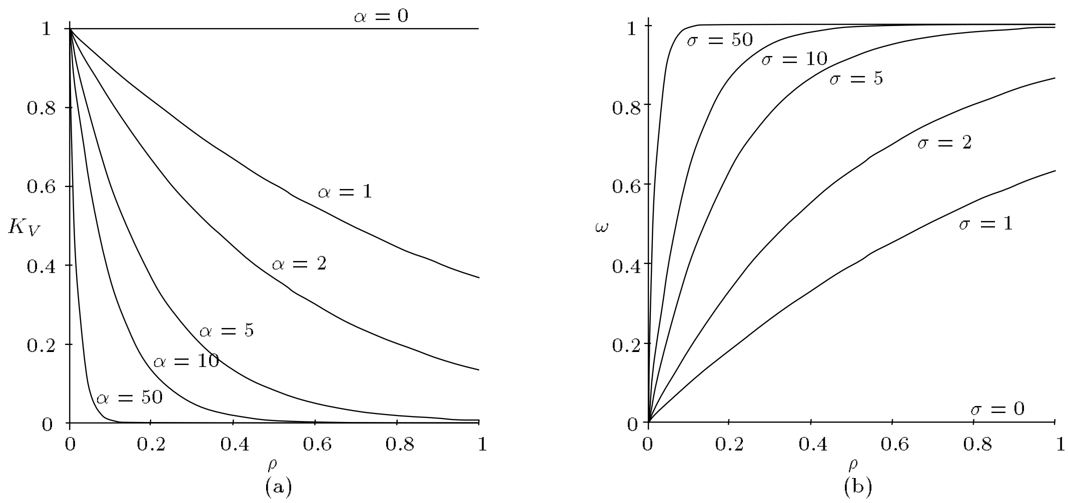


Figure 3 : Volume attenuation factor (a) and Probability of interception (b)

Figure 5 shows a field visualised with $\alpha = 0$. Figure 6 shows the same field visualised with $\alpha = 0.1$. The image is darker because we only see the radiance of the surface of the field instead of the radiance of the whole field.

4.4 Scattering coefficient

The probability of interception depends on the scattering coefficient σ : it varies smoothly according to σ from the constant function 0 when $\sigma = 0$ to a function rapidily increasing to the asymptotic 1.

Consequently, the scattering coefficient σ controls intensity of the illumination. When σ is low, only high values of the field are visible, the lower values of the field not scattering enough light to be shown. When σ is high, all the field scatters light and is consequently visible.

Figure 9 shows our illuminated field with $\alpha = 0.0$ and $\sigma = 0.1$. Only high intensity values are visible. Their shadow projected on the surface below allows to evaluate easily their position in space. When we progressively increase σ up to 0.3 (Figure 7), then 1.0 (Figure 5), the rest of the field appears progressively.

4.5 Results

The 3D image used to illustrate the method is a sampling of magnetic field inducted by an electrical dipole. Its size is 64x64x64. All the figures are 400x400 pixels images. The energetization pass of Figure 4, tooks 2300 seconds for $28.1 * 10^6$ rays and 64x64x64 optical voxels. The energetization pass of Figures 5,6,7,9 took 450 seconds for $5.6 * 10^6$ illumination rays and 16x16x16 optical voxels. For Figure 8, it tooks 64 seconds for $2.35 * 10^5$ rays. The visualisation pass took 21 seconds for Figures 5,6,7,8,9 with a definition of the radiance storage structure of 16x16x16 and 31 seconds for Figure 4. The illumination and visualization time depends almost linearily on the number of rays and on thickness of the field. The number of rays necessary to obtain a correct visualisation increases with the number of optical voxels. Figure 4, with 64 times more optical voxels than the other figures, requieres 12 times more rays. The result of one illumination pass can be used for several images with different visualization parameters : observer position and absorption coefficient. These images have been generated on Silicon Graphics Onyx 10000 work station with our all purpose rendering program.

5 Conclusion

The presented method visualizes 3D scalar fields as haze. We use a physicaly based rendering algorithm in order to get a realistic result. This realistic rendering allows a more intuitive interpretation of the image. For example, the shadow of the haze on the floor helps to localise spatialy the different parts of the field. The two most important parameters of the method are the scatering coefficient σ controlling the intensity of the illumination and the absorption coefficient α controling the depth of the illumination and the visualization. Our model is monochromatic. It can be used with the mapping into color space proposed in [Sabe88] in order to highlight high valued regions. An interresting extension is the introduction of color not as a post-process, but directly in the rendering model. The use of several colors would allow to visualize distinct intervals of values. We could select the intervals we want to visualize by only displaying their color.

6 Literature

[BLS93] P. Blasi, B. Le Saëc, C. Schlick, *A Rendering Algorithm for Discrete Volume Density Objects*, Proc. EUROGRAPHICS'93, p103-116, 1993.

[BLS94] P. Blasi, B. Le Saëc, C. Schlick, *An Importance Driven Monte-Carlo Solution to the Global Illumination Problem*, Proc. of Fifth Eurographics Workshop on Rendering, p173-183, 1994.

[Cook84] R.L. Cook, *Stochastic Sampling in Computer Graphics*, ACM Transactions on Graphics, v5, n1, p51-72, 1986

[Hamm64] J.M. Hammerley, D.C. Handscomb, *Monte-Carlo methods*, Wiley, 1964.

[Inak91] M. Inakage, *Volume Tracing of Atmospheric Environments*, Visual Computer, p104-113, 1991.

[Kaji86] J.T. Kajiya, *The Rendering Equation*, Proc. SIGGRAPH 86, Computer Graphics, v20, n3, p143-145, 1986.

[LoCl87] W. Lorensen, H. Cline, *Marching Cubes: A high resolution 3D surface construction algorithm*, Proc. SIGGRAPH 87, Computer Graphics, Vol 21, p163-170, 1987.

[LSSC90] B. Le Saëc, C. Schlick, *A Progressive Ray Tracing based Radiosity with General Reflectance Functions*, Proc. of First Eurographics Workshop on Rendering (Rennes), p103-116, 1990.

[Meag82] D. Meagher, *Geometric Modeling Using Octree Encoding*, Computer Graphics and Image Processing 19(2), p129-147, June 1982.

[Sabe88] P. Sabella, *A Rendering Algorithm for Visualizing 3D Scalar Fields*, Proc. SIGGRAPH 88, Computer Graphics, v22, n4, p51-58, 1988.

[SaGe92] G. Sakas, M. Gerth, *Sampling and Anti-Aliasing of Discrete 3-D Volume Density Textures*, Computer & Graphics, vol 16, No 1, p121,134, 1992.

[ShWa92] P.S. Shirley, C Wang, *Distribution RayTracing : Theory and Practice*, Proc. of Third Eurographics Workshop on Rendering, 1992.

[SAWG91] F.Sillion, J.Arvo, S.Westin, D.Greenberg, *A Global Illumination Solution for General Reflectance Distribution*, Proc. SIGGRAPH 91, Computer Graphics, v25, n4, p187-196, 1991.

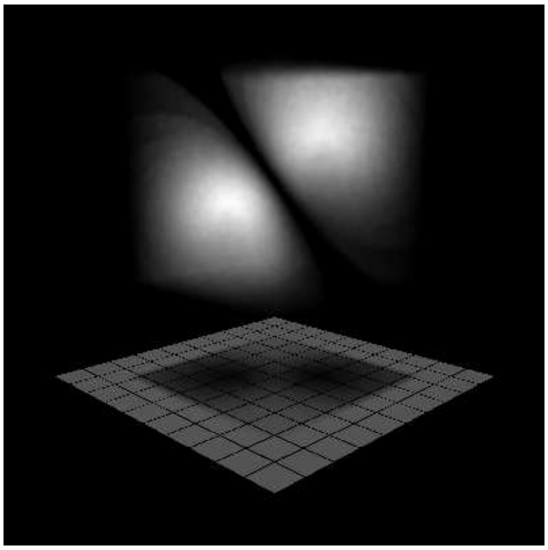


Figure 4

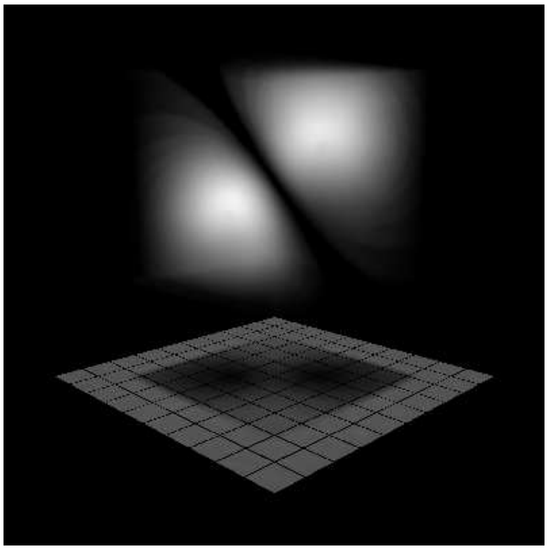


Figure 5

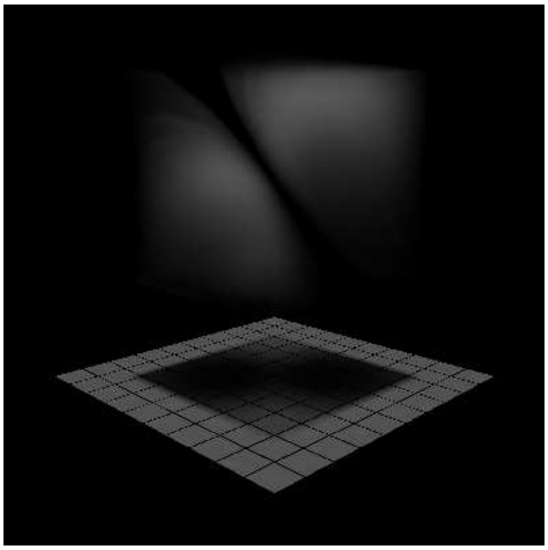


Figure 6

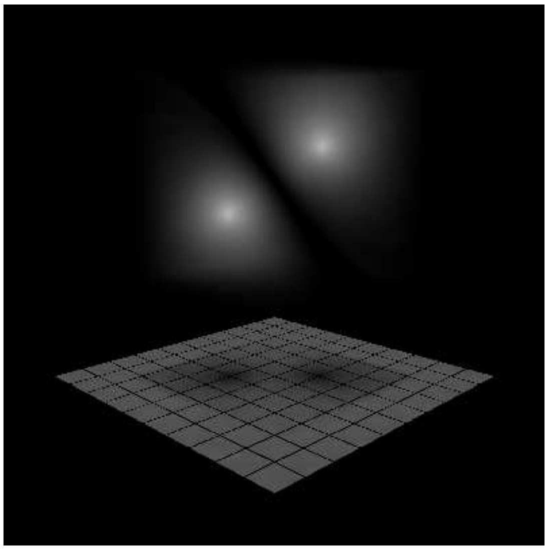


Figure 7

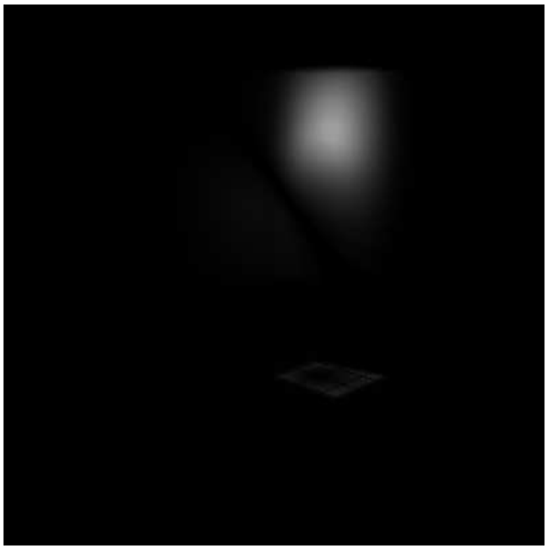


Figure 8

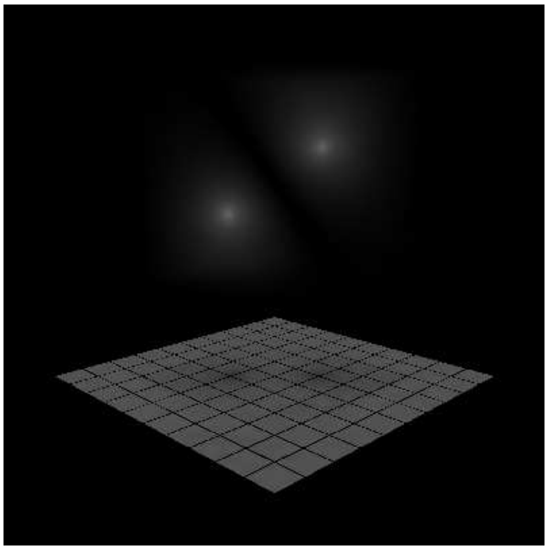


Figure 9