A Rendering Algorithm for Visualizing 3D Scalar Fields

Paolo Sabella*
Schlumberger-Doll Research

Abstract

This paper presents a ray tracing algorithm for rendering 3D scalar fields. An illumination model is developed in which the field is characterized as a varying density emittter with a single level of scattering. This model is equivalent to a particle system in which the particles are sufficiently small. Along each ray cast from the eye, the field is expressed as a function of the ray parameter. The algorithm computes properties of the field along the ray such as the attenuated intensity, the peak density, and the center of gravity, etc.. These are mapped into HSV color space to produce an image for visualization.

Images produced in this manner are perceived as a varying density 'cloud' where color highlights the computed attributes. The application of this technique is demonstrated for visualizing a three dimensional seismic data set.

CR Categories: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; I.4.0 [**Image Processing**]: General

General Terms: Algorithms

Additional Key Words and Phrases: 3D image, ray tracing, thresholding, light scattering.

*Author's curent affiliation: Visual Edge Software
Author's current address: 37 Dragon St.
Kirkland, PQ H9J-3B3 Canada

©1988 ACM-0-89791-275-6/88/008/0051 $00.75

1. Introduction.

Computer graphics has made an irreplaceable contribution in the presentation of scientific data. The computer is indispensable as an aid in the visualization of certain kinds of functions. This paper focuses on the display of single valued functions of three variables, i.e. scalar fields. Such functions can be analytically defined but more commonly are computed numerically using finite difference or finite element techniques. Examples of such fields are the stress distribution over a mechanical part or the pressure distribution within a fluid reservoir.

Alternately, the field can be empirically measured. For example, sonic waveforms or tomography radiation measurements can be processed to obtain density samples of a solid over a three dimensional volume. A convenient representation for this is a three dimensional array of samples referred to as a **3D image**. The sample points are called voxels (volume elements). The cuberille [3] is a special case of the 3D image in which all the voxels are identical cubes. The retangular prism that is the spatial extent of the array is called the **image extent**.

The 3D image is not the only representation for sampled scalar fields. Octrees [14, 12] provide a more compact, structured means of encoding field samples in which both the spatial extent and the image are hierarchically organized. In finite element analysis, samples of a field are computed at the mesh nodes. Hence the mesh itself serves as a representation for the field.

In this paper we use the 3D image representation. Figure 7 shows a 3D image of sub-soil density obtained from surface seismic measurements. The density value is mapped into a gray scale. Notice that most of the cube, i.e. the interior, is not readily visible.

Although the objectives of manipulation 3D data is application dependent, the goal when visualizing the data is to understand the spatial distribution of the field values over the domain on which the function is defined. It is important to be able to see the location of the occurrence of any range of values, such as, for example, high values or 'hot spots' or changes in the field gradient. It is also important to perceive the manner in which the function varies from it's minimum to maximum values. This is especially true for geophysical imaging in which there is hardly any spatial correlation in the data and which does not lend itself to existing visualization techniques.

In this paper a new rendering model for visualizing scalar fields is proposed. In it, the field is rendered as a varying density emitter (DE) object. It is related, but not restricted, to the modeling of light in naturally occurring cloud-like objects. The goal here is the perception and understanding of the field, not the realistic rendering of natural phenomena. The visualization technique is unique in three ways:

i) A color map is utilized to visualize individual attributes.
ii) The use of a phase function to visualize the field gradient.
iii) The Kajiya and Von Herzen ray tracing algorithm is recasted for computational efficiency, eliminating shadowing while retaining occlusion.

We first review the two existing rendering models of 3D images. Variations of these account for all attempts to date at rendering scalar fields. Next we look at scattering models and particle systems used to render realistic looking clouds and natural phenomena. We then propose the varying density emitter (DE) object which is actually a special case of a general scattering emitter and a generalization of the particle system. Finally, a ray tracing algorithm is described for rendering the DE object incorporating false coloring in order to present multidimensional information.

Although the data presented in the figures actually represent a sampled density field, the visualization model is intended to be used on any field. It can be adapted to render analytically defined fields or to incorporate bounding surfaces such as those of a solid over which the field is defined. It can also be extended to render octrees or finite element meshes.

2. Rendering Models for 3D images

There are two common techniques to date for rendering 3D images, cross-section rendering and threshold rendering. Each of these implicitly are models of the interaction of light on the 3D model. It is important to place the DE rendering model within the context of existing schemes.

2.1 Cross-section rendering

In cross-section rendering, the 3D image is considered to be an opaque array of voxels packed inside the image extent. A user can interactively remove portions of the image in order to see interior voxels. The rendering model is that of light illuminating the cross-sectional surfaces, or slices.

Good response time is valuable here in order to perceive the field in three dimensions. This perception is achieved by the user sequentially viewing multiple slices and mentally interconnecting the features of interest. For example, in order to search for a region of maximum value the user may scan through slices until one is found. Then many sections taken in possibly different orientations are viewed to see the extent of the peak region.

An important consideration is the coloring of the voxels. Color is used both to indicate the field value at a voxel as well as to provide surface orientation cues. The cross-section is therefore a texture mapped surface [1]. Usually the intensity is varied according to a shading model while the base color, either hue or gray scale is used to indicate field value.

2.2 Threshold rendering

Threshold rendering is common in medical applications when dealing with 3D images of densities measured by computer aided tomography (CAT) scanners . In this application there are different categories of materials each falling within a specific density range. In order to render a particular category, for example the bones, the rendering model considers voxels falling outside the range to be non-existent. Consequently, the resulting view is of the surface of a constant field value. Figure 8 shows a threshold rendering of the same 3D image shown in figure 7.

Thresholding is important since it is a three dimensional view of iso-value surfaces. There are three categories of rendering techniques for thresholding, back to front traversal of the 3D image [7], ray tracing [5], and surface reconstruction [11]. A variation of the threshold illumination model is to assume that the iso-value surface has a normal vector, computed from the field gradient. This allows the iso-valued surface to be shaded using Gouraud or Phong shading [8].

The threshold rendering model throws away much of the data held in the 3D image. Except for applications, like medical imaging, in which a small number of density ranges are of interest, it is like cross-section rendering in that it requires many views of the field which have to be mentally combined in order for a user to entirely perceive the field. In an application like geophysical imaging, a 3D seismic image does not portray as much coherence as a medical image. Hence a threshold rendering is not satisfactory because it generates fragmented pieces instead of cohesive surfaces.

3. Scattering Models

We wish to see more than just iso-value surfaces when rendering a 3D image. It is natural therefore to establish a model for viewing three dimensional translucent solids. We do not often encounter such solids with varying translucency in every day circumstances. However clouds, smoke, mist and other systems of suspended particles are common. Through these we do have experience in perceiving variations of density. A haze in the atmosphere has been modeled by Dugan [4] and opaque clouds, represented as elipsoidal height fields, was modeled by Fishman and Schacter [6]. However true cloud-like scenes are achieved by modeling the scattering of light.

Although the goal here is not the rendering of natural clouds, it is important to consider the interaction between light and a medium that occurs in clouds so that it may be used to exploit our ability to perceive density fields.

3.1 Background

Blinn's model [2] of the rings of Saturn was the first graphics model of the scattering through a thin uniform density cloud of low albedo. Max [13] modeled clouds bounded by quadric surfaces. The emphasis was in modeling the enclosing volume within which the particles reside.

More recently, Kajiya and Von Herzen [9] introduced a ray tracing technique for rendering 'volume density' models, their term for a 3D image of a density field. They presented a solution for multiple radiative scattering.

Another amorphous class of model is the particle system introduced by Reeves [16,17]. While they have been used to model many natural phenomena, such as blades of grass, the use of particle systems to model fire is most striking. This is because the particles are point light sources that are additively combined when rendered.

There are three effects observed when illuminating translucent objects.

i) Occlusion of a portion of the model occurs when light is scattered by the portions of the cloud closer to the observer.

ii) Shadows are created depending on the position of the light source.

iii) Color variations are caused by separation due to differing amounts of scattering at different wavelengths.

All of these are highly desirable cues for the realistic rendering of clouds. Computationally however, they are expensive; especially when combined with secondary effects like multiple scattering.

3.2 Varying Density Emitters

For the purpose of viewing scientific data, shadowing and scattering color variations, while visually appealing, may actually detract from perception of the density variation. On the other hand, since occlusion is proportional to density, it enhances this perception. A system that solely exhibits occlusion would be a good one to use, at the same time keeping the computation cost low. Such a system is a field of varying density emitters.

A DE object is a system of particle light sources. Unlike Reeve's particle system in which particles are modeled individually the DE object models the density of particles, not the particles themselves. The size of the particle is sufficiently small compared to other dimensions so that the density of the particles can be regarded as a continuous function.

For convenience we define the density ρ(x,y,z) non-dimensionally as the ratio of the volume occupied by particles, Vp to the total volume of the cloud V. Since the ratio is only valid locally we have

$$\rho(x,y,z) = \frac{dVp}{dV} \ .$$

This is actually the probability that a particle is present at the point (x,y,z). Each particle has a volume υ_p. The expected number of particles within a region Ω is

$$N_\Omega = \frac{\int_\Omega dVp}{\upsilon_p} = \frac{\int_\Omega \rho \ dV}{\upsilon_p} \ .$$

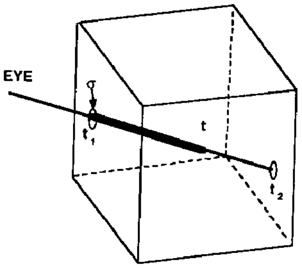

Figure 1 The volume encountered before the ray parameter t.

3.3 The Brightness Equation

Figure 1 sketches the path of a ray traversing the cube. We follow a derivation for the intensity of light, of a fixed wavelength, reaching the eye that is similar to Blinn [2] but allow for varying density and ignore self illumination. The density field ρ(x,y,z) can be parameterized along the ray as ρ(x(t),y(t),z(t)), or simply ρ(t). If each particle has an intensity κ, a cylindrical volume element of cross sectional area σ and length dt at a point t along the ray contributes

$$\kappa N_\sigma = \kappa \frac{\rho \ dV}{\upsilon_p} = \frac{\kappa\sigma}{\upsilon_p} \rho(t) \ dt$$

to the intensity of light emitted toward the eye.

This light emitted by the volume element is scattered backward when traveling on it's way toward the eye due to the particles lying within the volume between t and t_1. The expected number of particles N in this volume, V, is

$$N = \frac{\sigma}{\upsilon_p}\int_{t_1}^{t} \rho(\lambda)\, d\lambda \quad .$$

The intensity of light reaching the eye is equal to the light emitted, attenuated by the probability, P(0;V), that there exists zero particles in the volume V. Assuming ρ is small, this can be approximated by a Poisson distribution

$$P(0;V) = e^{-N} .$$

The intensity of the light reaching the eye from the point t is thus

$$I(t) = P(0;V)\ \kappa N_\sigma = e^{-N}\ \frac{\kappa\sigma}{\upsilon_p}\rho(t)\ dt \quad . \qquad (1)$$

The total intensity due to all contributions along the ray between t_1 and t_2 is the integral of (1)

$$B = \frac{\kappa\sigma}{\upsilon_p}\int_{t_1}^{t_2} e^{-\frac{\sigma}{\upsilon_p}\int_{t_1}^{t}\rho(\lambda)\, d\lambda}\ \rho(t)\ dt \quad . \qquad (2)$$

The term $\frac{\sigma}{\upsilon_p}$ can be replaced by one constant τ. It has dimensions 1/length and is related to the optical length. We can normalize the entire equation by choosing

$$\frac{\kappa\sigma}{\upsilon_p} = 1$$

since we are not dealing with actual radiation measurements. Equation (2) simplifies to

$$B = \int_{t_1}^{t_2} e^{-\tau\int_{t_1}^{t}\rho(\lambda)\, d\lambda}\ \rho(t)\ dt \qquad (3) \quad .$$

This is a simplified version of Kajiya and Von Herzen's brightness equation, omitting the line integrals from the light source [9]. In order for this to hold we have assumed that ρ(t) is small. We can adjust the value of τ to scale ρ(t) in the exponent integral. Higher values of τ increases the attenuation producing a medium that darkens more rapidly.

Finally, taking advantage of the fact that ρ lies between 0 and 1, the transformation

$$\rho' = \rho^{\gamma}$$

is used to control the spread of density values. Higher γ intensifies the appearance of dense portions relative to the more diffuse regions while lower γ makes the entire cloud appear more diffuse. This transformation is order preserving, i.e. the ordering of two values $\rho'(\lambda_1)$ and $\rho'(\lambda_2)$ is the same as that of $\rho(\lambda_1)$ and $\rho(\lambda_2)$. It is useful to spread out the function ρ since not all density variations are perceptible.

The total brightness along a ray is:

$$B = \int_{t_1}^{t_2} e^{-\tau\int_{t_1}^{t}\rho^{\gamma}(\lambda)\, d\lambda}\ \rho^{\gamma}(t)\ dt \qquad (4) \quad .$$

In figure 5 the field in figure 7 is ray-traced using equation (4). The results of varying the two parameters γ and τ are show . Each image is normalized to use the full intensity range. The image lies within a unit cube. As expected, the images on the upper row, in which τ = 1, are not sufficiently attenuated. This is corrected by making the value of τ larger than the distance t_2-t_1 in the second and third rows τ = 2 and τ = 3.

The density spread, controlled by γ, is highest in the first column in which γ = 1. In the adjacent columns γ equals 5 and 10 respectively. We observe, as predicted in figure 2a, the spread narrows and the higher densities dominate.

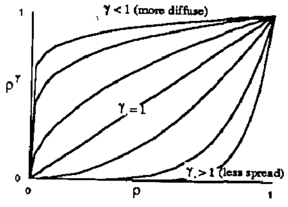

Figure 2a Transformations on ρ

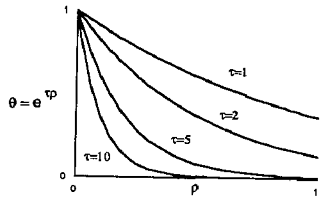

Figure 2b Transmittance functions

3.4 Equivalence to Discrete systems.

The model utilized in discrete systems for transmission of light through translucent surfaces or particles in a particle system is equivalent to equation (3). The total intensity remaining after traversing n translucent particles or surfaces, is

$$\sum_{i=1}^{n} b_i \prod_{j=1}^{i-1} \theta_j \ , \qquad (5)$$

Where b_i is the amount of light emitted from the i^{th} surface either by reflecting light received from light sources or due to the fact that the surface is itself a light source. The transmittance factor of the i^{th} surface, i.e. the proportion of entering light allowed to go through the surface, is θ_i. It varies between 0 and 1. Since the i^{th} surface is seen through the preceding i-1 surfaces, the total attenuation of b_i is the product of the preceding i-1 θ terms.

Equation (5) is actually equivalent to the discrete form of equation (3)

$$B = \sum_{i=1}^{n} e^{-\tau \sum_{j=1}^{i}\rho(t_j)\ \delta t_j}\rho(t_i)\ \delta t_i$$

$$= \sum_{i=1}^{n}\rho(t_i)\ \delta t_i \prod_{j=1}^{i} e^{-\tau\rho(t_j)\ \delta t_j} \qquad (6) ,$$

in which the surface for i=1 is at t=t_1 and the surface for i=n is at t=t_2. The i^{th} surface has a thickness δt_i over which the density can be assumed constant. The equivalent terms are

$$b_i = \rho(t_i)\ \delta t_i \ ,$$

and

$$\theta_j = e^{-\tau\rho(t_j)\ \delta t_j} \ . \qquad (7)$$

The threshold rendering model explicitly controls the relationship between θ and ρ. Figure 3a is an example of a threshold window on ρ and figure 3b shows the corresponding θ, ρ relationship.

The equivalent functions for the DE model are shown in figures 2a and 2b. The shapes of these graphs indicate that the DE model is a continuous version of the threshold model. In the limit, as γ goes to infinity, we obtain a threshold model with a range of zero width around ρ = 1.

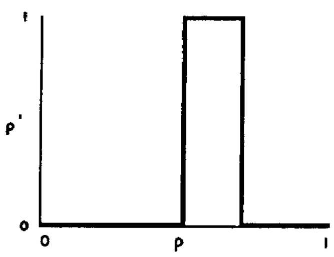

Figure 3a. A threshold window

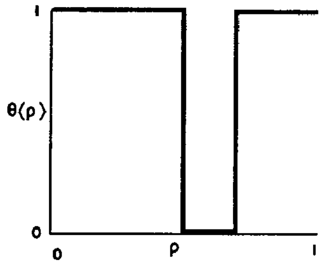

Figure 3b. The corresponding transmittance function

4. Mapping into Color Space

Since we have ruled out the color variations caused by differential wavelength scattering we are left with the question of whether or not to incorporate the use of color at all. Color can be the result of a realistic model of light or it can be utilized symbolically. Even though models of color space are three dimensional, human perception of color is not.

Robertson and O'Callaghan in [18] give details of their scheme to simultaneously display multiple 2D image data sets. In it, one image is rendered realistically as the surface of a height field while another is used to determine the hue of the displayed surface. The rendering of a DE object is different from their application in that here we are interested in rendering one 3D image. However, we can adapt the idea that realism cues the human visual system thereby enhancing the perception of additional variables.

In ray tracing a surface model, a ray is fired from the eye through a pixel to determine the pixel color. The closest surface that is encountered is the one whose color is chosen as a base color for the pixel. Even though the final color is computed by taking into account light sources and the shading model, the surface color indicates **symbolically** which surface was encountered. The same is true recursively for reflected and refracted rays.

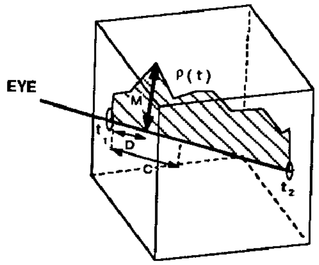

Figure 4. Properties along the ray

We have a more difficult problem when ray tracing a field. Instead of choosing a color that indicates which surface was hit, the ray encounters the density function ρ(t). Since it is not possible to choose one color that will uniquely represent all the values of the function over the ray, we use color to indicate certain characteristic properties of the function. The properties chosen to compute along the ray are (see figure 4):

$M = (\mathrm{Max}(\rho(t)))_{t_2}^{t_1}$ Peak value encountered along the ray,

D = Distance at which the peak value is encountered,

I = The attenuated intensity, equation (4),

$$C = \frac{\int_{t_1}^{t_2} \lambda\rho(\lambda)\, d\lambda}{\int_{t_1}^{t_2} \rho(\lambda)\, d\lambda}$$ The centroid or center of gravity.

In Robertson and O'Callaghan's scheme, the underlying model is that of rendering a surface whose height was determined by one data set and whose color was based on a model of paint pigmentation. The colors chosen were evenly spaced in the uniform color space proposed by Meyer and Greenburg [15].

We take a simpler view with regard to the choice of color scale. The goal in our scheme is the rough identification of extrema such as 'hot spots'. It is important to realize that such a scale is **symbolic** and are not intended for reading absolute values or distances.

Our scheme was implemented in HSV space. In it the Value component, corresponding to the light intensity, was obtained from the I variable and the Hue component from the peak encountered along the ray, M. The hue scale was reversed to be a scale in which 'hot' colors, reds, represent high values.

Saturation can be a strong depth cue because decreasing saturation gives the effect of seeing in a fog [10]. Hence a depth parameter such as D or C was chosen for the Saturation component.

To illustrate the scheme a charge field was computed for two point charges. Figure 6 shows the M, D and I properties computed as gray images as well as the (M->H, D->S, I->V) color image. Notice that the charge to the left which is further away appears more saturated.

Figure 9a shows in gray scale, the M property for the (τ=2, γ=5) image. Similarly, figures 9b and 9c show the D and I properties and figure 9d the (M->H, D->S, I->V) color image. In this image, if the appearance of a hue is unsaturated, i.e. appears gray, it indicates that it is further away. If it is darker, it is in a diffuse region.

Using the centroid, C property (figure 10a) for saturation is illustrated in figure 10b. This gives the effect of having a denser fog when the centroid is farther. Therefore a hue appears unsaturated due to the average densities, not just the peak, being further.

In both cases, the existence of any 'hot spot' or high value can be detected by the appearance of a region colored with a hue representing the value, or higher. Unfortunately, this implies that regions of 'low values' are hidden and more difficult to observe; an obvious result of mapping peak values.

The existence of any 'hot spot' or high value is detected by the appearance of a region colored with a hue representing that value, or higher. Unfortunately, this implies that regions of 'low values' are hidden and more difficult to observe; an obvious result of mapping peak values.

4.1 Incorporating Phase functions

To this point, the fact that the DE object is itself a distribution of backward scattering light sources meant that we could ignore the effects of the phase function $\Phi(s,\bar{s})$, the function characterizing the amount of scattering from the direction s to direction $\bar{s}$. Actually, any backward scattering phase function would be adequate for such a model. In a system of a large number of particles, such as we have assumed for the DE object, the phase function is not dependent on local properties of the medium. This is the isotropic case. The anisotropic case occurs when there is a preference in orientation as occurs in the case of Lambertian surfaces.

We diverge from the model of a physical system and assume that the DE object is anisotropic and has a phase function equal to the dot product between a preferred direction, the field gradient, and the direction of lighting. This provides a further attenuation in the brightness equation (3) of $j^{(s)}$ giving

$$B = \int_{t_1}^{t_2} e^{-\tau \int_{t_1}^{t} \rho(\lambda)\, d\lambda} \rho(t)\, j^{(s)}(x,y,z)\, dt \qquad (8).$$

We assume that $j^{(s)}$ is given by

$$j^{(s)}(x,y,z) = \sum_{i=1}^{n} \nabla\rho . \bar{L}_i ,$$

where there are n external light sources and L_i is the direction from the point (x,y,z) to the ith external light source.

Figure 11a is a gray scale rendering of the DE object shown in figure 9 using equation (8) with two light sources. The combined HSV image using the centroid image of figure 10a for Saturation is shown in figure 11b. The main difference between this figure and figure 9d is the darkening of the image in regions where the gradient points away from the light sources and there is a high density.

5. The ray tracer

The ray tracing algorithm is very straight forward. It consists of three steps, ray generation, computation, and display.

The ray generator fires rays from the eye through each screen pixel toward the scene.

When a ray intersects the extent of the 3D image it is stepped through the image extent and the four properties M, D, I and C are evaluated. The stepping algorithm is essentially identical to the one described by Snyder and Barr [19]. Since the centroid C requires the denominator line integral, this is also calculated and the final value for C is computed when the ray exits the image extent.

The four properties are stored as a temporary four dimensional image. When the entire image has been ray traced, an image may be created mapping functions of any three of the properties to the HSV components. As discussed in the previous section, the more intuitive combinations are the (M->H,D->S,I->V) and (M->H,C->S,I->V) mappings. The reason for storing the properties first is to establish a scale for the range of variation of each property. This allows the fullest usage of the HSV components.

Spatial coherence within the 3D image allows stepping the ray in the direction of increasing ray parameter. If there were a terminating criterion, such as minimum attenuation factor, the algorithm could stop the stepping before traversing the entire span of the extent. Unfortunately, with the exception of the intensity, all the properties have no such criterion. This is the price paid to be able to see the peak value behind even dim regions.

Even though there are no other types of rays fired, such as shadow or reflection rays, ray tracing was chosen as the rendering technique due to the simplicity in computing the attenuated integral in equation (3). Ray tracing matches well with the physical paradigm of light propagation and scattering in a non-homogeneous medium.

5.1 Results

The 3D images rendered in figures 5, 7, 8, 9, 10 and 11 have dimensions 64 x 64 x 400. The 256 x 256 images in figure 7 each took 450 seconds on a Sun 4 workstation to compute. The 512 x 512 images in the figures 9 and 10 took on the order of 2550 seconds to compute while figure 11 which took 4530 seconds. The 512 x 512 images in figures 2 and 3 took on the order of 76 seconds to compute, showing that thresholding, being a surface technique, is relatively fast.

The image generation phase takes on the order of 1 second on a Sun 3/160.

6. Conclusion

The technique presented for rendering a scalar field in color is by no means exhaustive. Color is added to the intensity computed in order to identify high valued regions, 'hot spots'. Even though the intensity calculation is a continuous generalization of the threshold rendering model, the full color image is superior since important hot spots cannot be occluded.

One obvious extension is the x-ray illumination model. In an x-ray, an attenuated line integral is computed similar to the brightness given in equation (3). This allows high density regions to be detected by the absence of light due to higher scattering.

Another extension is to go toward greater realism and perform shadowing and color scattering on non-density emitter models, i.e models that only scatter. This would allow the combination of backward and forward illumination. Discrete light sources could be placed within such a model to highlight regions of interest.

Another technique of visualization, which is important for three dimensional perception, is animation. As in particle systems, a fuzzy object 'comes alive' when animated.

7. Acknowledgements

The author thanks Schlumberger-Doll Research and in particular Stan Vestal, director of the systems science department, for providing the environment in which this work was possible. Indranil Chakravarty has always been there to discuss and help and has made suggestions in reviewing this paper.

8. References

[1] J. Blinn and M. Newell, "Texture and Reflection in Computer Generated Images," Comm. ACM, Oct. 1976, pp. 542-547.

[2] J. Blinn, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces",Computer Graphics 16(3), July 1982, pp. 21-29.

[3] L. Chen, G. Herman, R. Reynolds and J. Udupa, "Surface Shading in the Cuberille Environment", IEEE Computer Graphics and Applications 5(12), December 1985, pp.33-43.

[4] W. Dugan, "A terrain and cloud computer image generation model", Computer Graphics 13(2), 1979, pp.143-150.

[5] E. Farrell, "Color Display and Interactive Interpretation of Three-Dimensional Data", IBM J. Res. Develop 27(4), July 1983, pp.356-366.

[6] B. Fishman and B. Schacter, "Computer display of height fields", Computers and Graphics 5, 1980, pp.53-60.

[7] G. Frieder, D. Gordon, R. Reynolds, "Back-to-front Display of Voxel-based Objects", IEEE Computer Graphics and Applications, 5(1), Jan. 1985, pp.52-60.

[8] K. Hohne and R. Bernstein, "Shading 3D-Images from CT Using Gray-Level Gradients", IEEE Trans. on Medical Imaging MI-5, 1, March 1986, pp.45-47.

[9] J. Kajiya, B. Von Herzen, "Ray Tracing Volume Densities", Computer Graphics 18(3), July 1984, pp.165-174.

[10] R. Klassen, "Modeling the Effect of Atmosphere on Light", ACM Trans. on Graphics, 6(3), July 1987, pp.215-237.

[11] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", Computer Graphics 21(4), July 1987, pp. 163-169.

[12] X. Mao, T. Kunii, I. Fuhishiro and T. Noma, "Hierarchical Representations of 2D/3D Gray-Scale Images and Their 2D/3D Two-Way Conversion", IEEE Computer Graphics and Applications 7(12), December 1987 pp. 37-44.

[13] N. Max, "Light Diffuson through Clouds and Haze", Computer Vision, Graphics and Image Processing 33, 1986, pp.280-292.

[14] D. Meagher. "Geometric Modeling Using Octree Encoding" Computer Graphics and Image Processing 19(2), June 1982, pp 129-147.

[15] G. Meyer and D. Greenberg, "Perceptual Color Spaces for Computer Graphics", Computer Graphics 14, 1980, pp.247-261.

[16] W. Reeves, "Particle Systems- a technique for modeling a class of fuzzy objects", Computer Graphics 17(3), July 1983, pp.359-373.

[17] W. Reeves and R. Blan, "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems", Computer Graphics 19(3), July 1985, pp.313- 322 .

[18] P. Robertson and J. O'Callaghan, "The Application of Scene Synthesis Techniques to the Display of Multidimensional Image Data", ACM Trans. on Graphics , 4(4), Oct. 1985, pp.247-275.

[19] J. Snyder and A. Barr, "Ray Tracing Complex Models Containing Surface Tessellations", Computer Graphics 21(4), July 1987, pp.119-128.

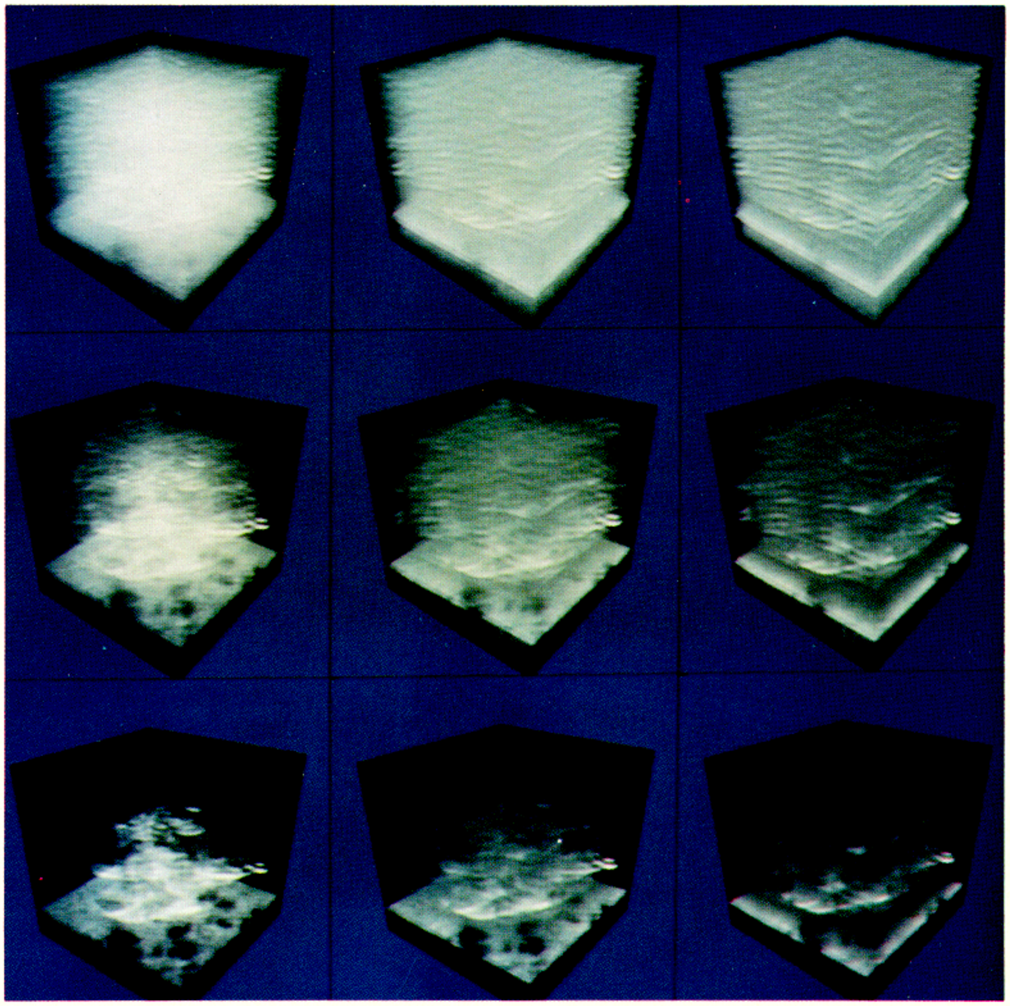

Figure 5, Intensity of the 3D image. $\tau = 1, 2, 3$ across the columns. $\gamma = 1, 5, 10$ down the rows.

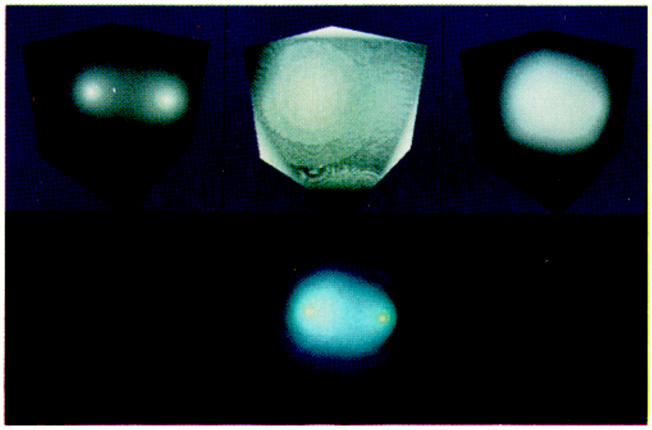

Figure 6, Rendering of peak (M), distance to peak (D), intensity (I) and the HSV mapping (M->H, D->S, I->V) for a simple field.

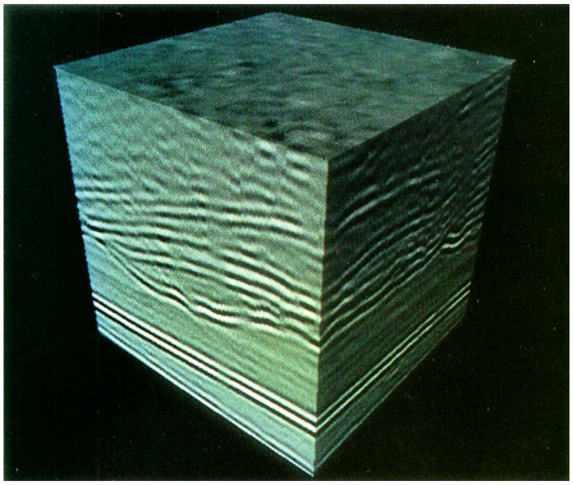

Figure 7. 3D image of sub-soil seismic data.

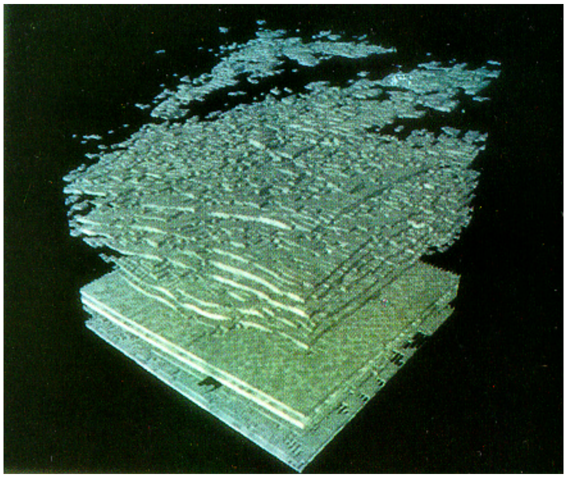

Figure 8. Thresholded view of seismic 3D image of figure 5.

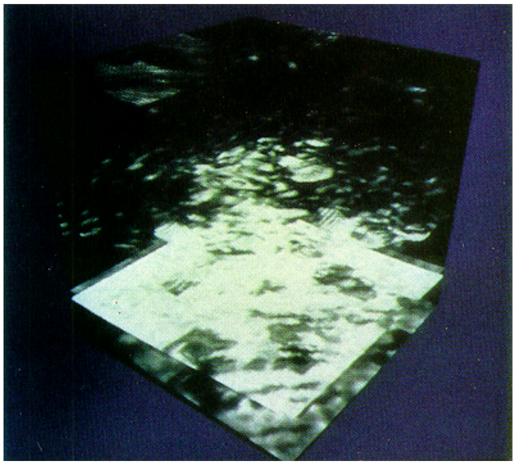

figure 9a. The peak values (M) for τ=2, γ=5.

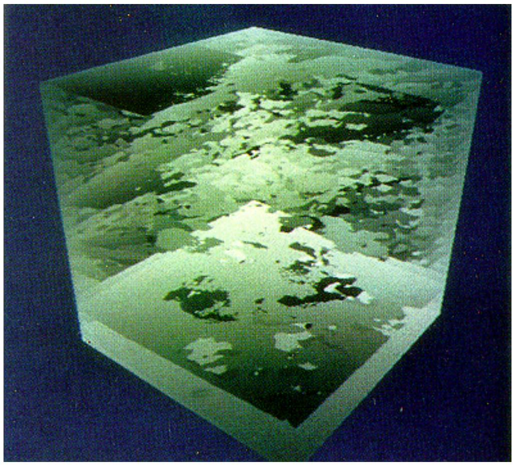

Figure 9b. Distance to the peak (D) for τ=2, γ=5.

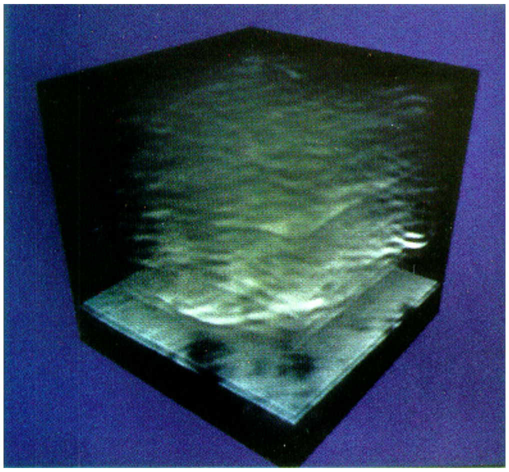

Figure 9c. The intensity (I) for τ=2, γ=5.

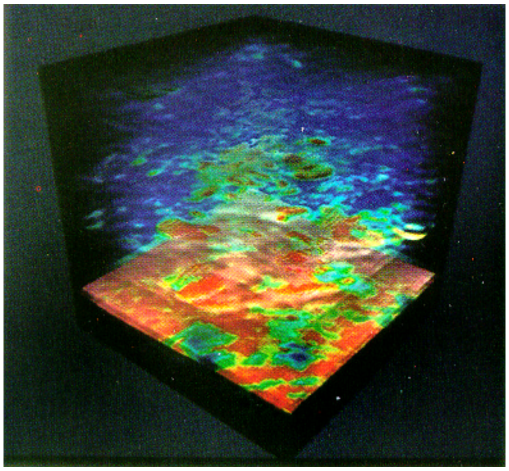

Figure 9d. The HSV mapping (M->H, D->S, I->V) combining figures 9a, 9b and 9c.

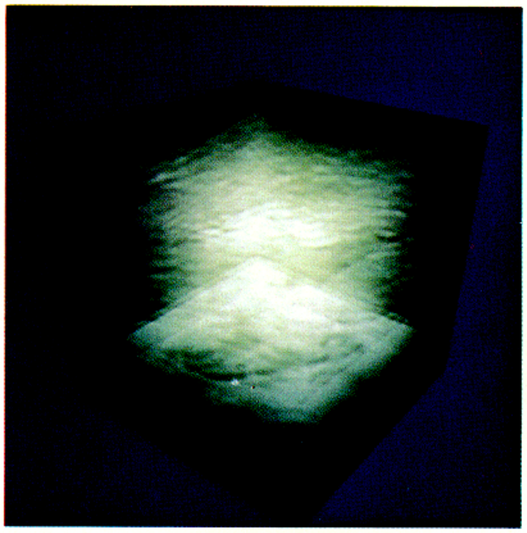

Figure 10a. the centroid (C).

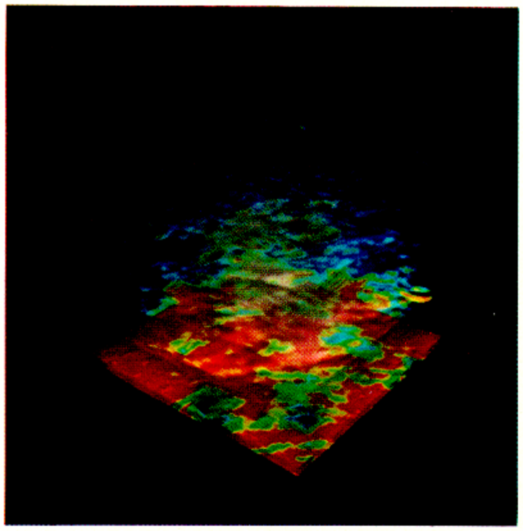

Figure 10b. The HSV mapping (M->H, C->S, I->V)
combining figures 9a, 10a, 9c.

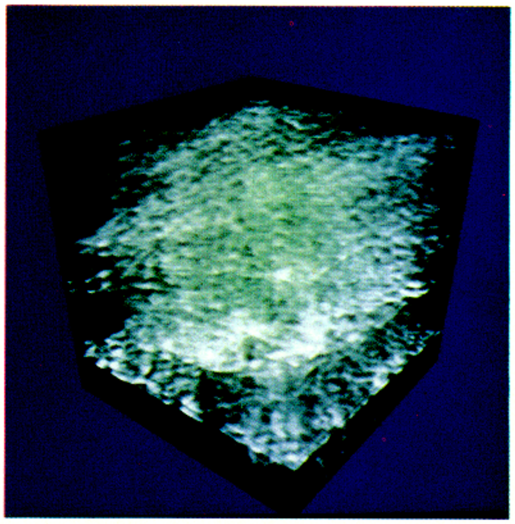

Figure 11a. the intensity (I)
using a phase function derived to highlight the gradient.

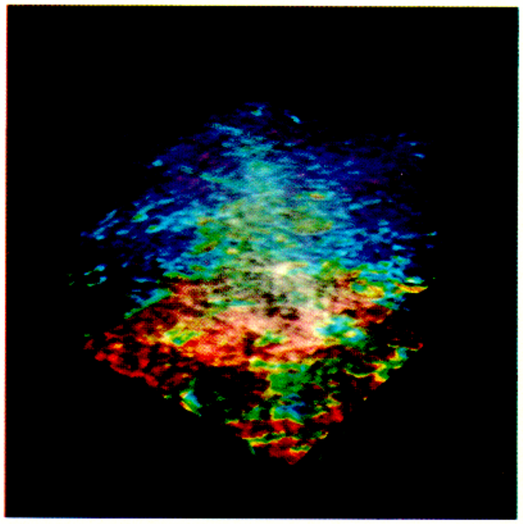

Figure 11b. The HSV mapping (M->H, C->S, I->V)
combining figures 9a, 10a, 11a.