Volume Rendering using the Fourier Projection-Slice Theorem

Marc Levoy

Computer Science Department
Center for Integrated Systems
Stanford University
Stanford, CA 94305-4070
Email: levoy@cs.stanford.edu

Abstract

The Fourier projection-slice theorem states that the inverse transform of a slice extracted from the frequency domain representation of a volume yields a projection of the volume in a direction perpendicular to the slice. This theorem allows the generation of attenuation-only renderings of volume data in $O(N^2 \log N)$ time for a volume of size N^3. In this paper, we show how more realistic renderings can be generated using a class of shading models whose terms are Fourier projections. Models are derived for rendering depth cueing by linear attenuation of variable energy emitters and for rendering directional shading by Lambertian reflection with hemispherical illumination. While the resulting images do not exhibit the occlusion that is characteristic of conventional volume rendering, they provide sufficient depth and shape cues to give a strong illusion that occlusion exists.

Keywords: Volume rendering, Fourier projections, Shading models, Scientific visualization, Medical imaging

1. Introduction

Volume rendering is a technique for visualizing a sampled scalar or vector volume of three spatial dimensions by modeling the propagation of light in a participating medium. In its most general form, parameters of the data are mapped to physical parameters of the medium such as energy density, absorption and scattering coefficients, and scattering phase function [Krueger90]. The integro-differential equation governing light transfer is then solved to obtain an energy equilibrium for the volume [Siegal81].

To reduce solution costs, practical volume rendering algorithms assume a low albedo, allowing them to ignore the effects of interreflection [Levoy88, Sabella88, Drebin88]. The resulting equation can be evaluated independently for each viewing ray using an expression of the form

$$I = \int_{-\infty}^{\infty} f(t) \exp\left[-\int_{-\infty}^{\infty} g(u)\, du\right] dt \tag{1.1}$$

where t represents distance along the ray. This expression is typically evaluated by digital compositing [Porter84], which in the context of volume rendering is equivalent to numerical integration using the rectangle rule. Given a cubical volume measuring N voxels on a side, and assuming pixel spacing equal to voxel spacing, the cost of generating an image using these algorithms is $O(N^3)$.

If we simplify the physical model still further by ignoring emission and first scatter, we produce an image that looks like an X-ray. The expression for each viewing ray now takes the form

$$I = \exp\left[-\int_{-\infty}^{\infty} g(t)\, dt\right]. \tag{1.2}$$

For an imaging geometry that employs uniformly spaced parallel rays, expressions of this form can be evaluated efficiently using the Fourier projection-slice theorem. This theorem allows us to compute integrals over volumes by extracting slices from a frequency domain representation of the volume. The application of this theorem to image synthesis has been independently proposed in [Dunne90] and [Malzbender]. An application to MR angiography is described in [Napel91].

The computational advantage of the projection-slice theorem is, at least in theory, considerable. Given a cubical volume and pixel spacing equal to voxel spacing as above, the cost per image is $O(N^2 \log N)$ after an $O(N^3 \log N)$ preprocessing step to compute the 3D frequency domain representation. For a volume and image of width $N = 256$, volume rendering using digital compositing requires $16M$ operations, while rendering using the projection-slice theorem requires only $512K$ operations, a speedup of more than an order of magnitude.

There are two disadvantages to this approach. From a practical standpoint, signal processing considerations have thus far prevented researchers from obtaining speedups matching the theoretical complexity. We expect these problems to be solvable, and, aside from a brief explanation in section 2, we do not address them further in this paper. A more fundamental disadvantage of the Fourier approach is that the resulting images do not exhibit occlusion and are consequently not as useful as volume renderings for many applications. Fortunately, occlusion is only one of many cues employed by the human visual system to determine the shape and spatial relationships of objects. Other cues include perspective, shading, texture, shadows, atmospheric attenuation, stereopsis, binocular convergence, ocular accommodation, head motion parallax, and the kinetic depth effect.

In this paper, we consider techniques for augmenting equation 1.2 to include as many depth and shape cues as possible while retaining the computational efficiency of evaluation using the projection-slice theorem. The key restriction placed on us is that, to avoid recomputing the 3D frequency domain representation for each view, the integrand must be independent of viewing direction. If we wish to support rotation of the volume relative to a light source, the integrand must also be independent of lighting direction. A class of shading models that satisfies these constraints has the form for each viewing ray

$$I = \sum_{i=0}^{n} w_i \left[\int_{-\infty}^{\infty} f_i(t)\, dt \right] \tag{1.3}$$

for scalars w_i and scalar volumes f_i. As we shall see, a surprisingly large class of shading models are expressible as linear combinations of this form where the f_i's are independent of both viewing and lighting directions. In particular, we consider the following three models:

- X-rays with depth cueing
- X-rays with directional shading
- X-rays with depth cueing & directional shading

The remainder of the paper is organized as follows. Section 2 describes the Fourier projection-slice theorem and gives an algorithm for using it to generate arbitrarily oriented X-rays of a sampled scalar volume. In section 3, we derive formulations for the three shading models listed above plus a few extensions. Section 4 describes our implementation and results, and section 5 gives conclusions and suggests directions for future research.

2. The projection-slice theorem

The Fourier projection-slice theorem states that the inverse 2D Fourier transform of a slice passing through the origin of the 3D Fourier transform of a function gives a projected image of the function, where the projection is in a direction perpendicular to the extracted slice. Each point on the projection is the integral of the function over the infinite ray passing through that point and normal to the slice [Dudgeon84]. For later use, we define a Fourier projection operator Π_V with input function f on $\mathbf{R}^3$ and output image I on $\mathbf{R}^2$ such that

$$I = \Pi_V(f) = \mathcal{F}_2^{-1} \{ \mathcal{F}_3 \{ f \} \delta_V \}. \tag{2.1}$$

δ_V restricts the spectrum to a plane passing through the origin and perpendicular to viewing direction $\mathbf{V}$, and $\mathcal{F}_N$ and $\mathcal{F}_N^{-1}$ are the forward and inverse N-dimensional Fourier transforms, respectively.

The Fourier projection-slice theorem is the basis for one variant of computed tomography (CT). Specifically, if the 2D Fourier transforms of a set of X-rays of an object taken at different angles are inserted into a 3D spectrum such that each slice passes through the origin and is perpendicular to the (parallel) X-ray beam used to acquire that slice, the inverse 3D Fourier transform of the composite spectrum yields a 3D radiometric facsimile of the object, i.e. a CT dataset [Macovski83].

Figure 1: Algorithm for generating an attenuation-only rendering of a volume dataset using the Fourier projection-slice theorem.

By reversing the CT acquisition process, we can generate X-rays from CT data. A typical implementation is shown in figure 1. In a preprocessing step, we compute the 3D discrete Fourier transform (DFT) of a scalar volume sampled on the three-dimensional integer lattice. For real volumes (i.e. having no imaginary component), a discrete Hartley transform (DHT) [Bracewell86] may be substituted. For each view, we extract a slice passing through the origin and compute its inverse 3D DFT. Since an arbitrarily oriented slicing plane does not typically pass through points on the integer lattice, slice values must be computed by interpolation from nearby samples.

For a volume measuring N voxels on a side where N is an integer power of two, the computation of a 3D spectrum using a Fast Fourier Transform (FFT) or Fast Hartley Transform (FHT) algorithm requires $O(N^3 \log N)$ operations. The cost of extracting a 2D slice measuring N samples on a side from the 3D spectrum is $O(W^3 N^2)$ where W is the nonzero width of the interpolating filter. To complete the algorithm, we perform a 2D inverse FFT or FHT, which requires $O(N^2 \log N)$ operations. Properly speaking, sampling theory demands that we employ 2*x* oversampling during slice extraction. We have found that the penalty of not doing so is minor for the datasets we typically encounter. Assuming $N = 256$ and $W = 4$, we have slice extraction and inverse transformation costs of 16M and 2.25M operations, respectively. Therefore, although the Fourier transformation is asymptotically dominant, the slice extraction step usually takes more time.

The data structures required by the algorithm (in addition to the input volume) include a 3D spectrum measuring N

Figure 2: Imperfect reconstruction in the frequency domain produces incompletely suppressed replicas in the spatial domain. These appear in the image as noticeable ghosting.

coefficients on a side and a 2D slice that also measures N coefficients on a side. While the input volume may be adequately represented by 8 bits per voxel, accurate frequency domain representations typically require 32-bit floating point coefficients, thereby increasing memory requirements by a factor of 4. Quantization or coding of Fourier coefficients such as specified in the JPEG still picture compression standard is possible, but reconstruction time must be considered.

The use of finite support interpolation filters in the slice extraction step can give rise to objectionable artifacts in Fourier projections. Although these problems are not the focus of this paper, they impact performance significantly, so a brief discussion of them is warranted.

Convolution of a sampled function with a filter of finite support causes incomplete suppression of periodic replicas in the other domain. Usually, convolution is performed in the spatial domain and the incompletely suppressed replicas occur in the frequency domain. In the Fourier projection algorithm, we convolve in the frequency domain, producing incompletely suppressed replicas in the spatial domain that appear as noticeable ghosting. The problem is summarized in one lower dimension in figure 2. Two solutions are known for this problem:

(1) Employ a better (and typically wider) interpolation filter. If we increase the nonzero filter width by a factor of k_1, $k_1 > 1$, we increase per-view processing time by a factor of k_1^3.

(2) Widen the separation between spatial domain replicas by zero padding the input volume. If we increase the volume width by a factor of k_2, which corresponds to surrounding the data on each of its six sides by a margin of zeros of width $N(k_2 - 1)/2$, we increase per-view processing time by a factor of k_2^2. We also increase memory requirements by a factor of k_2^3 and preprocessing time by a factor of $k_2^3\,(log\ k_2 + \log N)\,/\log N$, but these effects are less important.

The second solution has theoretically superior per-view performance. The relative impact of these two solutions on image quality has not been studied, however. Malzbender reports satisfactory results with a 5^3 tap filter and a 10% margin of zeros on each side of the data [Malzbender]. The images in this paper were made using a 16% margin ($k_2 = 4/3$) and a 4^3 tap filter.

3. Shading models

As discussed in the introduction, the class of integrals that can be directly evaluated using the Fourier projection-slice theorem does not allow the modeling of emission and scattering in a participating medium. In this section, we consider several shading models that are linear combinations of Fourier projections in the spirit of equation 1.3 and which restore some of the lost visual cues.

3.1. Depth cueing

The fraction of light lost to absorption or scattering in a participating medium is exponentially related to the distance it travels. This attenuation provides strong perceptual cues concerning the relative depth of objects in the scene. In this section, we develop an approximate shading model in which exponential attenuation with locally varying absorption and scattering coefficients is replaced by linear attenuation with a constant coefficient. It will be shown that images based on this model can be rendered efficiently using the projection-slice theorem.

Let us begin with exponential attenuation. We define a volume of density $\rho(x,y,z)$ and color $C(x,y,z)$ to emit light with an energy of $C\rho$ per unit length and absorb light with an opacity of $\tau\,\rho$ per unit length where τ is a constant. We also define a viewing ray $\mathbf{P}(t)$ parameterized by length t and two points $\mathbf{P}(a)$ and $\mathbf{P}(b)$ lying on the ray. Ignoring scattering, the total energy I_P arriving at $\mathbf{P}(b)$ due to emission and absorption along that portion of the ray lying between $\mathbf{P}(a)$ and $\mathbf{P}(b)$ is given by

$$I_P = \int_a^b \rho\,(\mathbf{P}(t))\; C\,(\mathbf{P}(t))\, \exp\Big[-\int_a^t \tau\,\rho\,(\mathbf{P}(u))\, du \Big]\, dt. \qquad (3.1.1)$$

The notation is adapted from [Max90]. Replacing exponential attenuation with locally varying coefficients $\tau\,\rho$ by linear attenuation with constant coefficient τ gives

$$I_P = \int_a^b \rho\,(\mathbf{P}(t))\; C\,(\mathbf{P}(t))\, [\tau/2 + \tau/2\, (\mathbf{D}\,(\mathbf{P}(t)) \cdot \mathbf{V})]\, dt \qquad (3.1.2)$$

where $\mathbf{D} = (D_x, D_y, D_z)$ is a linear function of position along the ray, i.e. $\mathbf{D} = m\,\mathbf{P} + b$ for constants m and b, and $\mathbf{V} = (V_x, V_y, V_z)$ is the normalized viewing direction as shown in figure 3. Note that $\mathbf{V}$ points away from the eye, not towards it as is common in many textbooks. We have chosen this convention in order to underscore the similarity between equation 3.1.2 and equation 3.2.3 in section 3.2.

In the introduction, we stated that in order to avoid recomputing the 3D frequency domain representation for each

Figure 3: Linear depth cueing for a volume centered at the origin is given by $\mathbf{D} = m\,\mathbf{P} + b$ for constants m and b where $\mathbf{P}(t)$ is position along a viewing ray and $\mathbf{V}$ is the normalized viewing direction.

view, all integrals over the volume must be independent of viewing direction. Looking back at the last equation, we observe that viewing direction $\mathbf{V}$ is independent of ray parameter t. We may therefore factor it out of the integral, producing

$$\begin{aligned} I_P = {} & V_x \int_a^b \rho\,(\mathbf{P}(t))\, C(\mathbf{P}(t))\, [\tau/2 + \tau/2\,(D_x(\mathbf{P}(t)))]\, dt \\ & + V_y \int_a^b \rho\,(\mathbf{P}(t))\, C(\mathbf{P}(t))\, [\tau/2 + \tau/2\,(D_y(\mathbf{P}(t)))]\, dt \\ & + V_z \int_a^b \rho\,(\mathbf{P}(t))\, C(\mathbf{P}(t))\, [\tau/2 + \tau/2\,(D_z(\mathbf{P}(t)))]\, dt \\ & + c_1 \int_a^b \rho\,(\mathbf{P}(t))\, C(\mathbf{P}(t))\, dt. \end{aligned} \tag{3.1.3}$$

where $c_1 = \tau/2 - \tau/2\,(V_x + V_y + V_z)$.

These four integrals are now in a form that can evaluated using the Fourier projection-slice theorem. Applying the projection operator defined in equation 2.1, we obtain the following expression for output image I

$$\begin{aligned} I = {} & V_x\, \Pi_V[(\tau/2 + \tau/2\, D_x(x,y,z))\, \rho\,(x,y,z)\, C(x,y,z)] \\ & + V_y\, \Pi_V[(\tau/2 + \tau/2\, D_y(x,y,z))\, \rho\,(x,y,z)\, C(x,y,z)] \\ & + V_z\, \Pi_V[(\tau/2 + \tau/2\, D_z(x,y,z))\, \rho\,(x,y,z)\, C(x,y,z)] \\ & + c_1\, \Pi_V[\rho\,(x,y,z)\, C(x,y,z)]. \end{aligned} \tag{3.1.4}$$

The algorithm is shown in figure 4. Four copies of $\rho\,(x,y,z)\, C(x,y,z)$ are created. Each is multiplied by the appropriate depth cueing function and Fourier transformed, yielding four 3D spectra. The first three spectra represent volumes that are depth cued along one of object space

Figure 4: Algorithm for generating a depth cued rendering of a volume dataset. During precomputation, volume $\rho\, C$ is multiplied by depth operator $\mathbf{D}$. For each view, the extracted slices are multiplied by viewing direction $\mathbf{V}$.

coordinate axes. The fourth spectrum represents the original volume. For each view, slices are extracted from each spectra, inverse Fourier transformed, weighted by the appropriate component of the current viewing direction as shown in the figure and summed, producing an image that is depth cued for that viewing direction.

To summarize, we have replaced per-voxel shading followed by projection with projection followed by per-pixel shading. This transformation is allowed because integration is a linear operator and the current viewing direction is constant for all voxels. Both formulations produce identical results, but the latter allows a more efficient implementation using Fourier projections.

3.2. Directional shading

In the previous section, we showed that a per-voxel dot product followed by projection could be replaced with projection followed by a per-pixel dot product. There are other places where dot products appear in per-voxel shading models. Most notable is probably the $|\mathbf{N} \cdot \mathbf{L}|$ product giving the irradiance on a surface of orientation $\mathbf{N}$ produced by a parallel light source originating from direction $\mathbf{L}$. Unfortunately, the presence of the nonlinear absolute value operator prevents evaluation of this dot product using the Fourier projection-slice theorem. In this section, we develop an alternative, physically valid, shading model in which illumination from a parallel light source is replaced by illumination from a hemispherical source. It will be shown that images based on this model can be rendered efficiently using the projection-slice theorem.

Let us begin with Lambertian reflection and parallel illumination. We define a volume of albedo $\Omega(x,y,z)$ and an orientation field $\mathbf{N}(x,y,z)$. The latter is typically estimated

Figure 5: Hemispherical illumination of a surface. Irradiance is equal to integral of the projection of shaded portion of the hemisphere down onto the plane containing the surface.

from the scalar input data using a central difference operator [Levoy88]. Ignoring attenuation, the energy I_P arriving at point $\mathbf{P}(b)$ along viewing ray $\mathbf{P}(t)$ is given by

$$I_P = \int_a^b I_s\, \Omega\, (\mathbf{P}(t))\ |\mathbf{N}\,(\mathbf{P}(t)) \cdot \mathbf{L}|\ dt \qquad (3.2.1)$$

where I_s is the intensity of a parallel light source originating from direction $\mathbf{L}$. This expression assumes illumination in both the $\mathbf{L}$ and $-\mathbf{L}$ directions. One-side illumination is obtained by replacing $|\mathbf{N} \cdot \mathbf{L}|$ with $\max(\mathbf{N} \cdot \mathbf{L},\ 0)$.

Since the absolute value (or max) operator is nonlinear, we cannot factor lighting direction $\mathbf{L}$ out of the integral as we did with viewing direction $\mathbf{V}$ in section 3.1. Omitting these operators entirely would cause voxels containing backward-facing normals (relative to the light source) to produce negative light that would combine under integration with positive light produced by voxels containing forward-facing normals to yield incorrect images.

To eliminate this problem, we consider an alternative illumination method - an emissive hemisphere. The irradiance E_i on a surface having normal vector $\mathbf{N}$ illuminated by a hemisphere whose pole points in direction $\mathbf{L}$ as shown in figure 5 is equal by Nusselt's analogue (as described in [Cohen85]) to the projection of the visible portion of the hemisphere down onto the plane containing the surface, or

$$\begin{aligned} E_i &= I_s \int_{-\frac{\pi}{2}+\theta}^{\frac{\pi}{2}} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos\phi \cos^2\psi\, d\psi\, d\phi \\ &= \pi/2 + \pi/2 \cos\psi \\ &= \pi/2 + \pi/2\, (\mathbf{N} \cdot \mathbf{L}). \end{aligned} \qquad (3.2.2)$$

Assuming Lambertian reflection with albedo $\Omega\,(x,y,z)$, we

Figure 6: Algorithm for generating a directionally shaded rendering of a volume dataset. During precomputation, volume $I_s\, \Omega$ is multiplied by orientation field $\mathbf{N}$. For each view, the extracted slices are multiplied by lighting direction $\mathbf{L}$.

have

$$I_P = \int_a^b I_s\, \Omega\, (\mathbf{P}(t))\ [\pi/2 + \pi/2\, (\mathbf{N}\,(\mathbf{P}(t)) \cdot \mathbf{L})]\ dt. \qquad (3.2.3)$$

Although seldom seen in the graphics literature (a notable exception is [Nishita86]), this shading model is entirely plausible - it models the appearance of diffuse objects outdoors on a cloudy day (ignoring interreflection among objects on the ground). The directional shading provided by this model is less sensitive to surface orientation than shading arising from parallel illumination, just as our perception of shape is poorer on a cloudy day than on a sunny day. Nevertheless, the model is a significant improvement over ignoring scattering effects entirely.

Continuing our derivation, we observe that equation 3.2.3 is nearly identical to equation 3.1.2. If we wish to support rotation of the volume relative to the light source yet avoid recomputing the 3D frequency domain representation for each view, the integrand must be independent of lighting direction. Looking at the equation, we see that hemisphere pole direction $\mathbf{L}$ is independent of ray parameter t and may be factored out of the integral. As in the case of depth cueing, this manipulation produces four integrals that can each be evaluated using the Fourier projection-slice theorem. Repeating the steps in section 3.1, we obtain the following expression for output image I

$$\begin{aligned} I \ &= \ L_x\, \Pi_V[(\pi/2 + \pi/2\, N_x(x,y,z))\, I_s\, \Omega\,(x,y,z)] \\ &+\ L_y\, \Pi_V[(\pi/2 + \pi/2\, N_y(x,y,z))\, I_s\, \Omega\,(x,y,z)] \\ &+\ L_z\, \Pi_V[(\pi/2 + \pi/2\, N_z(x,y,z))\, I_s\, \Omega\,(x,y,z)] \\ &+\ c_2\, \Pi_V[I_s\, \Omega\,(x,y,z)] \end{aligned} \qquad (3.2.4)$$

where $N_x(x,y,z)$ is the X-component of the orientation field at location (x,y,z) similarly for N_y and N_z, and $c_2 = \pi/2 - \pi/2\,(L_x+L_y+L_z)$.

The algorithm is shown in figure 6. Four copies of $I_s\,\Omega\,(x,y,z)$ are created. Each is multiplied by the appropriate component of $\mathbf{N}\,(x,y,z)$ and Fourier transformed, yielding four 3D spectra. Because of the method used to compute **N**, the first three spectra essentially represent volumes that have been gradient-enhanced in one of the object space coordinate directions. The fourth spectra represents the original volume. For each change in view or lighting direction, slices are extracted from each spectra, inverse Fourier transformed, weighted by the appropriate component of the current lighting direction as shown in the figure and summed, producing an image that is directionally shaded for that lighting direction.

As in the case of depth cueing, we have replaced per-voxel shading followed by projection with projection followed by per-pixel shading because one of the terms in a dot product - the current lighting direction - is constant for all voxels.

3.3. Depth cueing & directional shading

If depth cueing and directional shading as defined in sections 3.1 and 3.2 can each be expressed as linear combinations of Fourier projections, it should be possible to combine them, producing a composite shading model that exhibits both effects and which is expressible as a linear combination of Fourier projections.

Multiplying the integrand in equation 3.2.3 by the linear attenuation factor in equation 3.1.2, we obtain for a viewing ray

$$I_P = \int_a^b I_s\,\Omega\,(\mathbf{P}(t))\,[\pi/2 + \pi/2\,(\mathbf{N}\,(\mathbf{P}(t)\cdot\mathbf{L})]$$
$$\times\;[\tau/2 + \tau/2\,(\mathbf{D}\,(\mathbf{P}(t))\cdot\mathbf{V})]\,dt. \qquad (3.3.1)$$

After we factor and apply the Fourier projection operator, we obtain for the output image

$$I = \tau\,\pi/2\,\mathbf{V_L}\cdot\Pi_V[(1/2 + 1/2\,\mathbf{D_N}\,(x,y,z))\,I_s\,\Omega\,(x,y,z)]$$
$$+\;\tau\,\pi/2\,c_3\,\Pi_V[I_s\,\Omega\,(x,y,z)] \qquad (3.3.2)$$

for $c_3 = 1/2 - 1/2 \sum_i \mathbf{V}_{\mathbf{L}_i}$ where

$$\mathbf{V_L} = (V_xL_x,\, V_xL_y,\, V_xL_z,\, V_yL_x,\, V_yL_y,\, V_yL_z,$$
$$V_zL_x,\, V_zL_y,\, V_zL_z, V_x,\, V_y,\, V_z,\, L_x,\, L_y,\, L_z)$$

and

$$\mathbf{D_N} = (D_xN_x,\, D_xN_y,\, D_xN_z,\, D_yN_x,\, D_yN_y,\, D_yN_z,$$
$$D_zN_x,\, D_zN_y,\, D_zN_z,\, D_x,\, D_y,\, D_z,\, N_x,\, N_y,\, N_z).$$

This requires the extraction and inverse Fourier transformation of 16 slices from 16 precomputed spectra. While this algorithm is still asymptotically less expensive than conventional volume rendering, its high constant makes it slower except for very large volumes. We note that our composite shading model ignores emission $\rho\,C$ in equation 3.1.2. In the context of a directional shading model, this term can be thought of as representing ambient illumination. If included, it adds another 3 terms to equation 3.3.2 and hence another 3 slices that must be extracted.

3.4. Specular reflections and other extensions

Shiny surfaces exhibit a greater change in reflected light intensity for small variations in surface orientation than dull surfaces. A common formulation of the specular term in Phong's illumination model [Phong75] is $|\mathbf{N}\cdot\mathbf{H}|^n$ for some exponent n where **H** is the vector halfway between the directions of the light source, **L**, and the viewer, **V**. Using hemispherical illumination to avoid the absolute value operator, we have for a viewing ray

$$I_P = \int_a^b I_s\,\Omega\,(\mathbf{P}(t))\,[\pi/2 + \pi/2\,(\mathbf{N}\,(\mathbf{P}(t)\cdot\mathbf{H})^n]\,dt. \qquad (3.4.1)$$

After we factor and apply the Fourier projection operator, we obtain for the output image

$$I = \mathbf{H_n}\cdot\Pi_V[(\pi/2 + \pi/2\,\mathbf{N_n}(x,y,z))\,I_s\,\Omega\,(x,y,z)]$$
$$+\;c_4\,\Pi_V[I_s\,\Omega\,(x,y,z)] \qquad (3.4.2)$$

for suitable constant c_4 where $\mathbf{H_n}$ and $\mathbf{N_n}$ contain the list of factors in the polynomial $(\mathbf{N}\cdot\mathbf{H})^n$. The number of terms and hence the number of spectra is given by the multinomial theorem [Knuth73]. For $n = 10$, a typical value, we require 286 precomputed spectra, making this extension slower than conventional volume rendering even for very large volumes.

For scenes of high depth complexity, linear attenuation may provide an insufficiently steep intensity falloff to disambiguate depth relationships. Equation 3.4.1 can be adapted to provide n-degree polynomial depth cueing by replacing **N** and **H** with **D** and **V**, respectively, and adjusting the constants. For example, inverse square law falloff would be given by $(\mathbf{D}\cdot\mathbf{V})^2$ and would require 10 slices from 10 precomputed spectra.

4. Implementation and results

Figures 7 through 13 each show the application of a different shading model to a 96^3 voxel cube extracted from a CT scan of a human skull mounted in a lucite head cast. The right eye orbit and a portion of the zygoma (temple) are visible at center and left, respectively, and the lucite nose is visible at lower right. To facilitate comparisons, figure 7 is a conventional volume rendering generated using the techniques described in [Levoy88].

Figures 8 through 13 were generated using the Fourier projection-slice theorem as follows. The input data was first edge enhanced and nonlinear windowed (except for figure 8 as noted below), then surrounded with a 16% margin of zeros to avoid the ghosting problem discussed in section 2. The resulting 128^3 voxel cube was weighted according to equations 3.1.4, 3.2.4, or 3.3.2, and Fourier transformed using a 3D FFT algorithm. For each view, slices of width equal to the spectra were extracted using a tricubic interpolating spline, inverse transformed using a 2D FFT algorithm, weighted appropriately for the current viewing or lighting directions, and summed to produce a 128^2 image.

Memory requirements varied from 48 megabytes for equation 3.1.4 to 256 megabytes for equation 3.3.2. Per-view rendering time varied from 6 seconds for equation 3.1.4 (about 1.5 seconds per extracted slice) to over a minute for equation 3.3.2 (due mostly to paging). Timings are for a 36 MHz Silicon Graphics 4D/320 VGX and a sequential C implementation. No attempt was made to optimize the code.

By replacing the FFT with an FHT, one can immediately halve the per-view rendering time and the memory requirements. By optimizing the filter and trimming the zero padding, another factor of two speedup and some reduction in memory usage can be expected. For greater speedups, the algorithm can be parallelized at either a fine grain (within the FFT) or a coarse grain (between slices). It is also amenable to hardware acceleration using video coprocessors or video digital signal processors (DSP). These devices frequently operate on fixed-point representations of frequency spectra, suggesting additional opportunities for reducing memory costs.

Figure 8 shows an attenuation-only projection (i.e. an X-ray) generated using a variant of equation 1.2. The lack of occlusion and shape-from-shading cues is evident. Figure 9 shows an attenuation-only projection of the input volume after edge sharpening and nonlinear windowing of the density range. Figures 10 through 13 also include sharpening and windowing. Figure 10 shows a depth-cued X-ray generated using the shading model described in section 3.1. The weighting assigned to each point in the volume falls off linearly with increasing distance from the viewer. Figure 11 shows a directionally shaded X-ray generated using the shading model described in section 3.2. Each point in the volume is weighted as if it sat on a diffusely reflecting surface illuminated by a hemisphere placed on the right side of the picture and shining leftwards. Figures 12 and 13 show two views that each include both depth cueing and directional shading using the model described in section 3.3. Although no occlusion occurs in either of these figures, the shading effects enhance our ability to perceive shape and discern spatial relationships. When displayed in motion, the visual impact is sufficiently strong that viewers often overlook the fact that these are X-rays, not opaque renderings.

5. Conclusions and future work

We have described a family of algorithms for generating realistically shaded renderings of volume data using the Fourier projection-slice theorem. While the images produced using these algorithms do not exhibit occlusion, they provide sufficient depth and shape information to make them moderately understandable in static views and very understandable in motion sequences. In either case, they produce images richer in visual cues than the attenuation-only projections with which the projection-slice theorem is usually associated.

One aspect of the Fourier approach that we have not discussed is how it impacts interactive data segmentation. The edge sharpening and nonlinear windowing operators used in figures 9 through 13 are applied before the forward 3D Fourier transform. As a consequence, changing them requires recomputing the spectra. Depending on the application, it may be undesirable to hardwire data segmentation in this way.

Several strategies are available for avoiding this difficulty. If the data segmentation operator is expressible as a linear combination of basis functions, it can be handled using the same scheme we have employed for shading models. For example, if a CT dataset is segmented into fat, muscle, and bone volumes during preprocessing, yielding three spectra, the weights and colors assigned to each material can be altered at image generation time by applying different weights to the slices extracted from each spectrum.

If the segmentation operator is expressible as a convolution, it may be possible to implement it equally or more efficiently as multiplication in the frequency domain. For example, the edge sharpening operator used in figures 9 through 13 could have been implemented as multiplication of each extracted slice by a high-frequency emphasis function. Using this approach, the amount and type of enhancement can be specified at image generation time.

Conversely, if the segmentation operator is expressible as multiplication by a function of position in the spatial domain, it may be possible to implement it as convolution in the frequency domain. For an intriguing example of this strategy, consider the linear depth cueing described in section 3.1. Although the Fourier transform of this function is infinite in extent, equally effective depth cueing can be obtained using the first half period of a suitably oriented cosine function. The Fourier transform of a cosine is two delta functions. For each view, we determine the placement of these deltas, preconvolve them with our interpolation filter (which amounts to summing two shifted copies of the filter function), and use the composite filter during slice extraction. This technique requires only one spectrum as input, making it potentially more efficient than the technique described in this paper. We are currently investigating this approach.

Looking beyond shading models, sums of Fourier projections can also be used to compute integrals over irregular spatial domains. Specifically, the integral of any function over the ray segment connecting two points in the interior of a volume can be computed as a sum of integrals over spatially disjoint intervals of the ray that taken together span the segment. Assuming well behaved functions, each integral can be computed using the projection-slice theorem. We are currently investigating an algorithm based on recursive subdivision of volumes into octants that can compute images in $O(N^2 \log^2 N)$ time. Possible applications include rendering of spatially clipped volumes for medical visualization, fast calculation of inter-element attenuation for zonal radiosity algorithms, and approximate visibility determination for geometrically defined scenes.

Acknowledgements

This research was supported by the National Science Foundation, (NSF), the National Aeronautics and Space Administration (NASA), and the sponsoring companies of the Stanford Center for Integrated Systems (CIS). The SGI 4D/320 VGX workstation was donated by Silicon Graphics, Inc. The CT scan was provided by North Carolina Memorial Hospital. Discussions with Adam Levinthal were useful in the early stages of this project, Takashi Totsuka suggested the frequency domain implementation of depth cueing, and Alice Yu provided a helpful reading of the manuscript.

References

[Bracewell86] Bracewell, R., *The Fourier Transform and Its Applications*, McGraw-Hill, 1986.

[Cohen85] Cohen, M.F. and Greenberg, D.P., ''The Hemi-Cube: A Radiosity Solution for Complex Environments,'' *Computer Graphics (Proc. SIGGRAPH '85)*, Vol. 19, No. 3, July, 1985, pp. 31-40.

[Drebin88] Drebin, R.A., Carpenter, L., and Hanrahan, P., ''Volume Rendering,'' *Computer Graphics (Proc. SIGGRAPH '88)*, Vol. 22, No. 4, August, 1988, pp. 65-74.

[Dudgeon84] Dudgeon, D.E. and Mersereau, R.M., *Multidimensional Digital Signal Processing*, Prentice-Hall, 1984.

[Dunne90] Dunne, S., Napel, S. and Rutt, B., ''Fast Reprojection of Volume Data,'' *Proc. First Conference on Visualization in Biomedical Computing,* IEEE Computer Society Press, May, 1990, pp. 11-18.

[Hottel67] Hottel, H.C. and Sarofim, A.F., *Radiative Transfer*, McGraw-Hill, 1967.

[Knuth73] Knuth, D., *The Art of Computer Programming*, Addison-Wesley, 1973.

[Krueger90] Krueger, W., ''Volume Rendering and Data Feature Enhancement,'' *Computer Graphics (Proc. San Diego Workshop on Volume Visualization)*, Vol 24, No. 5, November, 1990, pp. 21-26.

[Levoy88] Levoy, M., ''Display of Surfaces from Volume Data,'' *IEEE Computer Graphics and Applications*, Vol. 8, No. 3, May, 1988, pp. 29-37.

[Macovski83] Macovski, A., *Medical Imaging Systems*, Prentice-Hall, 1983.

[Malzbender] Malzbender, T., ''Fourier Volume Rendering.'' Submitted for publication.

[Max90] Max, N., Hanrahan, P. and Crawfis, R., ''Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions,'' *Computer Graphics (Proc. San Diego Workshop on Volume Visualization)*, Vol. 24, No. 5, November, 1990, pp. 27-33.

[Napel91] Napel, S., Dunne, S. and Rutt, B.K., ''Fast Fourier Projection for MR Angiography,'' *Magnetic Resonance in Medicine*, Vol. 19, 1991, pp. 393-405.

[Nishita86] Nishita, T. and Nakamae, E., ''Continuous Tone Representation of Three-Dimensional Objects Illuminated by Sky Light,'' *Computer Graphics (Proc. SIGGRAPH '86)*, Vol. 20, No. 4, August, 1986, pp. 125-132.

[Phong75] Bui-Tuong, Phong, ''Illumination for Computer-Generated Pictures,'' *Communications of the ACM.* Vol. 18, No. 6, June, 1975, pp. 311-317.

[Porter84] Porter, T. and Duff, T., ''Compositing Digital Images,'' *Computer Graphics (Proc. SIGGRAPH '84)*, Vol. 18, No. 3, July, 1984, pp. 253-259.

[Rushmeier87] Rushmeier, H.E. and Torrance, K.E., ''The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium,'' *Computer Graphics (Proc. SIGGRAPH '87)*, Vol. 21, No. 4, July, 1987, pp. 293-302.

[Sabella88] Sabella, P., ''A Rendering Algorithm for Visualizing 3D Scalar Fields,'' *Computer Graphics (Proc. SIGGRAPH '88)*, Vol. 22, No. 4, August 1988, pp. 51-58.

[Siegel81] Siegel, R. and Howell, J.R., *Thermal Radiation Heat Transfer*, Hemisphere Publishing, 1981.

Figure 7: Conventional volume rendering of a CT scan of a human skull mounted in a lucite head cast.

Figure 8: Attenuation-only projection (i.e. X-ray). All figures on this page were generated using the Fourier projection-slice theorem.

Figure 9: Attenuation-only projection after edge sharpening and nonlinear windowing. Figures 10-13 also include these enhancements.

Figure 10: Depth-cued X-ray generated using the shading model described in section 3.1.

Figure 11: Directionally shaded X-ray generated using the shading model described in section 3.2.

Figure 12: Depth-cued and directionally shaded X-ray generated using the shading model described in section 3.3.

Figure 13: Rotated view of depth-cued and directionally shaded X-ray.