Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institute for Information Systems

Computer Graphics Research Group

Internal Report # 228

Fast Wavelet Based Volume Rendering by Accumulation of Transparent Texture Maps

L. Lippert, M. H. Gross

Computer Science Department
ETH-Zürich, Switzerland
E-Mail: grossm@inf.ethz.ch, lippert@inf.ethz.ch
http://www.inf.ethz.ch/department/IS/cg/

To appear in **COMPUTER GRAPHICS forum, Vol. 14, No. 3, 1995**

Fast Wavelet Based Volume Rendering by Accumulation of Transparent Texture Maps

L. Lippert, M. H. Gross
Institute for Information Systems
Computer Science Department
Swiss Federal Institute of Technology
ETH–Zentrum
8092 Zürich
Tel: +41 1 632 7114
Fax: +41 632 1172
E–mail: lippert@inf.ethz.ch, grossm@inf.ethz.ch

Key Words and Phrases: Volume Rendering, Intensity Integration, Wavelet Transform, Projection Slice Theorem, Texture Mapping, Accumulation Buffering, Fourier Transform

Abstract

In the following paper, a new method for fast and accurate volume intensity and color integration is elaborated, which employs wavelet decompositions and texture mapping. At this point, it comprises and unifies the advantages of recently introduced Fourier domain volume rendering techniques and wavelet based volume rendering. Specifically, the method computes analytic solutions of the ray intensity integral through a single wavelet by slicing its Fourier transform and by backprojecting it into the spatial domain. The resulting slices can be considered as RGB textures where R, G, and B account for the decomposed volume color function. Due to the similarity of the basis functions, the computation of the texture map has to be figured out only once for each 3D mother wavelet. Hence, the final volume rendering procedure turns out to be a superposition of self–similar, transparent and colored textures, which is supported by modern hardware accumulation buffers. Linear shading and attenuation can be introduced by modifications of the wavelet's Fourier transform.

The main advantages of this method are the provision of accurate solutions and quantification of error bounds, the absence of any expensive prefiltering and the independence of the computational costs from the image resolution. Furthermore, any required discretization, such as the resolution of the basis textures is defined within the computational framework of the wavelet transform. The method is not restricted to a specific type of wavelet unless is provides an analytic Fourier description, such as any B–spline wavelets do.

1 Introduction

Volume rendering is one of the mainstream research lines in Computer Graphics covering a broad range of applications, such as medical imaging [10], material sciences, computational chemistry, quantum physics or atmospherical optics. Basic work had been done by [3] and [11], where the relationship between the volume rendering problem and the transport equations of light in matter is pointed out. In the last decade, most attraction had been spent on the development of algorithms for fast numeric approximations of the low–albedo volume rendering integral [6]. [14] for instance, proposed a first order approximation of the integral using a uniform sampling of the intensity values on the ray along with trilinear interpolations. Shading of the volume is figured out by estimations of the normal through the volume intensity gradient. Good mathematical analysis of the error bounds in volume rendering is given in [19]. Isosurfaces are treated in different ways. [15] provide polygonial approximations using look–up table methods, others [18] compute more accurate descriptions with respect to the topological problems in isosurfaces.

Recently, various approaches were proposed based on different types of transform coding for volume rendering. [12] stressed a basis of hierarchical splatting functions and [17] employed computations of isosurfaces in volume data by ray–tracing the wavelet basis functions immediately. [9] approximated the volume rendering integral in wavelet space using piecewise polynomials and analytic integrations, whereas [22] figured out the rendering in the spatial domain and controlled the stepsize in wavelet space. The essential theoretical advantages of hierarchical decompositions are accurate computations of integrals, gradients and error bounds along with smooth, analytic approximations resulting in sophisticated level–of–detail control. Unfortunately, these advantages could not be harvested optimally up to now, since the approaches are either too expensive or too rough.

Apart from wavelet based volume rendering, Fourier domain methods, such as projection slicing have a long tradition in CT and MR technology [7]. [16] accomplished fast volume intensity integrations of order *$O(n^2 \log n)$* by slicing and backprojecting the Fourier transform of the volume intensity function. [20] extended this concept by adding linear shading and attenuation. The most time consuming step in this method is the application of a reconstruction filter. Although this method does not perform exponential self–occlusion, it is still important for many applications, especially with regard to hardware acceleration.

The method proposed in this paper combines the advantages of both wavelet decompositions and Fourier domain descriptions for fast and accurate volume intensity and color integration. For this end, we employ orthonormal wavelets, which are defined analytically in frequency space. The volume intensity integral along the ray results by superimposing the local integrals through all wavelets intersected by the ray. However, the local intensity integral through a single wavelet can easily be computed by slicing its Fourier transform. Since it is defined analytically, our slice is described analytically as well and no reconstruction filtering is required to compute the backprojection in spatial domain. Moreover, the sliced wavelets in Fourier space can be interpreted as local intensity integrals of bundles of intersecting rays. Due to the similarity of the basis functions, all slices are derived from each other just by scaling and translating one prototype function computed for each 3D mother wavelet. This can be accomplished for each color component R, G, B and the backprojected slices are treated as self–similar, transparent, colored textures. The required resolution of the basis textures can be computed from Nyquist's sampling theorem. The final rendering process is figured out by weighted accumulation of these colored and transparent wavelet textures, and is evidently supported by the hardware of most high performance graphics workstations [4]. Hence, the computational costs do not depend on the resolution of the image, because we replace the ray–tracing by texture mapping. The embedding of our method within the framework of the wavelet transform allows us to harvest all advantages arising from it, such as control of error and level of detail.

The organization of this contribution is as follows: First of all, the mathematical foundations of 3D separable wavelet bases are briefly addressed and different prototypes in frequency space are introduced. Then, the process of volume intensity integration in wavelet space is illuminated and light is shed on how to compute local line integrals by the projection slice theorem. The next section explains our method in detail and illustrates, how to solve the ray tracing problem by mapping of self–similar basis textures. Finally, we show the outperformance of our method on different volume data sets.

2 Mathematical Foundations

2.1 Construction of 3D–Orthonormal Wavelets

Separable, multidimensional extensions of orthonormal wavelet–functions can easily be obtained according to [5] by tensor products of three 1D prototype wavelets and scaling functions. Consequently, the respective 3D subspaces of $V_m^1 \in L^2(\mathbb{R})$ are defined as

$$V_m^3 = V_m^1 \otimes V_m^1 \otimes V_m^1 \tag{1}$$

The scaled and shifted versions of the scaling function $\phi \in L^2(\mathbb{R})$ with $\phi_{mp}(x) = 2^{-\frac{m}{2}}\phi(2^{-m}x - p)$ provide orthonormal bases of the vectorspaces V_m^1, whereas the scaled and shifted versions of the wavelet $\psi \in L^2(\mathbb{R})$ with $\psi_{mp}(x) = 2^{-\frac{m}{2}}\psi(2^{-m}x - p)$ span the orthonormal complement U_m^1. This results to:

$$\begin{aligned} V_m^3 &= \left(V_{m+1}^1 \oplus U_{m+1}^1\right) \otimes \left(V_{m+1}^1 \oplus U_{m+1}^1\right) \otimes \left(V_{m+1}^1 \oplus U_{m+1}^1\right) \\ V_m^3 &= V_{m+1}^3 \oplus U_{m+1}^{3,1} \oplus U_{m+1}^{3,2} \oplus \dots \oplus U_{m+1}^{3,7} \end{aligned} \tag{2}$$

The corresponding 3D versions of the wavelets and of the scaling function are yield from their one–dimensional relatives, as

$$\begin{aligned}
\phi_{mpqr}^3(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\phi(2^{-m}y - q)\phi(2^{-m}z - r) \in V_m^3 \\
\psi_{mpqr}^{3,1}(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\phi(2^{-m}y - q)\psi(2^{-m}z - r) \in U_m^{3,1} \\
\psi_{mpqr}^{3,2}(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\psi(2^{-m}y - q)\phi(2^{-m}z - r) \in U_m^{3,2} \\
\psi_{mpqr}^{3,3}(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\psi(2^{-m}y - q)\psi(2^{-m}z - r) \in U_m^{3,3} \\
\psi_{mpqr}^{3,4}(x,y,z) &:= 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\phi(2^{-m}y - q)\phi(2^{-m}z - r) \in U_m^{3,4} \\
\psi_{mpqr}^{3,5}(x,y,z) &:= 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\phi(2^{-m}y - q)\psi(2^{-m}z - r) \in U_m^{3,5} \\
\psi_{mpqr}^{3,6}(x,y,z) &:= 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\psi(2^{-m}y - q)\phi(2^{-m}z - r) \in U_m^{3,6} \\
\psi_{mpqr}^{3,7}(x,y,z) &:= 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\psi(2^{-m}y - q)\psi(2^{-m}z - r) \in U_m^{3,7}
\end{aligned} \tag{3}$$

We can constitute an orthonormal basis of $L^2(\mathbb{R}^3)$ by the dilated and translated versions of ψ and ϕ. In eq. (3) m denotes the scaling and p, q, r denote the shifts in x, y, z. By applying eq. (2) the reconstruction of a continuous approximation of the volume intensity function *f(x,y,x)* can be written as

$$(4) \qquad f(x,y,z) := \sum_{p,q,r} c_{Mpqr}\phi^3_{Mpqr}(x,y,z) + \sum_{m=1}^{M}\sum_{i=1}^{7}\sum_{p,q,r} d^i_{mpqr}\psi^{3,i}_{mpqr}(x,y,z)$$

where M denotes the depth of the pyramid and

$$(5) \qquad c_{Mpqr} = \left\langle f\ ,\ \phi^3_{Mpqr}\right\rangle \quad d^i_{mpqr} = \left\langle f\ ,\ \psi^{3,i}_{mpqr}\right\rangle \qquad <\ ,\ > \ : \textit{inner product operator}$$

Typically, the fast discrete wavelet transform of a given function is implemented using so–called quadrature mirror pair filter cascades (QMF), as illustrated in fig.1. The decomposition of an initial volume data set is accomplished by subsequent filtering and subsampling of each row, column and plane of the volume. A good mathematical analysis of QMFs is given in [1].

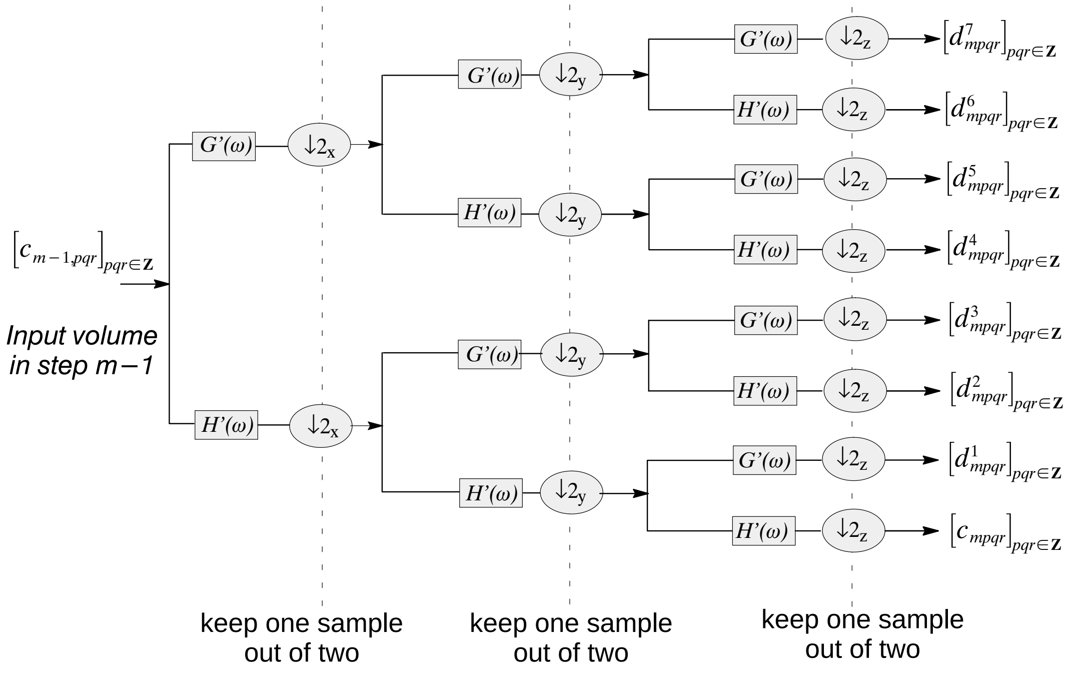
$[c_{m-1,pqr}]_{pqr\in\mathbf{Z}}$
Input volume in step m−1
G'(ω)
H'(ω)
↓2x
↓2y
↓2z
$[d^7_{mpqr}]_{pqr\in\mathbf{Z}}$
$[d^6_{mpqr}]_{pqr\in\mathbf{Z}}$
$[d^5_{mpqr}]_{pqr\in\mathbf{Z}}$
$[d^4_{mpqr}]_{pqr\in\mathbf{Z}}$
$[d^3_{mpqr}]_{pqr\in\mathbf{Z}}$
$[d^2_{mpqr}]_{pqr\in\mathbf{Z}}$
$[d^1_{mpqr}]_{pqr\in\mathbf{Z}}$
$[c_{mpqr}]_{pqr\in\mathbf{Z}}$
keep one sample out of two
keep one sample out of two
keep one sample out of two

Fig. 1 QMF–pyramid to implement the 3D–wavelet transform. The filters *G'* and *H'* are derived from the scaling function and from the wavelet, respectively (see [8])

2.2 Wavelets in Frequency Domain

Our approach requires wavelets, whose Fourier transform is described analytically. With regard to computational costs, the following three types are considered, although it has to be noted, that the method can be applied to any other type of wavelet satisfying the upper criterion.

1. Battle–Lemarie Wavelets

Lemarie [13] and Battle [2] studied a class of multiresolution approximations of $L^2(\mathbb{R})$. They constructed a set of wavelets by orthogonalizing B–spline functions and received symmetric wavelets and scaling functions. From a piecewise polynomial of degree n ($P_n(n) \in C^{n-1}$) they generated wavelets with $n+1$ vanishing moments. The Fourier transform of the BL–scaling function for $n = 3$ is defined as:

$$(6) \qquad \Phi(\omega) := \frac{1}{\omega^4 \sqrt{\Sigma_8(\omega)}}$$

with:

$$(7) \qquad \Sigma_8(\omega) := \frac{N_1(\omega) + N_2(\omega)}{105\left(\sin\frac{\omega}{2}\right)^8}$$

$$(8) \qquad N_1(\omega) := 5 + 30\left(\cos\frac{\omega}{2}\right)^2 + 30\left(\sin\frac{\omega}{2}\right)^2\left(\cos\frac{\omega}{2}\right)^2$$

$$(9) \qquad N_2(\omega) := 2\left(\sin\frac{\omega}{2}\right)^4\left(\cos\frac{\omega}{2}\right)^2 + 70\left(\cos\frac{\omega}{2}\right)^4 + \frac{2}{3}\left(\sin\frac{\omega}{2}\right)^6$$

2. Haar Wavelets

The generic Haar wavelet is simple and symmetric on the one hand, but however has a low regularity on the other. These wavelets are optimally localized in spatial–domain but, according to the Heisenberg uncertainty principle, the frequency localization has to be poor. One of the essential advantages of the Haar wavelet is the compact support along with non–overlapping basis functions.

For the mother scaling function and for its corresponding wavelet we obtain

$$(10) \qquad \phi(x) := \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \qquad \psi(x) := \begin{cases} 1 & \text{for } 0 \le x < 1/2 \\ -1 & \text{for } 1/2 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

And in frequency domain

$$(11) \qquad \Phi(2\omega) = e^{-i\omega}\frac{\sin(\omega)}{\omega} \qquad \Psi(2\omega) = ie^{-i\omega}\frac{\sin^2(\omega/2)}{\omega/2}$$

3. Shannon wavelets

Inverse properties of spatial/frequency localization brings up the Shannon wavelet. Here, the

Fourier transform is compactly supported, whereas the corresponding functions in spatial domain are spread out.

$$(12) \quad \Phi(\omega) := \begin{cases} 1 & \text{for } |\omega| < \pi \\ 0 & \text{otherwise} \end{cases} \qquad \Psi(\omega) := \begin{cases} 1 & \text{for } \pi < |\omega| < 2\pi \\ 0 & \text{otherwise} \end{cases}$$

$$(13) \quad \phi(x) := \frac{\sin(\pi x)}{\pi x} \qquad \psi(x) := \left[\frac{\sin\left(\frac{\pi}{2}x\right)}{\frac{\pi}{2}x}\right]\cos\left(\frac{3\pi}{2}x\right)$$

In fig. 2 the functional courses of the 3 mother scaling functions from above are depicted in spatial and in frequency domain. The different functional properties can be realized easily.

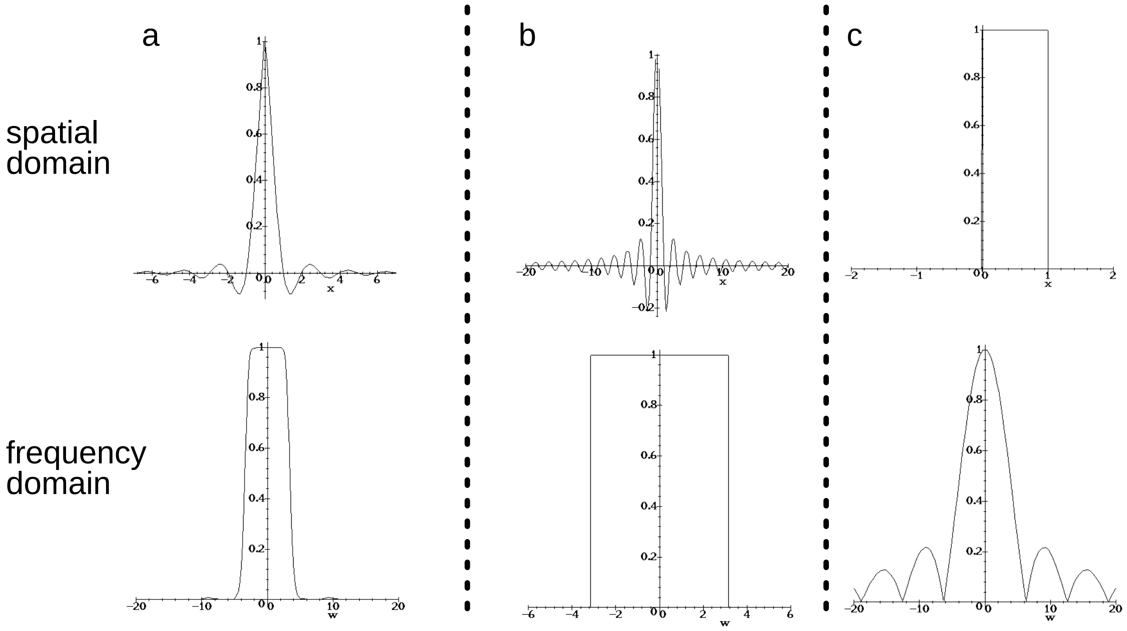

Fig. 2 Different scaling functions in spatial and frequency domain:
a) Battle–Lemarie wavelet
b) Shannon wavelet
c) Haar wavelet

The Fourier theorems state, that scaling and shifting of a prototype function has to hold Parseval's energy equivalent and is related according to eq. (14). In other words, a contraction in spatial domain is related to an expansion in frequency domain and vice versa. For that we have for any function g:

$$(14) \quad g_{ab}(x) = \frac{1}{\sqrt{a}}g\left(\frac{x-b}{a}\right) \Leftrightarrow G_{ab}(\omega) = \sqrt{a}\,G(a\omega)e^{-ib\omega} \qquad a \in \mathbb{R}^{+}, b \in \mathbb{R}$$

With eq. (14) for the dyadic case:

$$(15) \quad \phi_{mp}(x) = \frac{1}{\sqrt{2^m}}\phi(2^{-m}x - p) \Leftrightarrow \Phi_{mp}(\omega) = \sqrt{2^m}\,\Phi(2^m\omega)e^{-ip2^m\omega} \qquad m \in \mathbb{N}_0,\ p \in \mathbf{Z}$$

In eq. (1) we defined a separable 3D extension of a closed $L^2(\mathbb{R})$ subspace, whose basis functions are defined in eq. (3). Due to the separability of the Fourier transform, the corresponding definition in frequency domain results for the scaling function :

$$(16) \quad \Phi^3_{mpqr}(\omega_{1,}\omega_2, \omega_3) = \Phi_{mp}(\omega_1)\Phi_{mq}(\omega_2)\Phi_{mr}(\omega_3)$$

and for the wavelets:

$$(17) \quad \Psi^{3,k}_{mpqr}(\omega_{1,}\omega_2, \omega_3) = \Theta^{k\,div\,4}_{mp}(\omega_1)\Theta^{(k\,div\,2)\,\mathrm{mod}\,2}_{mq}(\omega_2)\Theta^{k\,\mathrm{mod}\,2}_{mr}(\omega_3) \qquad k \in [1, 7]$$

$$\Theta^0_{mn} := \Phi_{mn}(\omega)$$

$$\Theta^1_{mn} := \Psi_{mn}(\omega)$$

Note, that eq.(16) and (17) define the Fourier transform of a set of orthogonal, separable 3D scaling functions and wavelets, which will be processed further with our method.

3 Ray Tracing in Wavelet Space

3.1 Ray Intensity Function

As stated earlier, self–occlusion is not considered in this approach. Although the exponential term in the low–albedo equation [3] is of fundamental physical importance, yet for many applications a simple color integral renders sufficient results. Thus, the rendering problem reduces to solving line integrals along the ray *t* through the color function

$f^{col}(x, y, z) \qquad , col \in \{R, G, B\}$

Let $t_1$be the entrance point of the ray into the volume and t_2 be its exit, the intensity

i^{col} in each channel RGB is expressed by

$$(18) \quad i^{col} = \int_{t_1}^{t_2} f^{col}(x(t), y(t), z(t))dt$$

where *x(t), y(t), z(t)* are the coordinates in parameter space *t.*

Because of the wavelet expansion in eq. (4) the upper integrals can be considered as a superposition (accumulation) of the line integrals of 3D wavelets intersected by the ray

$$(19)\qquad i^{col} = \sum_{p,q,r} c^{col}_{Mpqr} \int_{-\infty}^{\infty} \phi^{3}_{Mpqr}(x(t), y(t), z(t))dt + \sum_{m=1}^{M} \sum_{k=1}^{7} \sum_{p,q,r} d^{k,col}_{mpqr} \int_{-\infty}^{\infty} \psi^{3,k}_{mpqr}(x(t), y(t), z(t))dt$$

Within this process the parametrization of the wavelet in *t* turns out to be essentially a projection onto the ray.

The goal of our further considerations is to find a fast algorithm to solve the fundamental relationship of eq. (19) analytically.

3.2 The Fourier Projection–Slice Theorem

One approach for rendering volume data sets is based on the inverse problem of tomographic reconstruction [16]. This is briefly reviewed in the following section. Generally, the aim is to achieve a fast computation of the density integral of a given 2D function *f(x,y)* along a line *l(t,μ)*. The Fourier projection–slice theorem is given as:

$$(20)\qquad P_\theta(\omega) = F(\omega\cos\theta, \omega\sin\theta)$$

where $P_\theta(\omega)$ is the Fourier transform of

$$(21)\qquad p_\theta(\mu) = \int_{-\infty}^{\infty} f(x(t,\mu), y(t,\mu))dt$$

and (ω, θ) define the frequency plane. Thus a slice of the Fourier transform of a given function *f(x,y)* at an angle θ represents the one–dimensional Fourier transform of its projection $p_\theta(\mu)$. Fig. 3 illustrates this relationship.

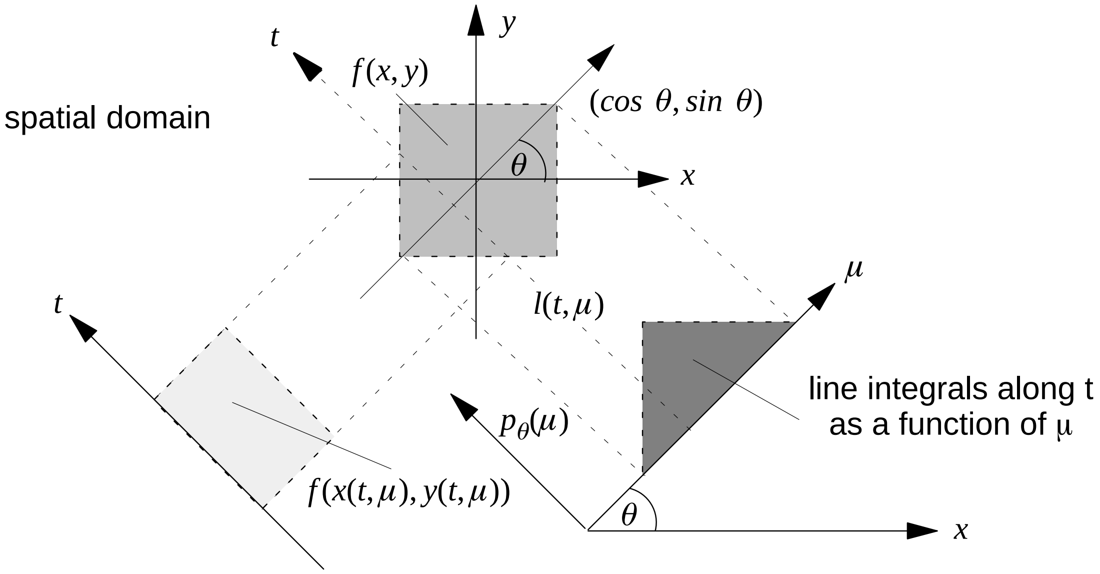

Fig. 3 Illustration of the Fourier projection slice theorem in 2D

In 3D, the Fourier projection–slice theorem can be used for calculating the intensity integrals of a function *f(x,y,z)* at a given point (μ, ν) onto an oriented projection plane along a perpendicular ray *t*. We get the intensity $i(\mu, \nu)$ in the image plane as:

$$i(\mu, \nu) = \int_{-\infty}^{\infty} f(x(t,\mu,\nu), y(t,\mu,\nu), z(t,\mu,\nu))dt \tag{22}$$

The computed intensity distribution and its parameters in spatial domain are represented in fig. 4.

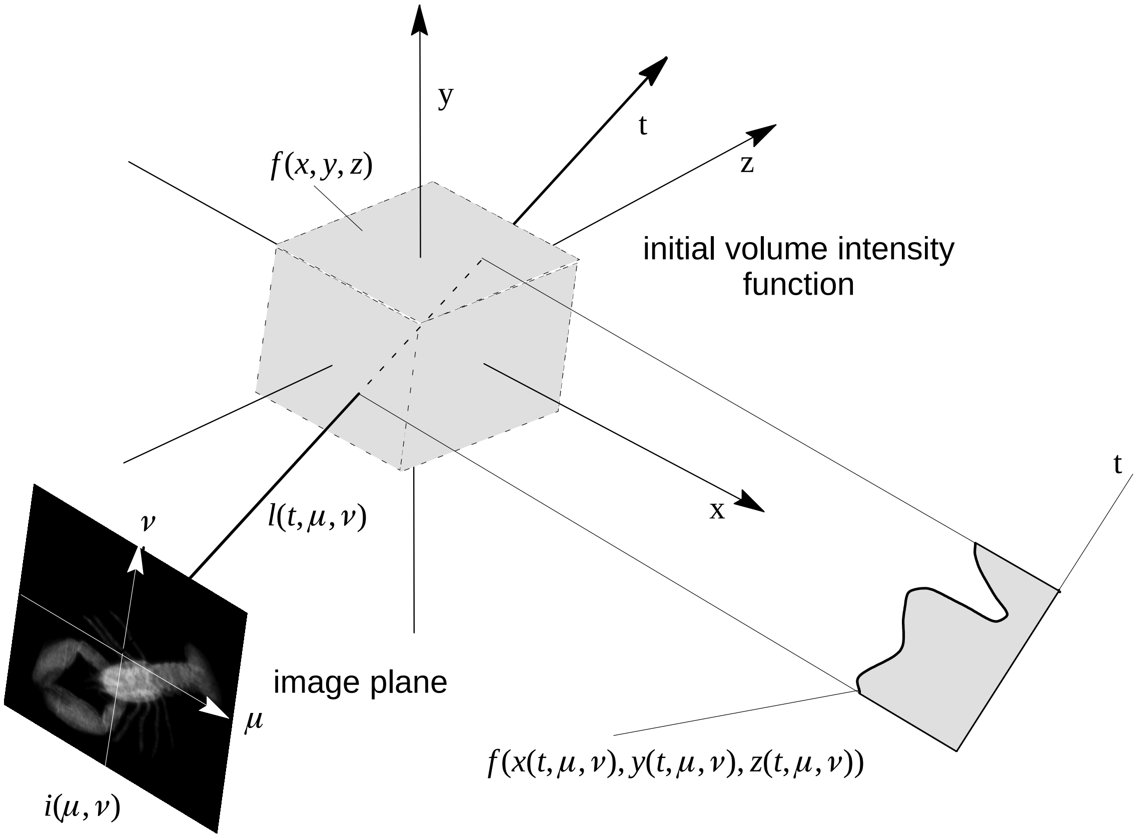
y
t
z
$f(x,y,z)$
initial volume intensity function
$l(t,\mu,\nu)$
ν
x
t
image plane
μ
$f(x(t,\mu,\nu),y(t,\mu,\nu),z(t,\mu,\nu))$
$i(\mu,\nu)$

Fig. 4 Illustration of the Fourier projection slice theorem in 3D

The 3D Fourier transform of our function *f(x,y,z)* is given by:

$$(23)\quad F(\omega_1,\omega_2,\omega_3) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)e^{-i2\pi(x\omega_1+y\omega_2+z\omega_3)}dxdydz$$

According to Fourier projection slice theorem [7] we have to compute $F(\omega_{1,}\omega_2,\omega_3)$ along the slicing plane through the origin defined by two vectors ***u*** and ***v*** in order to obtain *I(u,v).*

$$(24)\quad F(\omega_1(u,v),\omega_2(u,v),\omega_3(u,v)) = I(u,v)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)e^{-i2\pi(x\omega_1(u,v)+y\omega_2(u,v)+z\omega_3(u,v))}dxdydz$$

The inverse FFT of the resulting 2D function $I(u,v)$ leads to the intensity integral (eq. (22)), of a bundle of parallel rays perpendicular to the plane (***u***,***v***). This image can be considered as an X–ray image of the volume, because of the lack of attenuation or illumination effects.

3.3 Slicing 3D–Wavelets

In order to compute the upper intensity integral in wavelet space, each 3D wavelet intersected by the ray has to be sliced. This is accomplished by a slicing plane *P* perpendicular to the ray. It is specified by its normal **n** and by the vectors **u** and **v** which define the viewing coordinates:

$$(25)\quad \boldsymbol{n} := \begin{pmatrix} \cos\alpha\cos\beta \\ \sin\alpha\cos\beta \\ \sin\beta \end{pmatrix} \qquad \boldsymbol{u} := \begin{pmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{pmatrix} \qquad \boldsymbol{v} := \boldsymbol{n} \times \boldsymbol{u} = \begin{pmatrix} -\cos\alpha\sin\beta \\ -\sin\alpha\sin\beta \\ \cos\beta \end{pmatrix}$$

This procedure is shown in fig. 5.

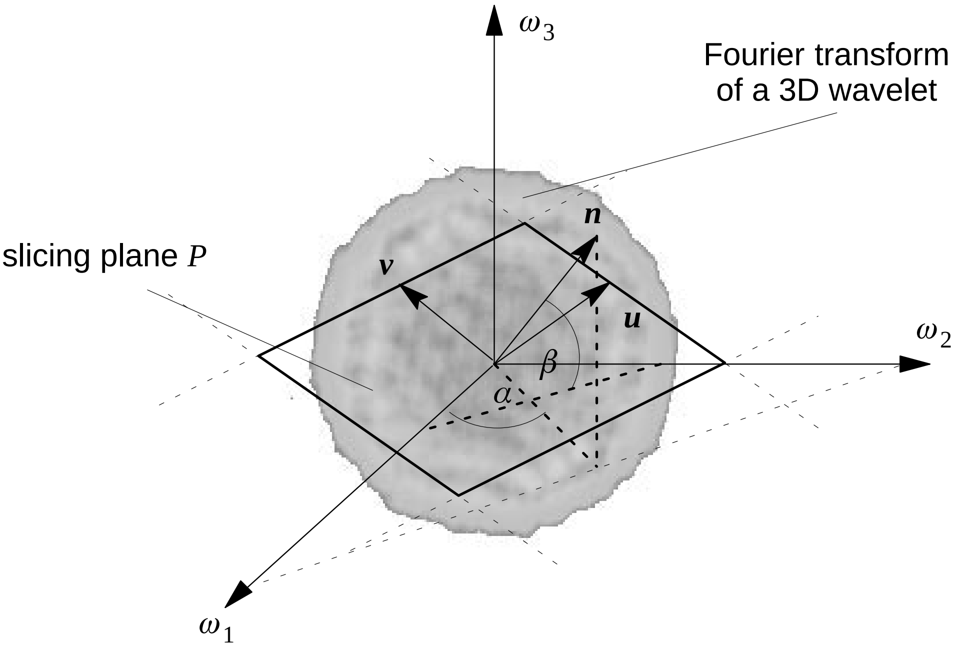

Fig. 5 Slicing the Fourier transform of a 3D wavelet, coordinate system

In most cases, the 2D slice $I(u, v)$ can be computed analytically, but not its counterpart $i(\mu, \nu)$. Therefore it is necessary to perform an inverse FFT algorithm and to generate a discrete version of size $(res_u + 1) \times (res_v + 1)$ of $I(u, v)$. This is figured out by the introduction of the two parameters *i* and *j* and by the stepsizes $\Delta_u = \frac{1}{res_u}$ and $\Delta_v = \frac{1}{res_v}$. The points $\boldsymbol{u}_i + \boldsymbol{v}_j$ define a discretization of the plane *P*, that slices the function $F(\omega_1, \omega_2, \omega_3)$:

$$(26)\qquad \boldsymbol{u}_i := \begin{pmatrix} -\sin\alpha\ \ i\ \Delta_u \\ \cos\alpha\ \ i\ \Delta_u \\ 0 \end{pmatrix} \qquad \boldsymbol{v}_j := \boldsymbol{n} \times \boldsymbol{u} = \begin{pmatrix} -\cos\alpha\sin\beta\ \ j\ \Delta_v \\ -\sin\alpha\sin\beta\ \ j\ \Delta_v \\ \cos\beta\ \ j\ \Delta_v \end{pmatrix}$$

with $i \in \left[-\frac{res_u}{2}, \frac{res_u}{2}\right]$ and $j \in \left[-\frac{res_v}{2}, \frac{res_v}{2}\right]$. The triple $(\omega_1, \omega_2, \omega_3)$ can be substituted by the parameterized plane such that

$$(27) \quad (\omega_{1,}\omega_{2,}\omega_3)_{i,j}{}^T := \begin{pmatrix} -\sin\alpha \ \ i\ \Delta_u - \cos\alpha\sin\beta\ j\ \Delta_v \\ \cos\alpha \ \ i\ \Delta_u - \sin\alpha\sin\beta\ j\ \Delta_v \\ \cos\beta\ j\ \Delta_v \end{pmatrix}$$

Applying eq. (27) on the 3D basis functions described in eq. (16) and eq. (17), yields for the 3D scaling function and for the 7 wavelets

$$\begin{aligned}(28) \qquad \Phi^3_{mpqr}\Big((\omega_{1,}\omega_2, \omega_3)_{i,j}\Big) &= \tilde{\Phi}^3_{mpqr}(i,j) \\ &= \Phi_{mp}(-\sin\alpha\ \ i\ \Delta_u - \cos\alpha\sin\beta\ j\ \Delta_v) \\ &\ \ \cdot \Phi_{mq}(\cos\alpha\ \ i\ \Delta_u - \sin\alpha\sin\beta\ j\ \Delta_v) \\ &\ \ \cdot \Phi_{mr}(\cos\beta\ j\ \Delta_v)\end{aligned}$$

$$\begin{aligned}(29) \qquad \Psi^{3,k}_{mpqr}((\omega_{1,}\omega_2, \omega_3)_{ij}) &= \tilde{\Psi}^{3,k}_{mpqr}(i,j) \\ &= \Theta^{k\, div\, 4}_{mp}(-\sin\alpha\ \ i\ \Delta_u - \cos\alpha\sin\beta\ j\ \Delta_v) \\ &= \Theta^{k\, div\, 4}_{mp}(-\sin\alpha\ \ i\ \Delta_u - \cos\alpha\sin\beta\ j\ \Delta_v) \\ &\ \ \cdot \Theta^{(k\, div\, 2)\bmod 2}_{mq}(\cos\alpha\ \ i\ \Delta_u - \sin\alpha\sin\beta\ j\ \Delta_v) \\ &\ \ \cdot \Theta^{k \bmod 2}_{mr}(\cos\beta\ j\ \Delta_v)\end{aligned}$$

$$\Theta^0_{mn} := \Phi_{mn}(\omega) \qquad \Theta^1_{mn} := \Psi_{mn}(\omega)$$

The inverse Fourier transform now renders the required line integrals along *t* through the basis functions ϕ and ψ at *(μ,ν)* because it holds

$$(30) \qquad \int_{-\infty}^{\infty} \phi^3_{mpqr}(x(t,\mu,\nu), y(t,\mu,\nu), z(t,\mu,\nu))dt = FT^{-1}\Big(\phi^3_{mpqr}\omega_1(u,v), \omega_2(u,v), \omega_3(u,v)\Big)$$

and

$$(31) \qquad \int_{-\infty}^{\infty} \psi^{3k}_{mpqr}(x(t,\mu,\nu), y(t,\mu,\nu), z(t,\mu,\nu))dt = FT^{-1}\Big(\psi^{3k}_{mpqr}\omega_1(u,v), \omega_2(u,v), \omega_3(u,v)\Big)$$

Our implementation employs an Cooley–Tukey FFT method. Note that the resolution at level *m* can be scaled with $res^m_{u,v} = res_{u,v}/m.$

3.4 Ray Casting by Accumulation of Sliced Wavelet Textures

The upper section addressed how to find the intensity of a given wavelet or scaling function. This concept can be used for computing the intensity integral of eq. (22) for any ray *t* perpendicular to our viewing plane *P(u,v)*. Supposing orthogonal projection (parallel rays) and inserting the definitions from above, we finally obtain a mathematical description of the intensity or color integral distribution $i^{col}(\mu, \nu)$ in wavelet space, according to eq. (18) and eq. (19).

$$
\begin{aligned}
(32) \quad i^{col}(\mu, \nu) &= \int_{-\infty}^{\infty} f^{col}(x(t,\mu,\nu), y(t,\mu,\nu), z(t,\mu,\nu))dt \\
&= \sum_{p,q,r} c^{col}_{Mpqr} \int_{-\infty}^{\infty} \phi^{3}_{Mpqr}(x(t,\mu,\nu), y(t,\mu,\nu), z(t,\mu,\nu))dt \\
&+ \sum_{m=1}^{M} \sum_{k=1}^{7} \sum_{p,q,r} d^{k,col}_{mpqr} \int_{-\infty}^{\infty} \psi^{3,k}_{mpqr}(x(t,\mu,\nu), y(t,\mu,\nu), z(t,\mu,\nu))dt
\end{aligned}
$$

Especially, this function defines the RGB image obtained by volume intensity integration of a function $f^{col}(x, y, z)$ approximated by a finite sum of self–similar ’colored’ wavelets.

Due to the alignment of the 3D wavelets onto the sampling grid, the rendering process can be interpreted as an accumulation of scaled and shifted versions of eight 2D intensity or color distributions. They have to be located at the respective wavelet’s position in space, are parallel to the viewing plane and are weighted with the coefficients of the transform. Fig. 6 reflects the entire rendering pipeline. First of all, the 3D wavelets and scaling function are sliced in frequency space. This is illustrated for the Shannon scaling function and can be computed analytically. This results in 8 different 2D textures, one for each mother wavelet representing their ray intensity integrals in frequency space. Furthermore, the initial RGB volume data set is decomposed by a 3D–WT, separately for each channel. This procedure provides the required coefficients c^{col} and d^{col}.

The WTs of the RGB volumes have only to be performed once at the beginning whereas the slicing has to be computed for each viewpoint. Next, the inverse FFT computes the intensity distribution in spatial domain. Note, that these distributions are the same for R,G and B but, following eq. (32) they have to be weighted with their respective coefficients c^{col} and d^{col}.

Superimposing these distributions for R,G and B results in an RGB texture representing the contribution of a single wavelet to the integral color function $f^{col}(x,y,z)$.

The linearity of eq. (32) allows us to accumulate scaled and shifted distributions to obtain the contributions of the corresponding basis functions. Hence, the final image is obtained by accumulation of a set of self–similar textures. Note, that these values are both positive and negative.

The advantage of this method is, that we can compute the slices in frequency space analytically and in some cases, the inverse FT as well. Thus, the volume rendering is taken as an accumulation of textures. The resolution of a single texture can be low and is defined by Nyquist’s theorem. In contrast to hierarchical spatting [12] we provide an expansion with orthonormal basis functions and have all advantages arising from it, such as smooth interpolations, level of detail, error control etc. These properties of WT–based methods are explained in detail in [9]. Another advantage lies in the hardware support of image accumulation. The SGI–ONYX, for instance, offers specific frame buffer operations or primitives, such as ’billboards’, which can be used immediately for implementing this method.

Fig. 6 Illustration of the entire rendering process using accumulation of scaled and translated texture maps

4 Examples

This section illustrates the method by rendering some volume data sets. In fig. 7 a Gaussian intensity distribution is rendered with Haar (a), Battle–Lemarie (b) and Shannon (c) wavelets. The RGB volume data set has a resolution of 64 x 64 x 64 voxels, where red refers to a density range of 0.2 to 0.35, green to a range of 0.35 to 0.6 and blue from 0.6 to 1.0. It is clear, that the different colors dominate in the respective spatial regions, but due to the superposition the center of the Gaussian appears in white color. The discontinuous Haar wavelet causes the picture to appear more ”boxlike” than the smooth wavelet types. The decomposition of the data was performed for M=0. Hence, the images are generated by scaling functions, only.

Figures 8–10 depict results obtained on the lobster data set at a resolution of 120 x 120 x 34. The outer hull of the lobster contains voxels of high density and is enhanced using red color (from

50 to 150), whereas the density range from 125 to 200 is rendered in green and 175 to 250 in blue. The figures present reconstructions with different wavelets at various decompositions.

Fig. 7 Gaussian density distribution rendered as an RGB image with
a) Haar wavelets
b) Battle–Lemarie wavelets
c) Shannon wavelets

The figures 8–10 (a) show the lobster at M=0, where only the scaling functions are used. These results are contrasted by fig. 8–10 (b,c), where decompositions with M=2 and M=1 were figured out and only the scaling functions are included, respectively. Evidently, the image appears more fuzzy, since the high frequency components are eliminated. Note, that the number of textures required for the reconstruction relates theoretically to $1/8^M$ in this case. Our implementation, however rejects any texture, whose coefficients are zero. Only the remaining textures are rendered. The respective percentages on non–vanishing textures are given in brackets.

Fig. 8 Lobster data set rendered with Haar wavelets at
a) M = 0 (14 % of the coefficients are used for rendering)
b) M = 1 (8 %)
c) M = 2 (2 %)

Fig. 9 Lobster data set rendered with Battle–Lemarie wavelets at
a) M = 0 (14 % of the coefficients are used for rendering)
b) M = 1 (8 %)
c) M = 2 (2 %)

Fig. 10 Lobster data set rendered with Shannon wavelets at
a) M = 0 (14 % of the coefficients are used for rendering)
b) M = 1 (10 %)
c) M = 2 (4 %)

The capabilities of the wavelet transform to control elegantly level of detail is finally elucidated in fig. 11. Different views are presented and the wavelet coefficients are filtered, such that all high frequency components are removed unless those within a small region around the left shear.

Fig. 11 Level of detail using Haar–wavelets
a) Original datasets (2 different views)
b) LOD control around the left shear at M=1 (14%)
c) LOD control around the left shear at M=2 (10%)

5 Conclusion

In this paper a method for fast volume intensity integration was introduced using RGB textures of wavelets sliced in the frequency domain. The computational advantages of this method lie specifically in the absence of additional prefiltering and in the self similarity of the textures. This allows to compute them only once for each mother wavelet. The rendering is accomplished by accumulation of transparent and colored textures and is supported in modern graphics hardware. Basically, the ray casting is replaced by polygon rendering or frame buffer operations and the computing costs are only correlated weakly to the resolution of the image.

Although we presented the pictures without shading and attenuation, it is mathematically straightforward to include these features as well. However, the problem of computing exponential self–occlusion and more complex nonlinear shading models is still unsolved. One way to approach it, is to replace the accumulation process by α–blending and depth sorting of the textured polygons. This can be thought as a linear approximation of the low–albedo integral and is in the focus of our current research activities. Furthermore, other biorthogonal B–spline wavelet types should be considered, such as in [21].

6 References

[1] Akansu, A. N., Haddad, R.A.: ”Multiresolution Signal Decomposition”. Academic Press, Inc., 1992

[2] Battle, G.: ”A block spin construction of ondelettes”. Comm. Math Phys., Vol. 110, pp. 601–615, 1987

[3] Blinn, J., F.: ”Light Reflection Functions for Simulation of Clouds and Dusty Surface”. Computer Graphics, Vol. 16, No. 3, pp. 116 – 123, 1993

[4] Cabral, B.; Cam N.; Foran, J.: ”Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware”. Proceedings 1994 Symposium on Volume Visualization, 1994

[5] Daubechies, I.: ”Ten Lectures on Wavelets”. No. 61 CBMS–NSF Series in Applied Mathematics, SIAM 1992

[6] Drebin, R.A.; Carpenter, L.; Hanrahan, P.: ”Volume Rendering”. Computer Graphics, Vol. 22, No. 4, pp. 125–134, 1988

[7] Dudgeon, D. E.; Mersereau, R.M.: ” Multidimensional digital signal processing”. Prentice–Hall, Inc., New Jersey, 1984

[8] Groß, M. : ”Visual Computing”. Springer–Verlag, 1994

[9] Groß, M.; Lippert, L.; Dreger, A.; Koch, R.: ”A New Method to Approximate the Volume Rendering Equation Using Wavelets and Picewise Polynomials”, Comptuers& Graphics, Vol. 19, No. 1, to appear, 1995

[10] Höhne, K.H.; Bomas, M.; Pommert, A.; Riemer, M.; Schiers, C.; Tiede, U.; Wiebecke, G.: ”3D Visualization of Tomographic Volume Data using Generalized Voxel Model”. The Visual Computer, Vol.6, No. 1, pp. 28–36, 1990

[11] Kajiya, J.T.; Von Herzen, B.P.: ”Ray Tracing Volume Densities”. SIGGRAPH ’84, pp. 165–174, 1984

[12] Laur, D.; Hanrahan, P.: ”Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering ”. Computer Graphics and Applications, Volume 25, Number 4, pp. 285–288, 1991

[13] Lemarie, P.G.: ” Ondelettes a localisation exponentielles”. J. Math. Pures Appl., Vol. 67, pp. 227–236, 1988

[14] Levoy, M.: ”Display of Surfaces from Volume Data”. IEEE Computer Graphics and Applications, Volume 8, Number 5, pp. 29–37, 1988

[15] Lorensen, W.E.; Cline, H.E.: ”Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. Computer Graphics, Vol. 21, pp. 163–196, 1987

[16] Malzbender, T.:”Fourier Volume Rendering”, ACM Transactions on Graphics, Vol. 12, No.3, pp. 233–250, July 1993

[17] Muraki, S.: ”Volumetric Shape Description of Range Data using ’Blobby Model’”. Computer Graphics, Vol. 25, No. 4, pp. 227–235, 1991

[18] Ning, P.; Hesselink, L.: ”Octree Pruning for Variable–Resolution Isosurfaces”. Visual Computing, Springer Verlag, Tokyo, pp. 349–363,1992

[19] Novins, K.; Arvo, J.: ”Controlled Precision Volume Integration”. ACM Workshop on Volume Visualization, pp. 83–89, 1992

[20] Totsuka, T., Levoy, M.: ”Frequency Domain Volume Rendering”. Computer Graphics Proceedings, Annual Conference Series, pp. 271–278, 1993

[21] Unser, M.; Eden, M.: ”Multiresolution Feature Extraction and Selection for Texture Segmentation”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, pp. 717 – 728, 1989

[22] Westermann, R.: ”A Multiresolution Framework for Volume Rendering”. Proceedings 1994 Symposium on Volume Visualization, 1994