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 # 220

A New Method to Approximate the Volume Rendering Equation Using Wavelets and Piecewise Polynomials

M. H. Gross; L. Lippert; A. Dreger; R. Koch

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 **Computers & Graphics, Vol. 19, No. 1, 1995**

A New Method to Approximate the Volume Rendering Equation using Wavelet Bases and Piecewise Polynomials *

M. H. Groß, L. Lippert, A. Dreger, R. Koch
Institute for Information Systems
Computer Science Department
Eidgenössische Technische Hochschule Zürich
ETH–Zentrum
8092 Zürich
Tel: +41 1 632 7291
Fax: +49 632 1172
E–mail: grossm@inf.ethz.ch

A Key Words and Phrases: Volume Rendering, Isosurface Reconstruction, Wavelets, Hierarchical Approximation, Ray–Casting, Spline Functions, QMF–Filtering

Abstract

In the following paper we describe a new generic method to find an approximate solution for the volume rendering equation using hierarchical, orthonormal wavelet basis functions. The approach is based on the idea that an initial volume data set can be decomposed into a pyramidal representation by means of a 3D wavelet transform. Once the wavelet function is described analytically, it is possible to approximate the volume density function. Moreover, when employing piecewise polynomial spline functions, as in our method, the rendering integral can also be approximated and gradient functions or related features of the data can be computed immediately from the approximation. Due to the localization properies of the wavelet transform both in space and in frequency on the one side and due to the pyramidal subband coding scheme on the other side, this technique allows additionally for the control of the local quality of the reconstruction and provides elegantly for level–of–detail operations.

Aside from the solution of the rendering equation itself, isosurfaces of the data can also be computed with either standard techniques, like marching cubes, or by more sophisticated algorithms that render the basis functions. All these additional rendering techniques can be embedded in a hybrid surface/volume rendering scheme. In our paper, we elucidate this new concept and show its capabilities by different examples.

1 Introduction

Volumetric rendering has become one of the mainstream research lines in Computer Graphics, since many applications require a true 3D insight into the data rather than to provide visualizations of surfaces only. For this end, many algorithms have been developed in the past to get realistic images from an initial volume data set. Application areas reach from medical imaging [10], [19] through to new material analysis and quantum physics [13] up to the simulation of fog and clouds [25]. Most of these approaches use a ray casting process along with a back to front sampling to get the final pixel intensity and color.

The basic goal of volume rendering is to find a good approximation of the low albedo volume rendering integral [2], [11] that expresses the relation between the volume intensity and opacity function and the intensity in the image plane. It is stressed by several authors, that the volume rendering problem goes back to a basic transport equation that regulates the interaction of light and matter in a particle model. Most of the standard volume rendering algorithms therefore approximate an integral equation with the ray parameter t of the type

$$I = \int_{t_1}^{t_2} C(t)\ e^{-\int_{t_1}^{t} \alpha(s)ds}\ dt \tag{1}$$

where *C(t)* stands for a volume intensity function and includes emitted, scattered and reflected light. $\alpha(t)$ is the opacity function of the data and can be used to encode data features to be enhanced in the final images. Hence, the inner integral includes the self–occlusion of the volume.

The most common way to get a numeric solution of eq. (1) employs a zero–order quadrature of the inner integral along with a first–order approximation of the exponential. The outer integral is also solved by a finite sum of uniform samples. We yield

$$I = \sum_{k=1}^{M} C_k\ \alpha_k \prod_{i=1}^{k-1} (1 - \alpha_i) \tag{2}$$

where α_k are the opacity samples along the ray and C_k are the local color values derived from the illumination model. Note that I has to be computed for each spectral sample λ, mostly in R,G,B.

A good mathematical analysis of the problem and error bounds of numeric quadrature is provided in [23]. The illumination of the volume has to follow an estimation of the surface normal in each voxel by means of the volume density gradient. Isosurfaces are usually provided either by modulation of the opacity function, where the opaque volume elements represent the required isosurface, or by polygon approximations as the marching cubes [17]. [5] proposed a probabilistic method to get the voxel colors associated to a specific tissue or material to avoid aliasing artifacts. Finally it can be stated that a good volume renderer has in particular to find a good approximation of the volume rendering integral. Due to the huge data sets to be handled, [15] proposed an hierarchical splatting of the volume with scalable basis functions based on

an octree data structure. Another interesting way was described by [22], who extended his initial method to approximate isosurfaces [20] taking advantage of wavelet basis functions. However this approach was limited to isosurface rendering and uses a very expensive scheme to ray–trace an implicit function that included the wavelet expansions.

But in spite of these shortcomings, once a hierarchical, compact and continuous approximation of the volume intensity function is found, it is straightforward to get the intensity function along the ray, to compute the respective gradients and normals and to adjust the quality of the reconstuction locally to interesting data features. Moreover, stressing the wavelet concept as mathematically formulated in [4], [18] or [9], the data set can be expanded by a set of orthonormal, self–similar basis functions derived from each other by scaling and shifting. Since the wavelet transform is both located in space and in frequency, it provides an elegant way to control locally the level of detail, as well as to get features that describe the local data properties. Therefore, wavelets have widely been used in image processing for texture feature extraction. As we know [6],[8], the segmentation of the initial volume data set is, however, a very important preprocessing step in the rendering process, and many researchers address this with different automatic [5] and semiautomatic[10] techniques. The properties of the wavelet transform offer a concept, that integrates both volume rendering and volume data analysis.

The following paper descibes a new method to approximate the volume rendering equation using wavelet transforms. For this purpose, the initial volume data set is transformed into wavelet space using separable 3D extensions of orthonormal wavelet types. Since some wavelets, as the Daubechies one, are not given in a closed form solution we approximate the basis
functions with piecewise polynomial splines. This allows a continuous hierarchical approximation of the 3D data set. Due to the local support of the wavelet bases, we can control efficiently the level–of–detail in the data. This enables the emphasizing of local features of interest. Once this continuous piecewise polynomial approximation is computed, the volume intensity function along the ray can easily be formulated and a linear approximation of the exponential absorption term provides a polynomial approximation of the entire rendering integral. This finally leads to an analytic solution.

Our paper is organized as follows: First of all, we elucidate the mathematical basics of the wavelet–transform, describe the different properties of the employed wavelets and show how to get separable 3D extensions. The second chapter sheds light on the continuous approximation of the data with wavelet bases and provides our approximation of the volume rendering integral for the ray–casting process. The third chapter explains how to get isosurfaces in wavelet spaces and compares results obtained from Kalra's [12] method to the marching cubes [17] reconstruction techniques. Finally we outline a concept for embedding volume data analysis into the rendering process.

2 Mathematical Foundations

2.1 Formulation of the Wavelet Transform

The wavelet–transform (WT) is an integral transform of any finite energy function $f(x) \in L^2(\mathbf{R})$ using a set of self–similar basis functions $\psi_{ab}(x)$. Its generic continuous form description is provided as the following inner product:

$$WT_{f,\psi}(a,b) = \langle f, \psi_{ab} \rangle = \int_{-\infty}^{\infty} \overline{\psi_{ab}(x)} f(x) dx \qquad a,b \in \mathbf{R} \tag{3}$$

The single basis functions are derived from each other by scaling and shifting one prototype function ψ controlled by the parameters a and b respectively [9].

$$\psi_{ab}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \tag{4}$$

One required property of the orthonormal bases is their band–pass behavior that is defined as

$$\Psi_{ab}(0) = 0, \ \Psi_{ab}(\omega) : \textit{Fourier Transform of } \psi_{ab}(x) \tag{5}$$

As any other type of linear transform the WT enables the decomposition and the expansion of the initial function $f(x)$ by superimposing the basis functions.

In order to handle this method with a computer, it is necessary to set up a discrete version. A dyadic scaling of the bases with $a = 2$ and a unit shift $b = 1$ yields:

$$\psi_{mn}(x) := 2^{-\frac{m}{2}} \psi(2^{-m}x - n) \qquad , (\psi_{mn})_{n \in \mathbf{Z}} \textit{ basis of vectorspace } U_m \tag{6}$$

In most cases, the bases are furthermore supposed to be orthonormal to each other [4]

$$\langle \psi_{mn}, \psi_{\tilde{m}\tilde{n}} \rangle = \int_{-\infty}^{\infty} \psi_{mn}(t) \psi_{\tilde{m}\tilde{n}}(t) dt = \delta_{m\tilde{m}} \delta_{n\tilde{n}} \tag{7}$$

$$\text{with} \quad \delta_{ij} := \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

Thus the transform with discrete orthonormal wavelets can be mathematically described as

$$DWT_{f,\psi}(m,n) = 2^{-\frac{m}{2}} \int_{-\infty}^{\infty} \overline{\psi(2^{-m}x - n)} \, f(x) \, dx \qquad m,n \in \mathbf{Z} \tag{8}$$

Mallat [18] stresses this concept by defining a set of multiresolution function spaces V_m that render an approximation of all $f(x) \in L^2(\mathbf{R})$. The space of resolution m is derived from the higher resolution step by a tensor product with the orthonormal complement U_m.

$$V_{m-1} = V_m \otimes U_m \tag{9}$$

It can be proofed that the self–similar so–called scaling functions $\phi \in L^2(\boldsymbol{R})$ with $\phi_{mn}(x) = 2^{-\frac{m}{2}}\phi(2^{-m}x - n)$ provide orthonomal bases of the vectorspaces V_m in each resolution step. The so–called wavelet $\psi \in L^2(\boldsymbol{R})$ with $\psi_{mn}(x) = 2^{-\frac{m}{2}}\psi(2^{-m}x - n)$ however is prooved to be a basis of the orthonormal complement U_m.

The statements explained above offer an iterated decomposition scheme, where an intial discrete function can successively be approximated for a given iteration depth M using the scaling function and wavelets in each orthonormal complement space U_m.

$$V_0 = V_M \otimes U_M \otimes \cdots \otimes U_0 \tag{10}$$

Hence, we obtain:

$$\begin{aligned} f(x) := \sum_j c_j^0 \phi_{0j}(x) &= \sum_j c_j^1 \phi_{1j}(x) + \sum_j d_j^1 \psi_{1j}(x) \qquad f(x) \in V_0 \\ &= \sum_j c_j^M \phi_{Mj}(x) + \sum_{k=1}^{M} \sum_j d_j^k \psi_{kj}(x) \end{aligned} \tag{11}$$

2.2 3D–Extensions and Implementation

So far, the definitions were restricted to the one–dimensional case. For multidimensional signal processing, as image analysis or volume rendering, it is necessary to extend the method to multiple dimensions. Beside of the non–trivial nonseparable case [26] there is a straightforward way to accomplish this by means of tensor products of the one–dimensional representatives of the function spaces and of their bases. Once V_m is given, we can define a 3D version V_m^3 by

$$\begin{aligned} V_m^3 &= V_m \otimes V_m \otimes V_m \\ V_m^3 &= \left(V_{m+1} \oplus U_{m+1}\right) \otimes \left(V_{m+1} \oplus U_{m+1}\right) \otimes \left(V_{m+1} \oplus U_{m+1}\right) \\ V_m^3 &= V_{m+1}^3 \oplus U_{m+1}^{3,1} \oplus U_{m+1}^{3,2} \oplus ... \oplus U_{m+1}^{3,7} \end{aligned} \tag{12}$$

Equation (12) shows, that at each decomposition step, the space is broken up into 7 orthonormal complements that account for the principal orientations of the data, respectively. The corresponding 3D versions of the wavelets and of the scaling function can easily be derived from their one–dimensional relatives, as

$$\begin{aligned} \phi_{mpqr}^3(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\phi(2^{-m}y - q)\phi(2^{-m}z - r) \\ \psi_{mpqr}^{3,1}(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\phi(2^{-m}y - q)\psi(2^{-m}z - r) \\ \psi_{mpqr}^{3,2}(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\psi(2^{-m}y - q)\phi(2^{-m}z - r) \\ \psi_{mpqr}^{3,3}(x,y,z) &:= 2^{-\frac{3m}{2}}\phi(2^{-m}x - p)\psi(2^{-m}y - q)\psi(2^{-m}z - r) \end{aligned} \tag{13}$$

$$\psi^{3,4}_{mpqr}(x,y,z) := 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\phi(2^{-m}y - q)\phi(2^{-m}z - r)$$

$$\psi^{3,5}_{mpqr}(x,y,z) := 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\phi(2^{-m}y - q)\psi(2^{-m}z - r)$$

$$\psi^{3,6}_{mpqr}(x,y,z) := 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\psi(2^{-m}y - q)\phi(2^{-m}z - r)$$

$$\psi^{3,7}_{mpqr}(x,y,z) := 2^{-\frac{3m}{2}}\psi(2^{-m}x - p)\psi(2^{-m}y - q)\psi(2^{-m}z - r)$$

Unfortunately, some of the wavelet types to be explained later are not defined in a closed form solution and consequently the convolution products of eq. (3) cannot be computed explicitely. The implementation of orthonormal wavelet transforms employs so–called quadrature mirror pair filters (QMFs). The basic scheme of this filter bank is illustrated in fig. 1 for a decomposition of volume data. The initial data set is filtered along the x– y– and z– axis and subsampled by the factor 2 using the two filters $H'(\omega) = \sum_{n=-\infty}^{\infty} h(-n)e^{in\omega}$ and $G'(\omega) = e^{-i\omega}\overline{H'(\omega + \pi)}$ with $h(n) := 2^{-\frac{1}{2}} \int_{-\infty}^{\infty} \phi\left(\frac{x}{2}\right)\phi(x - n)dx$ respectively. The result from this process are 7 detail signals $D^{d1}_m f, \cdots, D^{d7}_m f$ that account for the oriented wavelets in this channel and a low pass signal $A_m f$ that is decomposed further. This iterated scheme corresponds to a dyadic subband coding of the data.

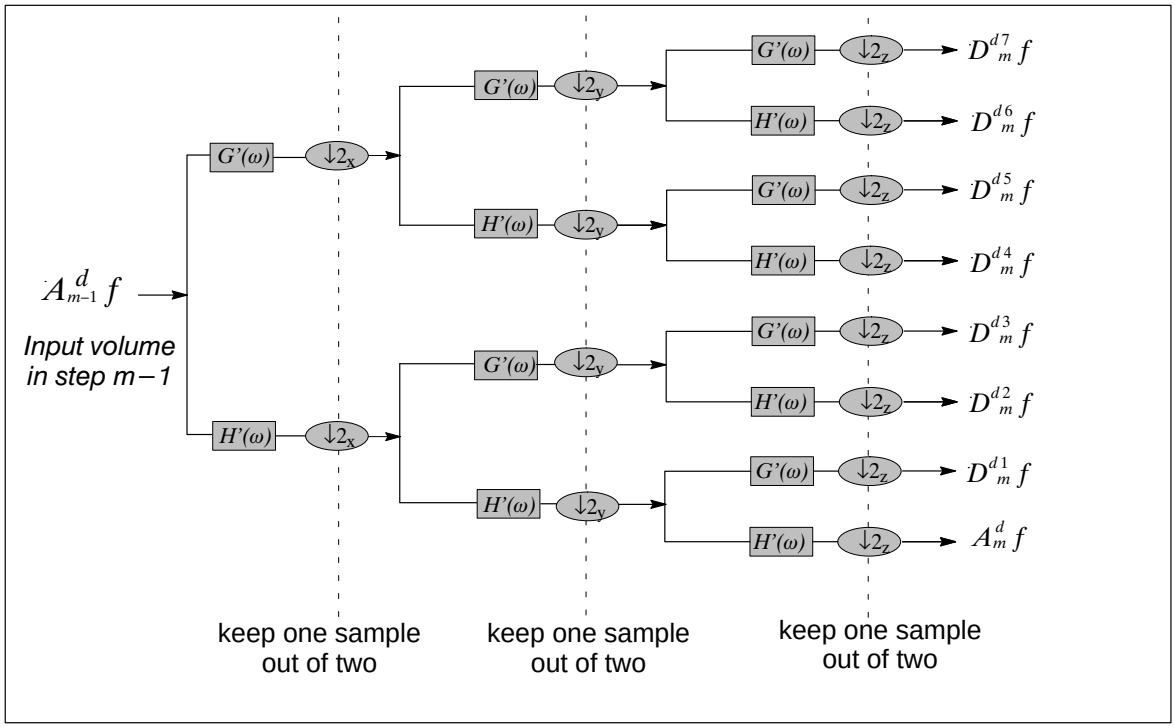

Fig. 1 Quadrature mirror pair filter pyramid to implement the 3D wavelet transform

The resulting data pyramid is illustrated in fig. 2. In each branch of this tree the different signal components are emphasized, that were extracted from the corresponding oriented wavelet function. This method al-

lows the efficient implementation of the WT without an explicit definition of the basis function. The problem reduces to the finding of the filter pairs $H'(\omega)$ and $G'(\omega)$.

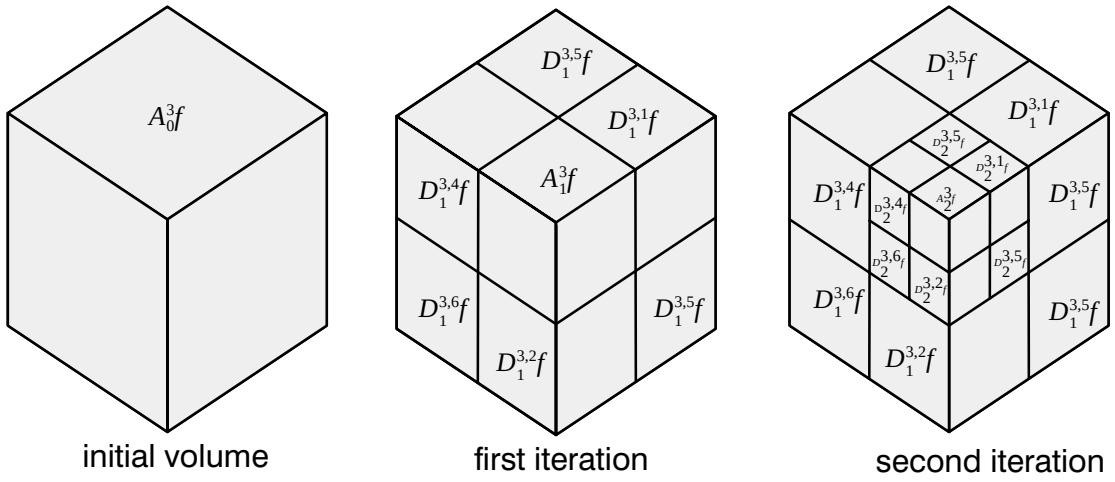

Fig. 2 Iterated decomposition of the initial volume data by the wavelet transform

2.3 Wavelet Basis Functions

The elucidations above do not further restrict the mathematical properties of the wavelet bases and there have been different types proposed in literature depending on smoothness, strict local support and other criteria. This section briefly introduces the most important wavelets we employ for our rendering method. The authors also used them successfully for texture analysis [8].

Haar Wavelets

A very simple, but discontinuous basis is given with the Haar wavelet, whose scaling function and corresponding wavelet is defined by

$$\phi(x) := \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \qquad \psi(x) := \begin{cases} 1 & \text{for } 0 \le x \le 1/2 \\ -1 & \text{for } 1/2 \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The Fourier transform of these functions (eq. (15)) show, that these expressions have optimal localization properties in the spatial domain on the one hand

$$\Phi(2\omega) = e^{-i\omega} \frac{\sin(\omega)}{\omega} \qquad \Psi(2\omega) = ie^{-i\omega} \frac{\sin^2(\omega/2)}{\omega/2} \tag{15}$$

but a rather bad localization in frequency on the other hand.

Daubechies Wavelets

In order to obtain better localization in frequency along with a minimal local support in space and smoother basis functions, Daubechies [4] proposes wavelet types as follows: the smoothness can be measured by the regularity *r*, (of any function ϕ) which is defined as the maximum of *r* such that

$$|\Phi(\omega)| \le \frac{c}{(1 + |\omega|)^{r+1}} \qquad c \in \mathbb{R}^+ \tag{16}$$

This relation also describes the continuity of $\phi(x)$, where $\phi(x) \in C^i(\mathbb{R})$ with $i \leq r$. The regularity of the Daubechies wavelets is proportional to the number of vanishing moments N, defined by

$$\int_{-\infty}^{\infty} t^n \psi(x) dx = 0 \quad n \in \{0, \ldots, N\} \tag{17}$$

The function ϕ is only constraint by eq. (18).

$$\int_{-\infty}^{\infty} \phi(x) dx = 1 \tag{18}$$

Note, that these wavelets are given by their corresponding QMF–pairs.

Coiflet – Bases

Restricting further the scaling function to a fixed number of vanishing moments following eq. (19).

$$\int_{-\infty}^{\infty} t^n \phi(x) dx = 0 \quad n \in \{1, \ldots, N'\} \tag{19}$$

results in a different wavelet type, the so–called Coiflet wavelet [1]. Besides the Haar–Wavelet, strict finite support comes along in the orthonormal case with a lack of symmetry. However there is a relationship between the number of vanishing moments, stated earlier, and the symmetry of the function. Thus Coiflet bases appear to be ”more symmetric” then Daubechies ones.

Battle–Lemarie Wavelets

Wavelets with an infinite support can be approximated in the frequency plane. An example is given by the equations (20) – (23) (see [18]).

$$\hat{\phi}(\omega) := \frac{1}{\omega^4 \sqrt{\Sigma_8(\omega)}} \tag{20}$$

where

$$\Sigma_8(\omega) := \frac{N_1(\omega) + N_2(\omega)}{105\left(\sin\frac{\omega}{2}\right)^8} \tag{21}$$

with

$$N_1(\omega) := 5 + 30\left(\cos\frac{\omega}{2}\right)^2 + 30\left(\sin\frac{\omega}{2}\right)^2\left(\cos\frac{\omega}{2}\right)^2 \tag{22}$$

and

$$N_2(\omega) := 2\left(\sin\frac{\omega}{2}\right)^4\left(\cos\frac{\omega}{2}\right)^2 + 70\left(\cos\frac{\omega}{2}\right)^4 + \frac{2}{3}\left(\sin\frac{\omega}{2}\right)^6 \tag{23}$$

The resulting wavelet type is often referred to in literature as the Battle–Lemarie wavelet.

Figure 3 gives again a graphical representation of the shape of some one–dimensional wavelet basis functions. We can cleary distinguish between the smooth shape of the Battle–Lemarie wavelet on the one side

and the strict local support of the Daubechies wavelet on the other side.

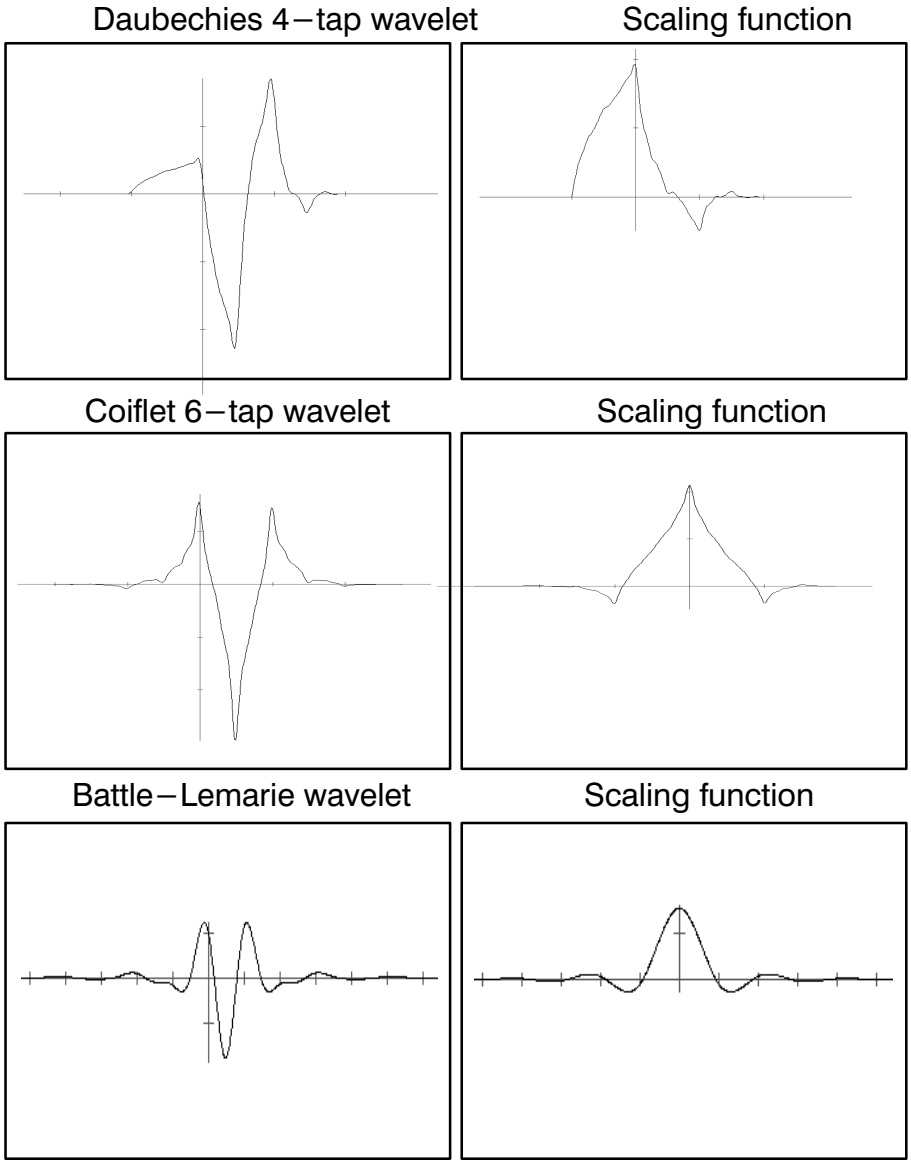

Fig. 3 Different wavelet types and their one–dimensional representatives

We should also note that for continuous approximations we face the competiting criteria of providing strict compact support and smooth symmetric shapes along with orthonormality in order to achieve a perfect reconstuction and compact coding.

3 Continuous Approximations using Wavelet Bases

3.1 The Volume Rendering Integral

From the sections above we know, that a 3D wavelet decomposition of a volume data set results in a compact pyramidal representation by the coefficients $\tilde{c}_j \in \left\{c^m_{pqr},\ d^{m,1}_{pqr},\ d^{m,2}_{pqr},\ d^{m,3}_{pqr},\ d^{m,4}_{pqr},\ d^{m,5}_{pqr},\ d^{m,6}_{pqr},\ d^{m,7}_{pqr}\right\}$. Let $w_j(x,y,z)$ be the wavelet $\psi^{m,l}_{pqr}$ or scaling basis function ϕ^m_{pqr}, the expansion of the volume density function $f(x,y,z)$ yields

$$f(x,y,z) = \sum_m \sum_p \sum_q \sum_r (c^m_{pqr}\phi^m_{pqr} + d^{m,1}_{pqr}\psi^{m,1}_{pqr} + d^{m,2}_{pqr}\psi^{m,2}_{pqr} + d^{m,3}_{pqr}\psi^{m,3}_{pqr} + d^{m,4}_{pqr}\psi^{m,4}_{pqr} + d^{m,5}_{pqr}\psi^{m,5}_{pqr} + d^{m,6}_{pqr}\psi^{m,6}_{pqr} + d^{m,7}_{pqr}\psi^{m,7}_{pqr}) = \sum_{j=1}^{N^3} \tilde{c}_j \cdot w_j(x,y,z) \quad (24)$$

where N represents the voxel resolution.

Once the basis functions are given continuously, eq. (24) provides an continuous approximation of the density function f with the constraint of perfectly reconstructing the discrete voxel values.

Due to the immense number of basis functions (for $128^3 > 2$ Mio.) we are forced to define a significance measure for the data to reject unimportant coefficients. In [20] a generic significance is proposed as $8^m|\tilde{c}_j|$ but when carefully analyzing the scaling properties on multidimensional WTs, we found that $2^{s/2+3m/2}|\tilde{c}_j|$ addresses the problem better. s represents the number of 1D scaling functions from which the respective 3D basis $w_j(x,y,z)$ is composed. Even more than this, we can also compute the local signal energy E_j from the Norm $\| \cdot \|_{L^2}$ as

$$E_j = \int_{\infty}^{\infty} \int_{\infty}^{\infty} \int_{\infty}^{\infty} \left| \tilde{c}_j \; w_j(x,y,z) \right|^2 dxdydz = |\tilde{c}_j| \cdot \| \, w_j(x,y,z) \, \|^2 \quad (25)$$

for each component [18]. Supposing the basis functions to hold

$$\left\langle w_j, w_j \right\rangle = 1 \quad (26)$$

we obtain the local energy by the square of the corresponding coefficients

$$E_j = \left(\tilde{c}_j \right)^2 \quad (27)$$

These equations allow us to estimate the error bounds of our approximation when rejecting unimportant coefficients and renders an alternate method to define a significance based on energy rather than on amplitude measures. Once the coefficients are filtered we obtain a modified approximation $\hat{f}(x,y,z)$ as

$$\hat{f}(x,y,z) = \sum_{j=1}^{K} \hat{c}_j \cdot \hat{w}_j(x,y,z) \quad (28)$$

In chapter 2 we elaborated that some wavelet types are not given in a closed form solution and properties of smoothness, symmetry and the keeping of the orthonormality can only be achieved by infinite support. For this end we have to cut off some wavelets on the one side and we have to find a generic interpolation scheme on the other side. In order to compare different wavelet types to each other and to be independent

of the basis function, we require a generic continuous representation scheme. Furthermore, the final goal is to find an approximate and analytic solution for the ray intensity function. Thus the discrete values of the wavelet functions obtained by iteration [4] are interpolated by piecewise polynomials using cubic splines in each intervall $I \subseteq \mathbb{R}^3$.

3.2 A Ray–Casting Method in Wavelet Space

The separability of the bases $\hat{w}_j$ allows us to write eq. (28) as

$$\hat{f}(x,y,z) = \sum_{j=1}^{K} \hat{c}_j \; b_j^1(x) \; b_j^2(y) \; b_j^3(z) \tag{29}$$

where the b_j represent the x,y and z–components of the basis functions.

The image generation with ray casting turns out to be a parametrization of the ray as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} \alpha_x t + \beta_x \\ \alpha_y t + \beta_y \\ \alpha_z t + \beta_z \end{bmatrix} \qquad \alpha := \left[\alpha_x, \alpha_y, \alpha_z\right]^T, \quad \beta := \left[\beta_x, \beta_y, \beta_z\right]^{T,} \tag{30}$$

α and β are the viewing direction and the eyepoint respectively.

We obtain the intensity function along the ray with

$$\hat{f}(t) := \hat{f} \begin{bmatrix} \alpha_x t + \beta_x \\ \alpha_y t + \beta_y \\ \alpha_z t + \beta_z \end{bmatrix} = \sum_{j=1}^{K} \hat{c}_j \cdot b_j^1(\alpha_x t + \beta_x) \cdot b_j^2(\alpha_y t + \beta_y) \cdot b_j^3(\alpha_z t + \beta_z) \tag{31}$$

This scheme is illustrated in fig. 4. The ray intensity function is provided by projecting the single basis functions onto the ray and by superimposing them. It is accomplished by scaling with α and by translating with β. Due to the piecewise spline interpolation we get now a continuous approximation of the ray intensity function. In each intervall $\left[t_n^j, t_{n+1}^j\right]$ and for each component of the resulting vector the polynomials $b_j^{i,n}(t)$ are given for each wavelet w_j as

$$b_j^{i,n}(t) = \sum_{k=0}^{Grad(b_j^{i,n})} \tilde{a}_k^{i,n} \cdot t^k \tag{32}$$

and their coefficients $\tilde{a}_k^{i,n}$ as

$$\tilde{a}_{k,j}^{i,n} = \sum_{l=k}^{Grad(b_j^{i,n})} \binom{l}{l-k} a_{l,j}^{i,n} \cdot \alpha^{\,k} \cdot \beta^{\,l-k} \tag{33}$$

$a_l^{i,n}$: Spline coefficients in intervall $[t_n, t_{n+1}]$

Thus we can write

$$b_j^{i,n}(t) = \sum_{k=0}^{Grad(b_j^{i,n})} \left[\sum_{l=k}^{Grad(b_j^{i,n})} \binom{l}{l-k} a_l^{i,n} \cdot \alpha^{\,k} \cdot \beta^{\,l-k} \right] t^{\,k} \quad with \;\; t \in [t_n^j, t_{n+1}^j] \tag{34}$$

The final expression for $\hat{f}(t)$ is obtained by

$$\begin{aligned} \hat{f}(t) = \sum_{j=1}^{K} \hat{c}_j & \left[\sum_{k=0}^{Grad(b_j^1)} \left[\sum_{l=k}^{Grad(b_j^1)} \binom{l}{l-k} a_l^{1,n} \, \alpha_x^{\,k} \cdot \beta_x^{\,l-k} \right] t^{\,k} \right] \\ & \cdot \left[\sum_{k=0}^{Grad(b_j^2)} \left[\sum_{l=k}^{Grad(b_j^2)} \binom{l}{l-k} a_l^{2,n} \, \alpha_y^{\,k} \cdot \beta_y^{\,l-k} \right] t^{\,k} \right] \quad t \in [t_n^j, t_{n+1}^j] \\ & \cdot \left[\sum_{k=0}^{Grad(b_j^3)} \left[\sum_{l=k}^{Grad(b_j^3)} \binom{l}{l-k} a_l^{3,n} \, \alpha_z^{\,k} \cdot \beta_z^{\,l-k} \right] t^{\,k} \right] \end{aligned} \tag{35}$$

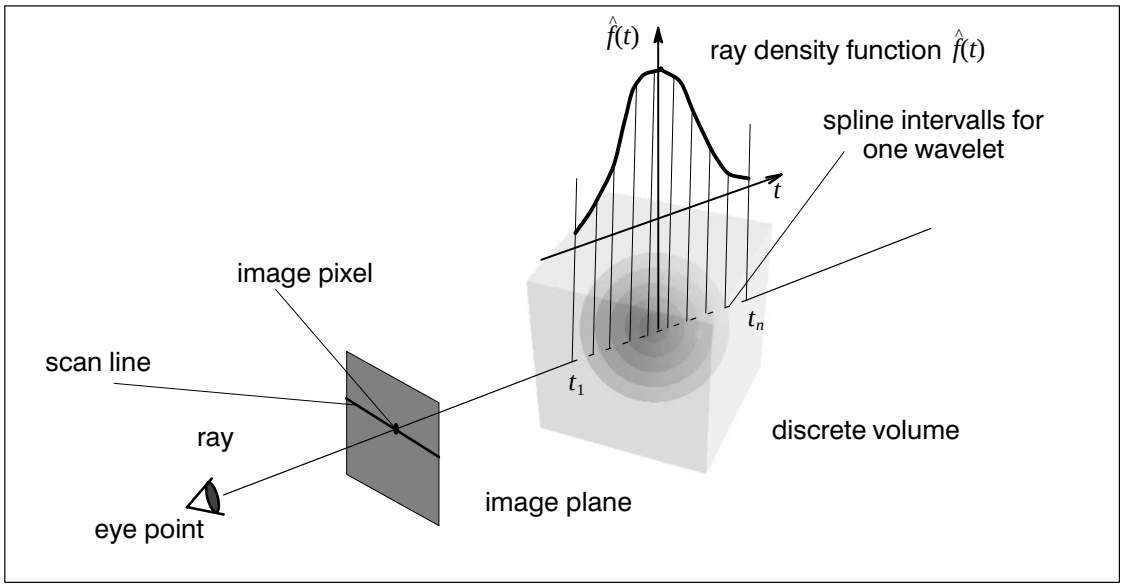

Fig. 4 Illustration of the rendering process

Eq. (35) stresses that we have a polynomial representation that can be integrated straightforwardly. Unfortunately the rendering equation incorporates an exponential absorption term.

Now, depending on the application we focus at, there are different ways to get the image intensity *I*. In this simple case, we do not consider self–occlusion, and eq. (1) simplifies to

$$I = \int_{t_1}^{t_2} f(t) \; dt \tag{36}$$

We obtain a piecewise analytic solution for each intervall $\left[t_n^j, t_{n+1}^j\right]$ by the piecewise primitive functions $W_j^n(t) = \int b_j^1(t) \cdot b_j^2(t) \cdot b_j^3(t)\ dt$, with

$$I = \sum_{j=1}^{K} I_j = \sum_{j=1}^{K} \hat{c}_j \sum_{n=0}^{L-1} \left[W_j^n(t_{n+1}^j) - W_j^n(t_n^j) \right] \qquad \begin{matrix} L^j: \text{ number of spline intervalls} \\ \text{for wavelet } w_j \end{matrix} \tag{37}$$

Note, that the size of the intervalls depend either on how the viewing ray intersects the wavelet and on the step size *m* of the pyramids. In the case of a close form representation of the $W_j(t)$ it simplifies to

$$I = \sum_{j=1}^{K} I_j = \sum_{j=1}^{K} \hat{c}_j \left[W_j(t_2) - W_j(t_1) \right] \tag{38}$$

In order to include a self–occlusion term and to evaluate the inner integral, we set for instance the function $\alpha(s) = \hat{f}(s)$ and obtain the following expression:

$$e^{-\int_{t_1}^{t} \hat{f}(s)ds} = e^{-\sum_{j=1}^{K} \hat{c}_j \sum_{n=0}^{L^j(t)-1} \left[W_j^n(t_{n+1}^j) - W_j^n(t_n^j) \right]} \qquad \begin{matrix} \forall j\ ,\ t_0^j = t_1 \\ t_{L^j(t)}^j = t \end{matrix} \tag{39}$$

Due to the local support of the WT, we have only to account a subset of wavelets along the ray. A linear approximation of the exponential function aligned to the spline intervalls yields

$$e^{-\int_{t_1}^{t} \hat{f}(s)ds} \approx \prod_{j=1}^{K} \hat{c}_j \prod_{n=0}^{L^j(t)-1} \left(1 - \left[W_j^n(t_{n+1}^j) - W_j^n(t_n^j) \right] \right) \tag{40}$$

Again, it has to be noted that a closed form representation provides a compact approximation, but a larger linearization error.

$$e^{-\int_{t_1}^{t} \hat{f}(s)ds} = \prod_{j=1}^{K} \hat{c}_j \left(1 - \left[W_j(t_2) - W_j(t_1) \right] \right) \tag{41}$$

The final discrete solution of the rendering equation depends on the shading model, but we can follow the principles of eq. (2):

$$I = \sum_{k=1}^{M} C_k \hat{f}_k \prod_{j=1}^{k} \hat{c}_j \prod_{n=0}^{L^j(k)-1} \left(1 - \left[W_j(t^j_{n+1}) - W_j(t^j_n)\right]\right) \tag{42}$$

Equation (42) provides then an approximate, analytic solution of the inner product. The gradient $\nabla f(x,y,z)$ that is required for shading can be computed easily from

$$\nabla \hat{f}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{j=1}^{K} \hat{c}_j \cdot \left(\frac{d}{dx} b_j^1(x) \cdot b_j^2(y) \cdot b_j^3(z)\ ,\ b_j^1(x) \cdot \frac{d}{dy} b_j^2(y) \cdot b_j^3(z)\ ,\ b_j^1(x) \cdot b_j^2(y) \cdot \frac{d}{dz} b_j^3(z)\right) \tag{43}$$

We have to mention here, that the implementation of these equations is not straightforeward. Although the picewise spline interpolation of the basis functions has only to be performed for the 1D–prototypes of ψ and ϕ, the calculation of the spline intervalls in t are computationally very expensive.

The problem of getting isosurfaces in the images will be treated separately and discussed in the next section. The pictures in this paper were based on eq. (36).

3.3 Examples

The pictures in fig. 5 and fig. 6 show the effects of continuous wavelet–based approximations for volume rendering and allow one to compare the influence of the shape of the basis function with the final image quality. In fig. 5 the surface of a simple voxelized model obtained by different basis functions is represented. The results are perfect reconstructions of the initial binarized volume intensities using only basis functions of the iteration m=0. We can cleary identify how the final approximation of the volume surface is influenced by the smoothness of the wavelet. The respective filter–coefficients are given in appendix A.

Fig. 5 Perfect reconstructions of a simple voxel object rendered an opaque surface with different wavelets of the iteration m=1
a) Daubechies wavelet, 4–tap (N=2)
b) Coiflet wavelet, 6–tap
b) Battle–Lemarie wavelet

In fig. 6 a Gaussian density distribution was voxelized at a resolution of 32^3 and rendered with different numbers of coefficients and iteration depths. The isosurface was set to $\tau = 0.5$ and surrounding it, the intensities below that threshold are represented as a bluish translucency. We should note that symmetry and shape of the wavelet strongly influence the shape of the isosurface. Assymetric and fractallike functions, as for instance the Daubechies wavelet, generate artifacts, such as rips or modulations of the translucency. It is interesting to compare them to the shapes obtained by a standard marching cubes on the initial volume data set.

Fig. 6 Images obtained from a Gaussian density distribution of 32^3 voxels:
(a) Isosurface at $\tau = 0.5$ with a marching cubes algorithm

b,c,d) Isosurfaces and translucent hull obtained from a Battle–Lemarie wavelet with
(b) 151
(c) 345
(d) 961 coefficients

e,f,g) Isosurfaces and translucent hull obtained from a Coiflet wavelet with
(e) 82,
(f) 345
(g) 1006 coefficients

h,i,j) Isosurfaces and translucent hull obtained from a Daubechies wavelet with
(h) 206,
(i) 494
(j) 1154 coefficients

The pictures were generated with our hybrid renderer that composes the volumetric intensities with the isosurfaces. In these examples, the isosurfaces were rendered using the implicit function approach of chapter 4 [12]. The coefficients were filtered according to the significance measures in section 3.1.

4 Rendering of Isosurfaces in Wavelet Spaces

4.1 Implicit Function Approach versus Marching Cubes

As we mentioned, the wavelet decomposition opens up different ways to get isosurfaces from volume data. Direct ways of volume integration to get opaque surfaces, as proposed by [5] or [16] usually base on shading models that estimate the surface normal by means of the volume gradient ∇f. They provide good–looking images but do not exactly solve the illumination equation. In these cases, the computation of the gradient follows eq. (43) in section 3.2. Another way to solve the isosurface problem is proposed by [21]. It bases on the idea, that isosurfaces of a threshold τ satisfy

$$\hat{f}(x,y,z) = \tau \;\Rightarrow\; \hat{f}(x,y,z) - \tau = 0 \tag{44}$$

in the continuous approximation of eq. (28).

Equation (44) leads to an implicit function that can be rendered using methods as in [12]. Since we deal with polynomials, all required terms can be computed analytically, as for instance the Lipschitz condition

$$
\begin{aligned}
\mathcal{L} &\geq \max_{\mathcal{R}} \left|\nabla \hat{f}(x,y,z)\right| \\
\mathcal{L} &:= \sum_{j=1}^{K} \left|\hat{c}_j\right| \cdot \left(\max_{[x_{min},x_{max}]} \left|\frac{d}{dx} b_j^1(x)\right| \cdot \max_{[y_{min},y_{max}]} \left|b_j^2(y)\right| \cdot \max_{[z_{min},z_{max}]} \left|b_j^3(z)\right| \right. \\
&\quad + \max_{[x_{min},x_{\max}]} \left|b_j^1(x)\right| \cdot \max_{[y_{min},y_{\max}]} \left|\frac{d}{dy} b_j^2(y)\right| \cdot \max_{[z_{min},z_{\max}]} \left|b_j^3(z)\right| \\
&\quad \left. + \max_{[x_{min},x_{\max}]} \left|b_j^1(x)\right| \cdot \max_{[y_{min},y_{\max}]} \left|b_j^2(y)\right| \cdot \max_{[z_{min},z_{\max}]} \left|\frac{d}{dz} b_j^3(z)\right| \right)
\end{aligned}
\tag{45}
$$

and the gradient function *g(t)* along the ray

$$
g(t) := \frac{d}{dt} \hat{f}(t) \tag{46}
$$

The pictures of fig. 6 were generated using this method. Unfortunately, there are several shortcomings in this approach: Due to the huge number of basis functions, Kalra's method becomes extremely time consuming. Furthermore, the appearance of the isofurface is strongly influenced by the shape of the wavelet, that has to be represented in a continuous form. Hence, it becomes interesting to apply a simple marching cubes technique [17] on the back–transformed data set and to compose the polygons with the volume data during the rendering process. In these cases, the isosurface generation is no longer performed in wavelet space, but the local data quality is still controlled by the wavelets.

4.2 Examples

Figure 7a shows the bust of Johann Strauss, as it is derived from a 3D laserscanner. This model was illuminated and voxelized with a resolution of 32^2x64 voxels. The isosurfaces obtained from a marching cubes are presented for a flat and for a smoothly shaded reconstruction in figs. 7b and 7c respectively. Figures 8a,c,b show the isosurface reconstruction with Kalra's method [12] that is accomplished immediately on using the basis functions in wavelet space. In figure 9a,b,c the same reconstructions are presented after an inverse transform of the filtered data and with a marching cubes algorithm. In both cases, an increasing number of coefficients were employed to encode the data. We can clearly recognize that the level of detail increases as the number of wavelets is raised. In figures 8a and 9a only the low–pass coefficients represent the data and therefore the fine grain details of the surface shape included in the higher frequencies do not appear at all. With an increasing number of coefficients, more and more surface details become visible. In order to compare the influence of the wavelet, the same situation is presented for Coiflet bases in fig. 10 and in fig. 11. Similar to our Gaussians, the smoothness of the isosurface strongly depends on the shape of the wavelet. Hence, the Coiflet wavelet renders "rougher" shapes. The local properties of the WT are demonstrated with the pictures in fig. 12. The shapes were rendered with a spatially varying level–of–detail. The top of the head is perfectly reconstructed, whereas in the lower part of the bust only the scaling

functions were employed to encode the data. Figure 12a shows the original and 12b, c and d local reconstructions with Haar–, Coiflet and Battle–Lemarie wavelets, respectively.

Fig. 7 a) Range data of a bust of Johann Strauss
b) Voxelized volume data at a resolution $32^2 \times 64$
c) Isosurface reconstruction using a marching cubes algorithm

Fig. 8 Isosurface reconstruction using Kalra's method and 3D–wavelets with τ=0.5
a) 1180 coefficients, m = 3 only
b) 2832 coefficients, m = 2 only
c) 9601 coefficients, m = 1 only

Fig. 9 Isosurface reconstruction using marching cubes methods with τ=0.5 and Battle–Lemarie–wavelets
a) 1180 coefficients, m = 3 only
b) 2832 coefficients, m = 2 only
c) 9601 coefficients, m = 1 only

Fig. 10 Isosurfaces obtained from Coiflet–wavelets and Kalra's method
a) 1397 coefficients, m = 3 only
b) 2422 coefficients, m = 2 only
c) 13177 coefficients, m = 1 only

Fig. 11 Isosurfaces obtained from Coiflet–wavelets and marching cubes method
a) 1397 coefficients, m = 3 only
b) 2422 coefficients, m = 2 only
c) 13177 coefficients, m = 1 only

Fig. 12 Spatially varying reconstruction using different kinds of wavelets. The upper part is perfectly reconstructed, whereas the lower was filtered by our significance filter, described in 3.1. after a decomposition of depth m=2
a) Original
b) Haar wavelets
c) Coiflet wavelets
d) Battle–Lemarie wavelets

Fig. 13 Isosurfaces from a volume that was decomposed, filtered and reconstructed by using a Battle–Lemarie wavelet for different thresholds (2832 coeff, marching cubes)
a) τ = 0.1
b) τ = 0.3
c) τ = 0.5
d) τ = 0.8

Finally, fig. 13a,b,c, expresses the influence of the threshold τ defined for the isosurface. Since the inverse WT provides floating point values normalized between 0 and 1 the shape obtained from the marching cubes depends also on the selected isosurface value. Too small values give rise to some holes in the reconstruction (see fig. 13a). All other pictures were rendered using intermediate values for τ = 0.5. With respect to the computational costs, the marching cubes method has to be favourable to the expensive Kalra's method although the latter provides the accurate solution for a particular isosurface.

5 Integration of Data Analysis and Visualization

5.1 The Vision

In the above sections most emphasis was placed on the mathematical analysis of wavelet–based volume and isosurface rendering. The WT however, has sucessfully been used for data feature extraction as well [28],[14],[3],[8]. In particular in the field of image processing, several approaches to texture analysis have

been proposed so far. Most of these techniques have been performed by using a local WT on single coherent texture samples. In order to overcome this restriction, we proposed a new concept for texture feature extraction in images based on a global WT [8]. The features are derived from the local wavelet coefficients and take advantage of the localization properties of the WT. These results motivated us to extend the segmentation scheme to 3D and to embed it into the rendering process – all that in the underlying data space of the WT. Figure 14 illustrates again our pipeline, where one global WT is first performed on the initial data set.

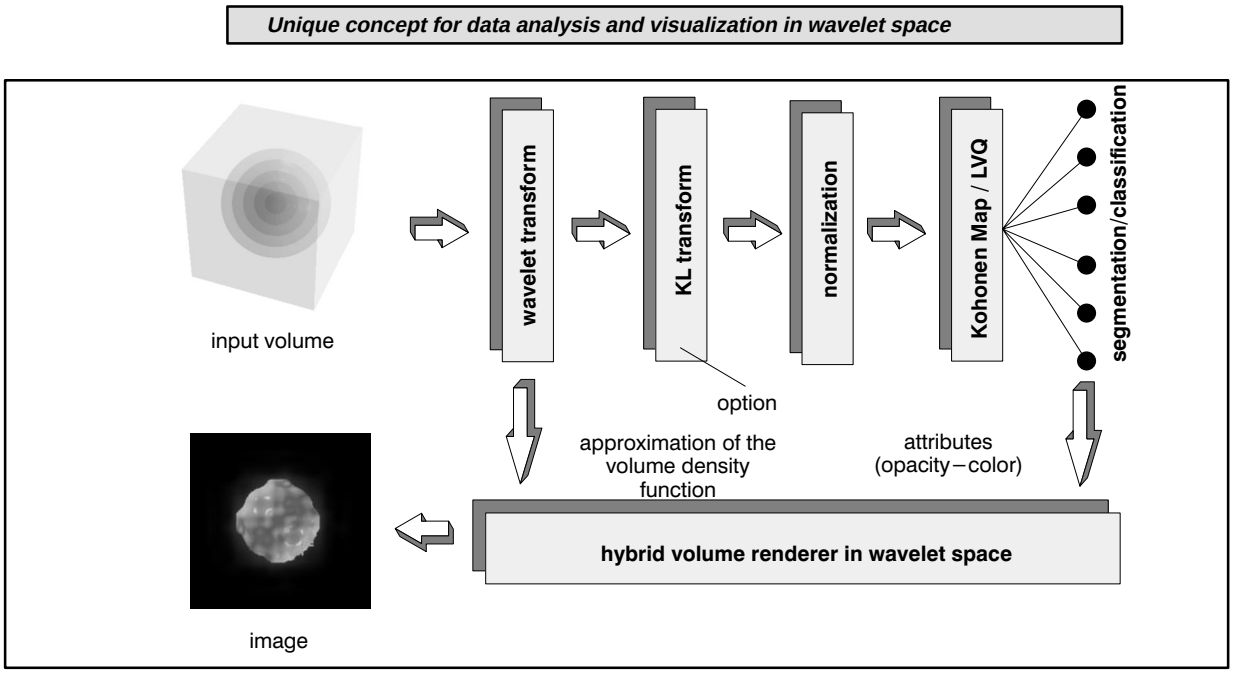

Fig. 14 Framework for integrated volume rendering and data analysis in wavelet spaces

Once the data is transformed, the segmentation pipeline associates a segmentation to all voxels in the data set. Therefore, we first extract a wavelet stream of coefficients from the wavelet pyramide for each voxel. This vector is taken as the feature vector describing the local properties of the data surrounding the voxel. These data have to be decorrelated, because our extraction scheme renders slightly correlated features, in spite of the orthonormality of the transform. After normalizing the features, they are feed into a neural network [7] that accompishes clustering and classification. The result from this process is a segmentation map, that can be imported into the volume renderer.

5.2 Including Segmentation Results

There are different ways to incorporate segmentation results into the rendering process. [16], for instance used an α–map to enhance local details. [5] choosed a probabilistic model and distributed the percentage of material to each pixel.

From the pictures above we know that it is possible to render opaque voxel object surfaces using appropriate isosurface techniques. In particular fig. 5 corresponds to a splatting of the data with wavelet basis func-

tions. Hence we could splat the respective voxels with basis functions of different iterations. The general description of the method can be obtained as

$$\tilde{f}(x,y,z) = \sum_{i,j,k \in \text{subvolume}} b_{i,j,k}(x,y,z) \tag{47}$$

where $\tilde{f}(x,y,z)$ is the subvolume to be splatted. The single splatting functions are defined as shifted versions of one prototype *b(x,y,z)*.

$$b_{i,j,k}(x,y,z) := b(x-i, y-j, z-k) \tag{48}$$

If we choose radial basis functions of the type

$$b(x,y,z) := b^1(x)^2 + b^2(y)^2 + b^3(z)^2 \tag{49}$$

the single voxels will be respresented as small spherical objects. This is illustrated for an experimental situation in fig. 15. In this picture, we employed cubic basis functions. The coherance of the resulting shape depends on the support of the splatting function and on the threshold choosen for the isosurface.

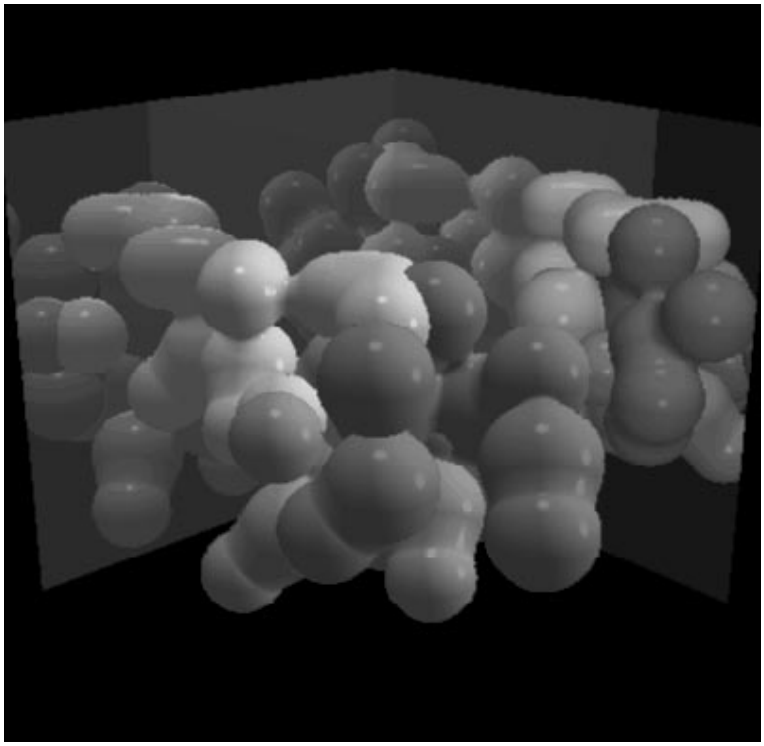

Fig. 15 Splatting of spherical basis functions to illustrate the possibility to encode local data properties in terms of color.

6 Conclusion

In this paper, we proposed a new method to get approximate solutions of the low–albedo volume rendering equation in wavelet spaces. We proposed different techniques to render both isosurfaces and translucent volumes using piecewise cubic splines to interpolate the wavelet function. The framework of the wavelet transform offers an elegant way to combine localisation and level–of–detail. Furthermore this method provides piecewise analytic solutions of the intensity integral, since the underlaying volume is approximated continuously by polynomials. The quality of the results depends strongly on the types of the selected basis function. Yet, it turns out, that the projection of the bases onto the ray is computationally expensive. Hence we have to find a wavelet, that is smooth, of strict finite support, orthonormal and that provides a closed–form integral in t. These properties account for both rendering and data analysis, but cannot be satisfied in common. Thus a compromise has to be found.

7 Literature

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

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

[3] Chang, T.; Kuo, C.: "Texture Analysis and Classification with Tree–Structured Wavelet Transform". IEEE Transactions on Image Processing, Vol. 2, No. 4, pp. 429 – 441, 1993

[4] Daubechies, I.: "The Wavelet Transform, Time–Frequency Localization and Signal Analysis". IEEE Transactions on Information Theory, Vol. 36, pp. 961 – 1005, 1990

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

[6] Groß, M. : "Visual Computing". Springer–Verlag, 1994

[7] Groß, M.; Seibert, F.: "Visualization of Multidimensional Image Data Sets using a Neural Network". The Visual Computer, Vol. 10, pp. 145 – 159, 1993

[8] Groß, M.; Koch, R; Lippert, L.; Dreger, A.: "Multiscale Image Texture Analysis in Wavelet Space". IEEE–ICIP'94, to appear 1994

[9] Grossmann, A.; Morlet, J.: "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape". SIAM Journal of Mathematical Analysis, Vol. 15, pp. 723–736, 1984

[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] Kalra, D.; Barr, A.H.: "Guaranteed Ray Intersection with Implicit Surfaces". Computer Graphics (Proc. SIGGRAPH '89), Vol. 23, pp. 297–306, 1989

[13] Krueger, W.: "The Application of Transport Theory to Visualization of 3D Scalar Data Fields". IEEE, pp. 273–279, 1990

[14] Laine, A.; Fan, J.: "Texture Classification by Wavelet Packet Signatures". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1186 – 1191, 1993

[15] 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

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

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

[18] Mallat, S.: "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, pp. 674–693, July 1989

[19] Meinzer, H.P.; Meetz, K.; Scheppelmann, D.; Engelmann, U.; Baur, H.J.: "The Heidelberg Ray Tracing Model". IEEE Computer Graphics and Application, Nov. , pp. 34–43, 1992

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

[21] Muraki, S.: "Approximationand Rendering of Volume Data Using Wavelet Transform". IEEE Computer Graphics and Application, Vol. 9, pp. 21–28, 1992

[22] Muraki, S.: "Multiscale 3D Edge Representation of Volume Data by a DOG Wavelet". Symposium on Volume Visualization, to appear 1994

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

[24] Sabella, P.: "A Rendering Algorithm for Visualizing 3D Scalar Fields". Computer Graphics, Vol. 22, No.4, pp. 160–164, 1988

[25] Sarkas, G.; Gerth, M.: "Sampling and Anti–Aliasing of Discrete 3D Volume Density Textures". Eurographics, Vienna, Sep., pp. 87–102, 1991

[26] Simoncelli, E. P.; Adelson, E.H.: "Non–Separable Extensions of Quadrature Mirror Filters to Multiple Dimensions". Proceedings of the IEEE, Vol. 78, No. 4, pp. 652–664, 1990

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

[28] 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

[29] Vetterli, M.: "Wavelets and Filter Banks: Theory and Design". IEEE Transactions on Signal Proc., Vol. 40, pp. 2207 – 2231, 1992

8 Appendix A

Discrete filter coefficients *h(n)*

n	Daubechies 4–tap	Coiflet 6–tap	Battle–Lemarie
–14	0.0	0.0	–.01103739
–13	0.0	0.0	0.001882134
–12	0.0	0.0	0.002186714
–11	0.0	0.0	–.003882426
–10	0.0	0.0	–.004353840
–9	0.0	0.0	0.008201477
–8	0.0	0.0	0.008685294
–7	0.0	0.0	–.017982291
–6	0.0	0.0	–.017176331
–5	0.0	0.0	0.042068328
–4	0.0	0.0	0.032080869
–3	0.0	0.0	–.110036987
–2	0.0	0.0	–.050201753
–1	0.0	0.0	0.433923147
0	0.482962913145	–.072732619494	0.766130398
1	0.836516303738	0.337897662369	0.433923147
2	0.22414386 8042	0.852572019987	–.050201753
3	–.129409522551	0.384864846763	–.110036987
4	0.0	–.072732965094	0.032080869
5	0.0	–.015655728131	0.042068328
6	0.0	0.0	–.017176331
7	0.0	0.0	–.017982291
8	0.0	0.0	0.008685294
9	0.0	0.0	0.008201477
10	0.0	0.0	–.004353840
11	0.0	0.0	–.003882426
12	0.0	0.0	0.002186714
13	0.0	0.0	0.001882134
14	0.0	0.0	–.01103739