WAVELET-BASED VOLUME VISUALIZATION

JOS B.T.M. ROERDINK AND MICHEL A. WESTENBERG

INSTITUTE FOR MATHEMATICS AND COMPUTING SCIENCE
UNIVERSITY OF GRONINGEN
P.O. BOX 800, 9700 AV GRONINGEN, THE NETHERLANDS
EMAIL: ROE@CS.RUG.NL,MICHEL@CS.RUG.NL

ABSTRACT. We consider multiresolution visualization of large volume data sets based on wavelets. Starting from a wavelet decomposition of the data, a low resolution image is computed; this approximation can be successively refined. The practical need for such a multiresolution approach is motivated. The mathematical techniques involve Fourier and wavelet analysis.

1. BACKGROUND AND MOTIVATION

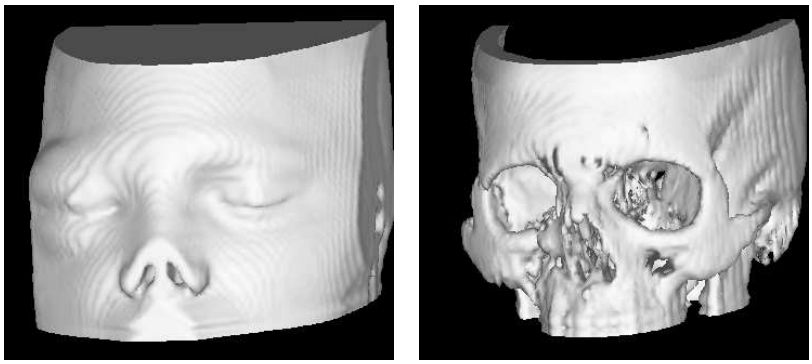
There is a growing interest in the visualization and exploration of digital data representing large three-dimensional (3D) volumes, which can result from measurement by physical equipment, or from computer simulation. Such data arise in many scientific areas, such as computerized tomography (CT), astronomy, computational physics or chemistry, fluid dynamics, seismology, environmental research, non-destructive testing, etc. Since it is very hard to interpret the interior of volumes by viewing individual slices, volume visualization techniques have been developed for viewing these data from different viewpoints, using advanced computer graphics techniques such as illumination, shading and colour.

With the increasing popularity of the Internet and the data exchange through systems such as the World Wide Web, fast and efficient methods for transfer and display of 3D data are required. To relieve the demand on the server capacity, there is a tendency for transmitting the data first to the client's workstation on which the visualization of the data takes place. Due to their large size the transmission of the data sets is time consuming. This motivates the need for visualizing the data *incrementally* as it arrives ('progressive refinement') in order to achieve a reasonable response time for the user. For this purpose multiresolution models are developed, allowing the systematic decomposition of the data into versions at different levels of resolution. Other benefits of such approaches are local *level-of-detail* (LOD), i.e. using a lower resolution for small, distant or unimportant parts of the data. Wavelets are a natural candidate for such a multiresolution approach.

1.1. **Surface rendering.** Although not the subject of this paper, we first mention the class of volume visualization methods known as *surface rendering*, where one reduces the volume to a number of surfaces representing the boundary between materials [5, 7]. These are usually defined as iso-surfaces $S(c)$ of a density function $f(x, y, z)$, given by an equation $f(x, y, z) = c$, where c is a constant. The information available about this function f is the 3D voxel array of data values $f_{k,l,m} = f(k\Delta_x, l\Delta_y, m\Delta_z), k = 1, \ldots N_1, l = 1, \ldots N_2, m = 1, \ldots N_3$, with $\Delta_x, \Delta_y, \Delta_z$ the sampling step sizes in the three spatial directions. By taking different values for the constant c, distinct surfaces can be extracted from the data. Figure 1 shows an example of two iso-surfaces reconstructed from CT data of a human head, corresponding to skin and bone, respectively.

Key words and phrases. Direct volume visualization, X-ray transform, Fourier slice theorem, wavelet splatting, wavelet X-ray transform.

To appear in: Proc. Dutch Mathematical Congress, 1998.



(a) (b)

Figure 1. *Surface rendering of human head CT data, corresponding to (a) skin and (b) skull.*

Characteristic of surface rendering methods is the use of intermediate graphical primitives, such as triangle meshes. The size of these meshes may become very large in real world applications such as CAD models, digital terrains or medical volume data. Therefore there is currently a very active interest in developing techniques for *mesh simplification* [11], especially for real-time, interactive environments. Wavelet-based techniques are among the collection of recently proposed methods [8].

1.2. **Direct volume rendering.** A different approach to volume visualization, called *direct volume rendering* [4], does not make use of intermediate graphical primitives, but tries to map the information in the 3D data set directly on the view plane.

A standard method, called *X-ray volume rendering*, is to integrate the function f along the line of sight. This turns out to be one of the preferred techniques for medical applications, because physicians are well-trained in interpreting such X-ray like images for diagnosis. The corresponding mathematical concept is the *X-ray transform*, well-known from computerized tomography [10]. Consider the line integrals of a square integrable function $f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^3$, along a direction vector $\boldsymbol{\theta}$. Let $\mathbf{u}$ and $\mathbf{v}$ be two orthogonal vectors in the plane perpendicular to $\boldsymbol{\theta}$, cf. Fig. 2. The X-ray transform of f is defined by

$$\mathcal{P}_{\boldsymbol{\theta}} f(u,v) = \int_{\mathbb{R}} f(u\mathbf{u} + v\mathbf{v} + t\boldsymbol{\theta})\, \mathrm{d}t\,. \tag{1.1}$$

In the remainder of this paper, we describe two wavelet-based methods for direct volume rendering, which are both extensions of standard volume visualization techniques based on the X-ray transform.

2. Wavelet Splatting

The splatting algorithm [15] contains the following steps: (i) *Reconstruction* of a continuous function from the discrete data using convolution with a reconstruction filter; (ii) Mapping the continuous function to the image plane as a superposition of building blocks called *splats* or *footprints*. The reconstruction step is necessary because the lines parallel to a given view direction $\boldsymbol{\theta}$ passing through the pixels of the view plane do not necessarily pass through the grid points of the initial 3D data set. We next outline the wavelet splatting method, first proposed in [6]. The difference with the original splatting method is, that (i) *wavelets* are used as reconstruction filters, and (ii) it is possible to compute images at different levels of detail.

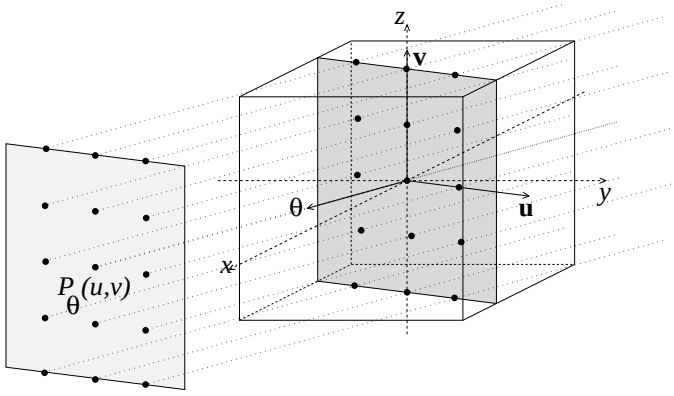


Figure 2. *Parameters of the X-ray transform.*

2.1. **Wavelet representation.** A 1D orthonormal wavelet basis can be constructed from a *scaling function* ϕ with associated wavelet ψ, and corresponding basis functions $\{\phi_{j,k}\}$ and $\{\psi_{j,k}\}$, $j,k \in \mathbb{Z}$, where $\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x - k)$ and $\psi_{j,k}(x)$ is defined similarly [1, 2, 13, 14]. Here j is the scale parameter and k the translation parameter. From this, a 3D separable orthonormal wavelet basis can be constructed with 8 basis functions, i.e. one scaling function $\Phi^0_{j,k,l,m}(x,y,z)$ and 7 wavelet basis functions $\Psi^\tau_{j,k,l,m}, \tau \in T = \{1,2,3,4,5,6,7\}$ defined as follows:

$$\begin{aligned}
\Phi^0_{j,k,l,m}(x,y,z) &= \phi_{j,k}(x)\phi_{j,l}(y)\phi_{j,m}(z), & \Psi^1_{j,k,l,m}(x,y,z) &= \phi_{j,k}(x)\phi_{j,l}(y)\psi_{j,m}(z)\\
\Psi^2_{j,k,l,m}(x,y,z) &= \phi_{j,k}(x)\psi_{j,l}(y)\phi_{j,m}(z), & \Psi^3_{j,k,l,m}(x,y,z) &= \phi_{j,k}(x)\psi_{j,l}(y)\psi_{j,m}(z)\\
\Psi^4_{j,k,l,m}(x,y,z) &= \psi_{j,k}(x)\phi_{j,l}(y)\phi_{j,m}(z), & \Psi^5_{j,k,l,m}(x,y,z) &= \psi_{j,k}(x)\phi_{j,l}(y)\psi_{j,m}(z)\\
\Psi^6_{j,k,l,m}(x,y,z) &= \psi_{j,k}(x)\psi_{j,l}(y)\phi_{j,m}(z), & \Psi^7_{j,k,l,m}(x,y,z) &= \psi_{j,k}(x)\psi_{j,l}(y)\psi_{j,m}(z).
\end{aligned}$$

Then f can be represented in the following way:

$$f(x,y,z) = \sum_{k,l,m} c^L_{k,l,m}\,\Phi^0_{L,k,l,m}(x,y,z) + \sum_{j=1}^{L}\sum_{\tau\in T}\sum_{k,l,m} d^{j,\tau}_{k,l,m}\,\Psi^\tau_{j,k,l,m}(x,y,z). \tag{2.1}$$

Here the integer L denotes the number of decomposition levels, $c^L_{k,l,m} = \langle f, \Phi^0_{L,k,l,m}\rangle$ are the *approximation coefficients* and $d^{j,\tau}_{k,l,m} = \langle f, \Psi^\tau_{j,k,l,m}\rangle$ the *detail coefficients*, respectively, where $\langle\cdot,\cdot\rangle$ denotes the inner product in the space $L^2(\mathbb{R}^3)$ of square integrable functions. These coefficients are computed by a discrete wavelet transform of the initial voxel array $f_{k,l,m}$ of dimension $M := N_1 \times N_2 \times N_3$ (in practice we assume each N_i to be a power of two). At scale j, the vector (k,l,m) of translation integers in (2.1) runs over a downsampled 3D array whose size is $8^{-j}\cdot M$. This representation is easily extended to a basis of *biorthogonal wavelets* [1–3], which is for example needed when using B-spline wavelets.

In practice the decomposition and reconstruction step are performed by the fast wavelet transform [1, 14], which operates on discrete data by down- or upsampling and convolution with discrete filters which are associated uniquely to the scaling and wavelet basis functions.

2.2. **Wavelet domain rendering.** In (1.1) replace f by its wavelet representation (2.1):

$$\begin{aligned}
\mathcal{P}_{\boldsymbol{\theta}} f(u,v) = &\sum_{k,l,m} c^L_{k,l,m} \int_{\mathbb{R}} \Phi^0_{L,k,l,m}(u\mathbf{u} + v\mathbf{v} + t\boldsymbol{\theta})\,\mathrm{d}t\\
&+ \sum_{j=1}^{L}\sum_{\tau\in T}\sum_{k,l,m} d^{j,\tau}_{k,l,m} \int_{\mathbb{R}} \Psi^\tau_{j,k,l,m}(u\mathbf{u} + v\mathbf{v} + t\boldsymbol{\theta})\,\mathrm{d}t\,.
\end{aligned} \tag{2.2}$$

This equation expresses $\mathcal{P}_{\boldsymbol{\theta}} f(u,v)$ as a weighted summation of 2D functions on the view plane (these are the *splats* or *footprints*). The integrals have to be evaluated only once for a given view direction at the coarsest scale $j = L$ for $(k,l,m) = (0,0,0)$, resulting in eight *prototype footprints*. It is possible to compute all required footprints for finer scales and/or other translation vectors from these prototypes by rescaling and shifting.

2.3. **Computing the prototype footprints.** An efficient way to compute the required integrals is based upon the *Fourier slice theorem* well-known from computerized tomography [10]. Denote the n-dimensional Fourier transform of a function $f \in L^2(\mathbb{R}^n)$ by $\mathcal{F}_n f$:

$$\mathcal{F}_n f(\vec{\omega}) = \int_{\mathbb{R}^n} e^{-2\pi i \vec{\omega}\cdot\vec{x}}\, f(\vec{x})\, \mathrm{d}\vec{x}\,, \qquad \vec{\omega} \in \mathbb{R}^n.$$

Then the 2D Fourier transform of the X-ray transform $\mathcal{P}_{\boldsymbol{\theta}} f(u,v)$ for fixed $\boldsymbol{\theta}$ is given by slicing the 3D Fourier transform $\mathcal{F}_3 f(\omega_x, \omega_y, \omega_z)$ with a plane (called the 'slice plane') through the origin and perpendicular to $\boldsymbol{\theta}$, cf. Fig. 2:

$$\mathcal{F}_2 \mathcal{P}_{\boldsymbol{\theta}} f(\omega_u, \omega_v) = \mathcal{F}_3 f(\omega_u \mathbf{u} + \omega_v \mathbf{v}). \tag{2.3}$$

A subsequent inverse 2D Fourier transform gives the desired footprints in the view plane.

When analytic expressions for the Fourier transform of the scaling function and wavelets exist, such as for the Haar and cardinal B-spline wavelets [2], no interpolation from discrete samples is necessary. A slice of Φ^0 or Ψ^τ is computed according to a discretization of the view plane satisfying the Nyquist sampling conditions.

2.4. **Wavelet splatting algorithm.** The algorithm consists of the following steps:

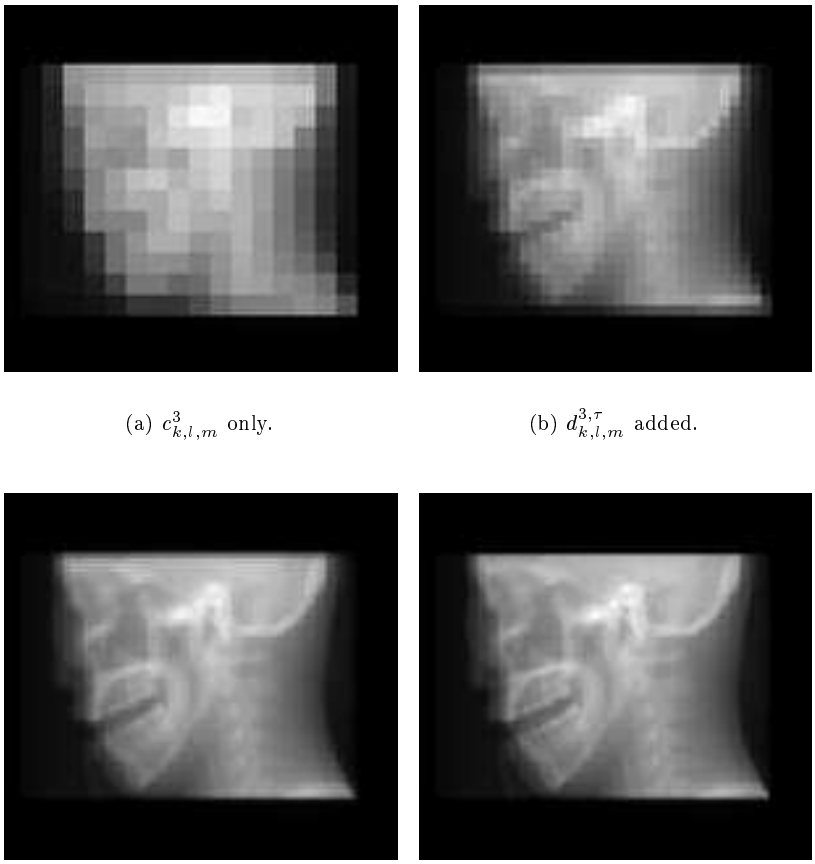
- *Pre-processing.* Perform a 3D wavelet transform (depth L) of the volume data.
- *Actual volume rendering.* For each direction $\boldsymbol{\theta}$, do:
  1. compute *prototype* footprints at level L in the view plane orthogonal to $\boldsymbol{\theta}$;
  2. from these, compute footprints for *lower levels* by scaling and downsampling;
  3. compute a *low resolution image* as a summation of scaled and translated footprints weighted by the approximation coefficients $c^L_{k,l,m}$;
  4. *refine image* successively by using detail coefficients $d^{j,\tau}_{k,l,m}$, $j = L, L-1, \ldots, 1$, yielding approximations at level $j = L-1, L-2, \ldots, 0$, respectively. The $j = 0$ approximation coincides (up to discretization errors) with the original function $\mathcal{P}_{\boldsymbol{\theta}} f$.

For each change in view direction, the whole process is repeated. In applications one would normally use a low resolution version when rotating the data set, and refine this when the rotation is stopped for detailed visual inspection.

2.5. **Example.** The algorithm was applied to a CT data set (size 128^3 voxels) of a human head which was decomposed by a three level ($L = 3$) wavelet decomposition. For illustration purposes, we first show the results when using the Haar wavelet, see Fig. 3. The image on the left is a low resolution approximation at level 3, where only 0.2% of the wavelet coefficients are used. This image was refined by adding detail information so that 1.56%, 12.5% and finally 100% of the wavelet coefficients are used. From the pictures in Fig. 3(b) it can be observed that the quality of the level-2 approximation is low, due to the discontinuous character of the Haar wavelet. So, in this case one has to use at least a level-1 approximation, cf. Fig. 3(c).

We also used a second order B-spline wavelet, which gives much smoother results at large compression ratios. In that case, we need a biorthogonal wavelet basis, defined by filter coefficients of unequal length for decomposition and reconstruction. For efficiency reasons, we use the long filter for the initial 3D wavelet decomposition (performed only once), and the short filter for the reconstruction at each view angle. The 1D filters out of which the 3D wavelet basis is constructed have length 41 and length 5, respectively. Results are shown in Fig. 4 for a level-2 decomposition. Only two levels are used in this case because, due to the downsampling operation, the support of the image in every spatial dimension is halved at each additional decomposition level. So for

$L = 3$ these support sizes would become smaller than the filter size, which does not yield useful results. In this case a level-2 approximation is quite acceptable, due to the smoothness of the second order B-spline wavelet, cf. Fig. 4(a).



(a) $c^3_{k,l,m}$ only. (b) $d^{3,\tau}_{k,l,m}$ added.

(c) $d^{2,\tau}_{k,l,m}$ added. (d) $d^{1,\tau}_{k,l,m}$ added.

Figure 3. *Reconstruction from a 3-level Haar wavelet decomposition of human head CT data.*

3. Fourier domain volume rendering

The splatting algorithm has time complexity $\mathcal{O}(N^3)$ for a $N \times N \times N$ data set. To achieve higher speed Fourier domain visualization techniques have been developed in the past, based upon the Fourier slice theorem [9]. After an initial 3D Fourier transform of the data, this algorithm runs in almost the same way as sketched above for the computation of footprints (Section 2.3). The only additional step is interpolation in Fourier space to obtain the values of the Fourier transform of the function f at a regular pixel grid in the slice plane from the values of the 3D discrete Fourier transform of the data. The time complexity is dominated by the 2D inverse Fourier transform from the slice plane to the view plane, hence is $\mathcal{O}(N^2 \log N)$. In practice, the interpolation step is the most time consuming operation.

We propose here a modification of this algorithm using wavelets allowing a multiresolution visualization with the same time complexity as ordinary Fourier domain rendering.

(a) $c^2_{k,l,m}$ only. (b) $d^{2,\tau}_{k,l,m}$ added. (c) $d^{1,\tau}_{k,l,m}$ added.

Figure 4. *Reconstruction from a 2-level B-spline wavelet decomposition of human head CT data.*

3.1. **Wavelet Representation of X-ray projections.** The idea is to directly expand the X-ray transform $\mathcal{P}_{\boldsymbol{\theta}} f$ in a 2D wavelet series as follows:

$$\mathcal{P}_{\boldsymbol{\theta}} f(u,v) = \sum_{k,l} c^L_{k,l}(\boldsymbol{\theta}) \Phi^0_{L,k,l}(u,v) + \sum_{j=1}^{L} \sum_{\tau \in T} \sum_{k,l} d^{j,\tau}_{k,l}(\boldsymbol{\theta}) \Psi^\tau_{j,k,l}(u,v), \tag{3.1}$$

with scaling function $\Phi^0_{j,k,l}(u,v) = \phi_{j,k}(u)\phi_{j,l}(v)$ and 3 wavelet basis functions $\Psi^\tau_{j,k,l}, \tau \in T = \{1,2,3\}$ constructed from the 1D wavelet basis functions as follows:

$$\begin{aligned} \Phi^0_{j,k,l}(x,y,z) &= \phi_{j,k}(x)\phi_{j,l}(y), \quad & \Psi^1_{j,k,l}(x,y,z) &= \phi_{j,k}(x)\psi_{j,l}(y) \\ \Psi^2_{j,k,l}(x,y,z) &= \psi_{j,k}(x)\phi_{j,l}(y), \quad & \Psi^3_{j,k,l}(x,y,z) &= \psi_{j,k}(x)\psi_{j,l}(y). \end{aligned}$$

Note that now the coefficients $c^L_{k,l}, d^{j,\tau}_{k,l}$ depend on the view direction $\boldsymbol{\theta}$.

This transform can be viewed as a close relative of the *wavelet X-ray transform* which combines integration over a line with a simultaneous 1D wavelet transform along this line [16]. The difference is that in our case we perform a 2D wavelet transform in the plane perpendicular to the line.

Now we can state the main result.

Theorem. *The coefficients in the wavelet representation (3.1) for the X-ray transform of* $f \in L^2(\mathbb{R}^3)$ *are given by*

$$c^L_{k,l}(\boldsymbol{\theta}) = \mathcal{F}_2^{-1}\left(\mathcal{F}_2 \mathcal{P}_{\boldsymbol{\theta}} f \cdot \mathcal{F}_2 \widetilde{\Phi}^0_L\right)(2^j k, 2^j l) \tag{3.2}$$

$$d^{j,\tau}_{k,l}(\boldsymbol{\theta}) = \mathcal{F}_2^{-1}\left(\mathcal{F}_2 \mathcal{P}_{\boldsymbol{\theta}} f \cdot \mathcal{F}_2 \widetilde{\Psi}^\tau_j\right)(2^j k, 2^j l), \tag{3.3}$$

where

$$\widetilde{\Phi}^0_L(u,v) = \overline{\Phi^0_{L,0,0}(-u,-v)}, \quad \widetilde{\Psi}^\tau_j(u,v) = \overline{\Psi^\tau_{j,0,0}(-u,-v)}.$$

Proof. We only prove (3.3), the result (3.2) follows analogously. Using the Plancherel formula, we observe

$$\begin{aligned} d^{j,\tau}_{k,l}(\boldsymbol{\theta}) &= \langle \mathcal{P}_{\boldsymbol{\theta}} f, \Psi^\tau_{j,k,l} \rangle = \langle \mathcal{F}_2 \mathcal{P}_{\boldsymbol{\theta}} f, \mathcal{F}_2 \Psi^\tau_{j,k,l} \rangle \\ &= \iint \mathrm{d}\omega_u \, \mathrm{d}\omega_v \, \mathcal{F}_2 \mathcal{P}_{\boldsymbol{\theta}} f(\omega_u, \omega_v) \, \overline{\mathcal{F}_2 \Psi^\tau_{j,k,l}(\omega_u, \omega_v)}. \end{aligned} \tag{3.4}$$

Now,

$$\begin{aligned}\overline{\mathcal{F}_2\Psi^\tau_{j,k,l}(\omega_u,\omega_v)} &= \iint \mathrm{d}u\ \mathrm{d}v\, e^{2\pi i(\omega_u u+\omega_v v)}\overline{\Psi^\tau_{j,k,l}(u,v)}\\ &= \iint \mathrm{d}u\ \mathrm{d}v\, e^{2\pi i(\omega_u u+\omega_v v)}\widetilde{\Psi}^\tau_j(2^jk-u,2^jl-v)\\ &= e^{2\pi i(2^jk\omega_u+2^jl\omega_v)}\iint \mathrm{d}u'\ \mathrm{d}v'\, e^{-2\pi i(\omega_u u'+\omega_v v')}\widetilde{\Psi}^\tau_j(u',v')\\ &= e^{2\pi i(2^jk\omega_u+2^jl\omega_v)}\,\mathcal{F}_2\widetilde{\Psi}^\tau_j(\omega_u,\omega_v).\end{aligned}$$

Using this in (3.4), we find

$$\begin{aligned}d^{j,\tau}_{k,l}(\boldsymbol{\theta}) &= \iint \mathrm{d}\omega_u\ \mathrm{d}\omega_v\, \mathcal{F}_2\mathcal{P}_{\boldsymbol{\theta}}f(\omega_u,\omega_v)\, e^{2\pi i(2^jk\omega_u+2^jl\omega_v)}\,\mathcal{F}_2\widetilde{\Psi}^\tau_j(\omega_u,\omega_v)\\ &= \mathcal{F}_2^{-1}\left(\mathcal{F}_2\mathcal{P}_{\boldsymbol{\theta}}f\ \cdot\ \mathcal{F}_2\widetilde{\Psi}^\tau_j\right)(2^jk,2^jl).\end{aligned}$$

□

Note that by the Fourier slice theorem (2.3), $\mathcal{F}_2\mathcal{P}_{\boldsymbol{\theta}}f(\omega_u,\omega_v)=\mathcal{F}_3f(\omega_u\mathbf{u}+\omega_v\mathbf{v})$. Therefore the wavelet coefficients at scale j in (3.1) can be computed by multiplying a slice of the 3D Fourier transform of f by the 2D Fourier transform of the scaling or wavelet function at scale j, followed by an inverse 2D Fourier transform evaluated at the points of the form $(2^jk,2^jl)$ in the view plane.

3.2. **Algorithm for discrete input data.** The implementation of the proposed extension of Fourier domain volume rendering is facilitated by the fact that wavelet decomposition and reconstruction can be performed in the Fourier domain, without the need of repeatedly switching between the spatial and Fourier domain [12,14]. Omitting the details, the algorithm for discrete 3D input data then takes the following form:

- *Pre-processing.* Compute the 3D Fourier Transform of the volume data.
- *Actual volume rendering.* For each direction $\boldsymbol{\theta}$, do:
  1. Interpolate the Fourier Transform on a regular grid of points in the slice plane orthogonal to $\boldsymbol{\theta}$, as in ordinary Fourier domain rendering.
  2. Perform a 2D wavelet decomposition (depth L) in Fourier space.
  3. Do a partial reconstruction in Fourier space and perform a 2D inverse Fourier transform to obtain an approximation in the spatial domain.

A comparison of Fourier domain X-ray wavelet rendering and wavelet splatting is currently carried out, w.r.t to both efficiency and accuracy. A disadvantage of the Fourier domain X-ray wavelet rendering method is that first a slice in Fourier space at full resolution has to be interpolated. Therefore, we plan to investigate the combination of both methods to take advantage of the strengths of each of them.

References

[1] Burrus, C. S., Gopinath, R. A., and Guo, H. *Wavelets and Wavelet Transforms.* Prentice-Hall, Englewood Cliffs, NJ, 1998.

[2] Chui, C. K. *An Introduction to Wavelets.* Academic Press, New York, 1992.

[3] Cohen, A., Daubechies, I., and Feauveau, J. C. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math. 45* (1992), 485–560.

[4] Drebin, R., Carpenter, L., and Hanrahan, P. Volume rendering. *Computer Graphics (SIGGRAPH '88 proceedings) 22*, 4 (1988), 65–74.

[5] Levoy, M. Volume rendering: Display of surfaces from volume data. *IEEE Computer Graphics and Applications 8*, 3 (1988), 29–37.

[6] Lippert, L., and Gross, M. Fast wavelet based volume rendering by accumulation of transparent texture maps. *Computer Graphics Forum 14*, 3 (1995), 431–443.

[7] Lorensen, W. E., and Cline, H. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics 21*, 4 (July 1987), 163–169.

[8] Lounsberry, M., DeRose, T. D., and Warren, J. Multiresolution analysis of surfaces of arbitrary topological type. *ACM Trans. Graphics 16*, 1 (1997), 34–73.
[9] Malzbender, T. Fourier volume rendering. *ACM Transactions on Graphics 12*, 3 (1993), 233–250.
[10] Natterer, F. *The Mathematics of Computerized Tomography*. B.G. Teubner & J. Wiley, 1986.
[11] Puppo, E., and Scopigno, R. Simplification, LOD and multiresolution principles and applications. *Proceedings Eurographics'97, Computer Graphics Forum 16*, 3 (1997).
[12] Rioul, O., and Duhamel, P. Fast algorithms for discrete and continuous wavelet transforms. *IEEE Trans. Inform. Theory 38* (1992), 569–586.
[13] Strang, G., and Nguyen, T. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
[14] Vetterli, M., and Kovačevič, J. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
[15] Westover, L. Footprint evaluation for volume rendering. *Computer Graphics 24*, 4 (1990), 367–376.
[16] Zuidwijk, R. A. The wavelet X-ray transform. Report PNA-R9703, Centre for Mathematics and Computer Science, Amsterdam, Mar. 1997.