Wavelet Projections for Volume Rendering

Stefan Horbelt, Michael Unser, Martin Vetterli

Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland
email:{stefan.horbelt, michael.unser, martin.vetterli}@epfl.ch
URL:http://bigwww.epfl.ch

Abstract

We extended Gross's [3] method of volume wavelet rendering by computing splats via an orthogonal projection operator. The method decomposes the volume data into a wavelet pyramid representation in the spline domain. The splats of the basis functions are approximated on a multiresolution grid. Using least-squares approximation ensures the smallest possible error for a given sampling step size. The approximation error on the grid is derived as a function of the sampling step h. The choice of the appropriate wavelet space and spatial resolution at each step produces the smallest possible filters. Our approach reduces the number of computations and allows full control of the image quality.

keywords: *volume rendering, wavelet splat, approximation theory, multiresolution grid*

1. Introduction

Volume rendering [5] is a useful technique in biomedical visualisation, material testing, physics or atmospherical forecast. It is computationally expensive and consumes a lot of memory and bandwidth. Fast implementations rely on hardware rendering and coarse approximations. A user would like to browse interactively around and through a volume to get a better sensation of its three-dimensionality. One might accept the compromise of lower image resolution or quality for the sake of fast feedback. Then, once a viewpoint of the volume has been chosen, the resolution increases again[3, 4, 6].

We are interested in interactive volume rendering with multiresolution, similarly to Gross [3, 6] but with two important differences. First, instead of interpolation-based wavelet splatting, we use wavelet splats that are optimal in the least-square sense, hence better quality. Second, the computation of the projection is done on a grid whose sampling step size adapts to the size of the basis function. This gives an additional speed-up at coarse resolutions.

2. Rendering in wavelet space

In the following, we consider a volume of size M^3 which is projected on a rendering screen of size N^2 for the purpose of visualization. This operation is called the X-ray transform or volume rendering. The volume data is represented in a L_2-optimal multiresolution pyramid [2, 7].

2.1. The volume rendering integral

In its simplest form, the volume rendering integral is equal to the parallel projection p_θ in the direction $\vec{\theta}$ defined by

$$g_\theta(u,v) = p_\theta f(\vec{y}) = \int f(t\cdot\vec{\theta} + u\cdot\vec{e}_u + v\cdot\vec{e}_v)dt, \qquad (1)$$

where $f(\vec{x})$, $\vec{x} \in R^3$ is the volume, and where $g_\theta(\vec{y})$, $\vec{y} = (u,v) \in R^2$ is the projection of the volume on a plane perpendicular to $\vec{\theta}$ and spanned by the vectors $\vec{e}_u$ and $\vec{e}_v$.

Equation (1) needs three coordinate systems (see Figure 1): Cartesian coordinates $\vec{x} = (x,y,z)$, spherical coordinates (α,β,r) and projection plane coordinates $\vec{u} = (t,u,v)$. The origins of the three coordinate systems are identical. The projection direction $\vec{\theta}$ is aligned with the normal $\vec{e}_t$ of the projection plane. This plane is spanned by $\vec{e}_u$ and $\vec{e}_v$, which can be expressed in Cartesian coordinates as a function of the two Euler angles α and β (see Figure 1).

The coordinate change between Cartesian coordinates and projection plan coordinates is a simple rotation: $\vec{x} = R\cdot\vec{u}$ where the rotation matrix is $R = [\vec{e}_t \;\; \vec{e}_u \;\; \vec{e}_v]$, or, written explicitly,

$$R = \begin{bmatrix} -\cos\alpha\cdot\cos\beta & \cos\alpha\cdot\sin\beta & -\sin\alpha \\ -\sin\alpha\cdot\cos\beta & \sin\alpha\cdot\sin\beta & \cos\beta \\ -\sin\beta & -\cos\alpha & 0 \end{bmatrix}. \qquad (2)$$

The volume data is represented in the wavelet domain by

Published by the Eurographics Association, ISSN 1017-4656.

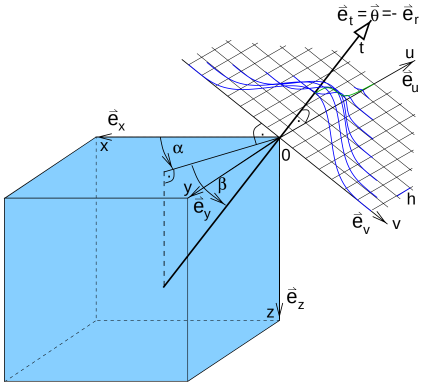

Figure 1: *Coordinate systems used for parallel projection and Fourier slice theorem: Cartesian coordinates (x, y, z), spherical coordinates (α, β, r) and intersection plane coordinates (t, u, v).*

linear combination of a set of basis functions $\psi_k(\vec{x})$:

$$f(\vec{x}) = \sum_k c(k)\psi_k(\vec{x}). \tag{3}$$

Using this decomposition, the volume rendering integral from (1) becomes

$$\begin{aligned} p_\theta f(\vec{y}) &= \sum_k c(k)\int \psi_k(t\cdot\vec{\theta}+u\cdot\vec{e}_u+v\cdot\vec{e}_v)dt = \\ &= \sum_k c(k)\cdot \underbrace{p_\theta\psi_k(\vec{y})}_{\text{wavelet splat } \chi_{\theta,k}(\vec{y})} \quad , \end{aligned} \tag{4}$$

which is a linear combination of wavelet splats $\chi_{\theta,k}(\vec{y}) = p_\theta\psi_k(\vec{y})$, where $\vec{y} = (u,v) \in R^2$. Note that k may code for a combination of shifts and scaling (wavelet decomposition).

2.2. The B-spline splat

We use the B-spline as underlying separable basis function. It has numerous advantages, like compact support and explicit formulas, for details see [1]. The Fourier transform of a B-spline of degree n is given by

$$\hat{\beta}^n(\omega) = (sin(\omega/2)/(\omega/2))^{n+1}. \tag{5}$$

In order to get the wavelet splat, the three-dimensional B-spline function $\beta^n(x,y,z) = \beta^n(x)\beta^n(y)\beta^n(z)$ has to be projected along the direction $\vec{\theta}$. The Fourier slice theorem states that the intersection of a volume by the plane spanned by $\vec{e}_u$ and $\vec{e}_v$ in the Fourier domain, is equivalent to the 2D Fourier transform

$$\hat{g}(\omega_u,\omega_v) = \hat{\beta}^n(\omega_x)\hat{\beta}^n(\omega_y)\hat{\beta}^n(\omega_z) \tag{6}$$

of the wavelet splat $\chi_\theta(u,v)$, with the substitution

$$\left[\ \omega_x \quad \omega_y \quad \omega_z\ \right]^\top = R\left[\ \omega_u \quad \omega_v \quad \omega_t\ \right]^\top_{|\omega_t=0}. \tag{7}$$

We use the Fourier transform of the B-spline splats to evaluate the approximation error.

2.3. Approximation on an adaptive grid

In Gross' approach[6], the wavelet splats are always calculated at the full resolution $h = 1$. Here, we investigate an alternative approach where the resolution of the grid h may be adapted to the size of the basis functions and where, in order to get high quality, we use least-squares approximation rather than the usual interpolation method.

The wavelet splat $\chi_{\theta,j}(\vec{y})$ is approximated onto a reconstruction grid with the sampling step h_j using basis functions $\varphi(\frac{\vec{y}}{h_j} - l)$, where $h_j = 2^j h$ adapts to the scale j. We also use the fact that $\varphi(\vec{y})$ satisfies a two-scale relation to get a fast full-screen interpolation using digital filters.

With the approximation operator P_{h_j}, the approximation of the wavelet splat is

$$P_{h_j}\chi_{\theta,j}(\vec{y}-\vec{z}) = \sum_{\vec{l}\in Z^2} c_{h_j}(\vec{l})\varphi(\frac{\vec{y}}{h_j} - \vec{l}) \tag{8}$$

with

$$c_{h_j}(\vec{l}) = \langle \chi_{\theta,j}(\vec{y}-\vec{z}), \tilde{\varphi}(\frac{\vec{y}}{h_j} - \vec{l})\rangle \frac{1}{h_j}, \tag{9}$$

where $\tilde{\varphi}(\vec{y})$ is the dual function of $\varphi(\vec{y})$ [1].

Here, $\chi_{\theta,j}(\vec{y}-\vec{z})$ denotes the continuously-defined B-spline splat at direction $\vec{\theta}$, scale j and shift $\vec{z}$. The coefficients $c_{h_j}(\vec{l})$ yield its discrete representation on the visualization grid with the sampling step h_j.

As the sum (4) is performed over the whole, huge volume, it is wise to select a function of *short support* for the dual basis function $\tilde{\varphi}(\vec{x})$ in (9). This gives short and efficient filters. The cubic B-spline $\beta^3(\vec{x})$ is a good candidate. Ordinary sampling, as used in [6], is equivalent to the choice $\tilde{\varphi}(\vec{x}) = \delta(\vec{x})$, but the quality is not as good as with the least-squares approximation.

The error one does when approximating a function f on a grid with the sampling step h is given in [2] by

$$\varepsilon(h) = \|f - P_h f\|^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} E(h\omega)|\hat{f}(\omega)|^2 d\omega \tag{10}$$

where $\hat{f}(\omega)$ is the Fourier Transform of the function f. The error kernel $E(\omega)$ is defined by

$$E(\omega) = 1 - \frac{|\hat{\varphi}(\omega)|^2}{\sum_n |\hat{\varphi}(\omega + 2n\pi)|^2}. \tag{11}$$

We evaluate the relative approximation error $\varepsilon_r(h) = \|f - P_h f\|/\|f\|$ for wavelet splats of cubic B-splines, $\hat{f}(\omega) = \hat{\beta}^3(\omega)$, using grid basis functions that are B-splines of degree $n = 0, 1, 3$. Cubic B-splines perform significantly better than linear or piecewise constants (see Table 1 and Figure 2).

h	Haar $n=0$	Linear $n=1$	Cubic $n=3$	Quintic $n=5$
1	32.6%	15.6%	11.0%	10.1%
0.75	25.2%	7.8%	3.4%	2.6%
0.5	17.2%	2.8%	0.34%	0.1%
0.25	8.7%	0.62%	0.010%	0.001%

Table 1: *Relative approximation error of cubic B-spline splat projected on grid with B-splines of degrees n and sampling steps h. We can choose an appropriate h to keep the approximation error below some threshold.*

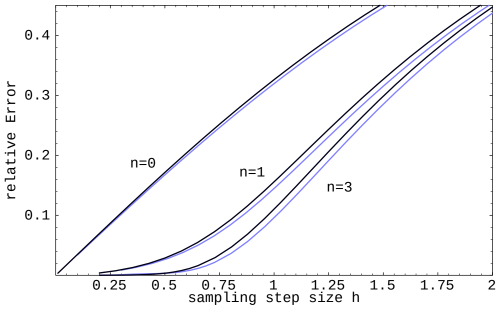

Figure 2: *Relative approximation error of cubic B-spline splat projected on grid with B-splines of various degrees n and sampling steps h. For each degree, the plot shows the error for the shortest cut in light blue (gray) and the slightly bigger error for the longest cut in black.*

2.4. Fast splatting with table lookup

The wavelet splat $\chi_{j,\theta}(\vec{y})$ in (8) depends on the projection direction $\vec{\theta}$ and the scale j. The approximation of this wavelet splat on the grid depends on the grid sampling step size h and the relative position z at which it hits the grid.

The coefficients $c_{h_j}(\vec{l})$ of the projection of the wavelet splat on the grid are given by the scalar product (9) between the wavelet splat $\chi_{j,\theta}(\vec{y})$ and the dual function $\tilde{\varphi}(\vec{y})$. For a fixed view, we can express (9) by a kernel function $\xi_{h,\theta}(\vec{s})$ as

$$c_{h_j}(\vec{l}) = \xi_{h,\theta}(\vec{s}) \;=\; \langle \frac{1}{h}\tilde{\varphi}(\frac{\vec{y}}{h}), \chi_{\theta}(\vec{y}+\vec{s})\rangle \qquad (12)$$

with a single shift parameter $\vec{s} = h\vec{l} - 2^{-j}\vec{z}$. The kernel $\xi_{h,\theta}(\vec{s})$ is calculated once for the current viewing direction at fine resolution in $\vec{s}$ and stored in a table. Finally a single table look-up is necessary to get one grid coefficients.

3. Computational complexity

In this section we compare our L_2-optimal wavelet splatting method with the one proposed by Gross and Lippert [3, 6], see Figure 3.

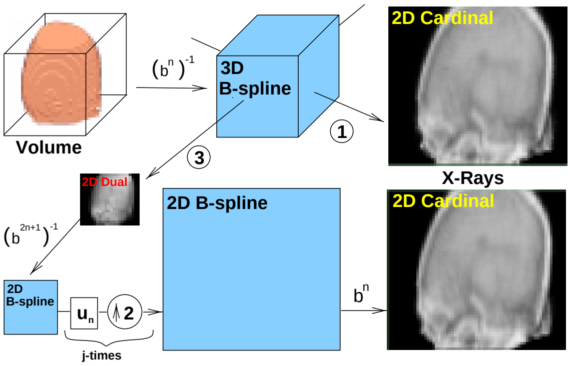

Figure 3: *Volume rendering using wavelet splats. Method 1 splats at full resolution. The proposed method 2 approximates on a low resolution grid in dual space, then interpolates in spline space and visualizes at full resolution in cardinal space.*

Method 1: Full resolution splat

Let us consider the 3D multiresolution wavelet decomposition of a volume. It is straightforward to splat each 3D coefficient to the rendering screen at full resolution as described in [6]. The splat is calculated once for the current viewing direction at full resolution. Then, for each coefficient, the splat is multiplied with the value of the coefficient and accumulated in the rendering buffer. The number of operations Op_1 depends on the size of the splat s^2 and on the number of wavelet coefficients M^3. We refer to this as Method 1.

It needs Op_1 operations

$$\mathrm{Op}_1 = k_1 \cdot \underbrace{\left(\frac{M}{2^j}\right)^3}_{\text{voxels at scale } j} \cdot \underbrace{\left(2^j\frac{s+1}{h}\right)^2}_{\text{size of spalt}} \sim 2^{-j}\cdot M^3, \quad (13)$$

where the support of the wavelet splat is $s(\Psi^n) = 2n-1$.

Method 2: Least-squares optimal splat

The volume is expanded in multiple resolutions in the spline space. Here, the wavelet splats are approximated in dual space on a grid with a sampling step *adapted* to the size of the basis function.

Splatting needs $\mathrm{Op}_{2.1}$ operations

$$\mathrm{Op}_{2.1} = k_3 \cdot \left(\frac{M}{2^j}\right)^3 \cdot \left(\frac{s+1}{h}\right)^2 \sim 2^{-3j}\cdot M^3, \qquad (14)$$

where the splat support for the B-spline scaling function $s(\varphi_n) = (n_v+1)+(n_g+1)$, or, if using the B-spline wavelet $s(\psi_n) = (2n_v-1)+(n_g+1)$. Note that the factor 2^j at the support s has disappeared. This is because the sampling step h of the grid adapts to the scale j, with $h_j = 2^j \cdot h$.

Once the volume has been projected on the grid, the projected data needs to be interpolated to the full-screen resolution N^2 and visualized in cardinal space. This is achieved in three steps: The first step changes from dual space to spline space, which is fast as the low-resolution data is smaller in size. The spline space has the shortest upsampling interpolation filters. In the second step, these filters are used to reach the final full resolution. The last step changes from spline to the cardinal space for the purpose of visualization.

Finally the total number of operations of Method 2 is:

$$\begin{aligned} \mathrm{Op}_2 \;&=\; k_3\left(2^{-j}M\right)^3\cdot(3n+1)^2h^{-2}+ \\ &+\; k_7N^2\left(2^{-2j}h^{-2}+2^{-1}\right)\cdot(2n+1)^2\sim \\ &\sim\; 2^{-3j}\cdot M^3+k_7N^2 \end{aligned} \tag{15}$$

Comparison of methods

The complexity of both methods are compared in Figure 4 for h=1 (no oversampling) and M=N, which means that the rendering screen has the same resolution per dimension as the volume data. The complexity of Method 2, Op_2, is significantly lower than the complexity of Method 1, Op_1, for scale $j > 1$. Method 2 is up to two order of magnitude faster at scale around $j \in \{4,5\}$ and for large volume $M \in \{512,1024,2048\}$. The complexity Op_2 decays with a factor 8^{-j} per scale j at small scales, until it reaches a constant that depends only on the rendering screen size N^2. The factor 8^{-j} per scale j is already the theoretical limit, as the complexity cannot decrease faster than the number of voxels decrease per scale j. The complexity Op_1 falls slower with a decay of only 2^{-j}. A clever algorithm can adapt N or j to the user behavior, reduce the resolution or zoom into areas of interest and take advantage of the high quality of the interpolation.

4. Conclusion

We have purposed a volume rendering method based on wavelet splat and multiresolution approximation grids. It allows full control over the approximation error and achieves the smallest error for a given sampling step size. To speed up computation, one may sacrifice some quality (see Figure 2) and adapt the rendering step to the size of the basis function. With this strategy, the proposed method can be up to two order of magnitude faster than standard ones[3]. The speed-up breaks down only at full resolution, where the complexity of the proposed algorithm is 50% higher than the direct wavelet splatting method, but we end up with a higher quality result, especially when the data has a significant high frequency content. To conclude, wavelet splatting with approximation grids may speeds up interactive or progressive volume rendering; it yields higher quality for a given resolution.

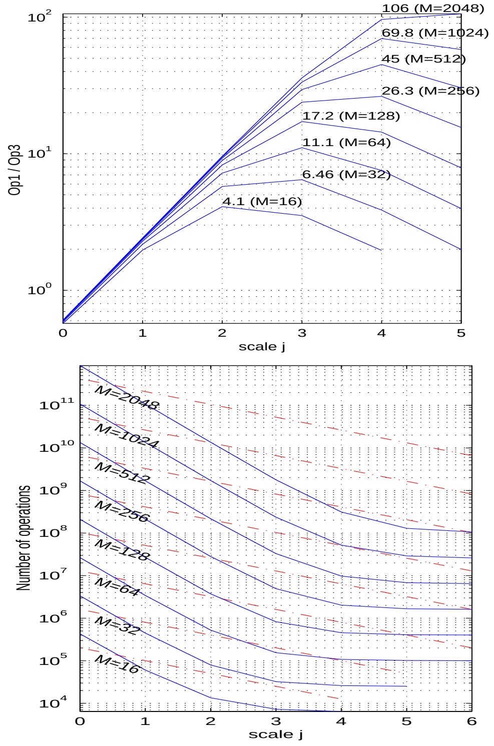

Figure 4: *Comparison of number of operations (1 Operation = 1 addition + 1 multiplication) of wavelet splatting methods 1 and 2. Top: Ratio Op_1/Op_2. Bottom: Number of operations. Op_1 is drawn as dashed line and Op_2 as solid line. We assume $N = M$ and $h = 1$. Op_2 decays with up to 2^{-3j} per scale j, whereas Op_1 decays with only 2^{-j}.*

References

1. M.Unser, A.Aldroubi and M.Eden, "B-spline signal processing: Part I—theory," IEEE Trans. Signal Proc., Vol. 41, No. 2, pp. 821–833, 1993.
2. T.Blu, M.Unser, "Quantitative Fourier Analysis of Approximation Techniques: Part I—Interpolators and Projectors, Part II—Wavelets", to appear in IEEE Trans. Signal Process., 1999.
3. M.H.Gross, L.Lippert, R.Dittrich, "Two Methods for Wavelet-Based Volume Rendering", Comp. & Graphics, Vol. 21, No.2, pp. 237–252, 1997.
4. D.Laur, R.Hanrahan, "Hierarchical Splatting: A Progressive Refinement Algorithm for Vol. Rendering", Siggraph'91, Vol.25, No.4, pp.285–289, 1991.
5. B.Lichtenbelt, R.Crane, S.Naqvi, "Introduction to Volume Rendering", Hewlett-Packard, Prentice-Hall, 1998.
6. L.Lippert, M.Gross, "Fast Wavelet-Based Volume Rendering by Accumulation of Transparent Texture Maps", Eurograph'95, Vol.14, No.3, pp.431–44, 1995.
7. M.Vetterli, J.Kovačević, "Wavelets and Subband Coding", Prentice Hall, 1995.