Institut für Computergraphik und Algorithmen
Technische Universität Wien
Karlsplatz 13/186/2
A-1040 Wien
AUSTRIA
Tel: +43 (1) 58801-18601
Fax: +43 (1) 58801-18698

Institute of Computer Graphics and Algorithms
Vienna University of Technology
email:
technical-report@cg.tuwien.ac.at
other services:
http://www.cg.tuwien.ac.at/
ftp://ftp.cg.tuwien.ac.at/

TECHNICAL REPORT

D^{2}VR: High-Quality Volume Rendering of Projection-based Volumetric Data

Balázs Csébfalvi
Budapest University of Technology
Hungary

Peter Rautek
Vienna University of Technology
Austria

Sören Grimm†
Vienna University of Technology
Austria

Stefan Bruckner†
Vienna University of Technology
Austria

Eduard Gröller†
Vienna University of Technology
Austria

TR-186-2-05-3
April 2005

Keywords: Volume Rendering, Reconstruction, Filtered Back-Projection

D^{2}VR: High-Quality Volume Rendering of Projection-based Volumetric Data

Balázs Csébfalvi*
Budapest University of Technology
Hungary

Peter Rautek†
Vienna University of Technology
Austria

Sören Grimm†
Vienna University of Technology
Austria

Stefan Bruckner†
Vienna University of Technology
Austria

Eduard Gröller†
Vienna University of Technology
Austria

Figure 1: CT scan of stag beetle (832x832x494): (a) DVR of original grid. (b) D^{2}VR of projection-based volumetric data (64 projections, each projection with a resolution of 64^2). (c) DVR of a 128^3 grid (d) DVR of a 64^3 grid. Grids are reconstructed from 64 filtered projections, each projection with a resolution of 64^2.

Abstract

Volume-rendering techniques are conventionally classified into two categories represented by direct and indirect methods. Indirect methods require to transform the initial volumetric model into an intermediate geometrical model in order to efficiently visualize it. In contrast, direct volume-rendering (DVR) methods can directly process the volumetric data. Modern 3D scanning technologies, like CT or MRI, usually provide data as a set of samples on rectilinear grid points, which are computed from the measured projections by discrete tomographic reconstruction. Therefore the set of these reconstructed samples can already be considered as an intermediate volume representation. In this paper we introduce a new paradigm for *direct* direct volume rendering (D^{2}VR), which does not require a rectilinear grid at all, since it is based on an immediate processing of the measured projections. Arbitrary samples for ray casting are reconstructed from the projections by using the Filtered Back-Projection algorithm. Our method removes an unnecessary and lossy resampling step from the classical volume rendering pipeline. Thus, it provides much higher accuracy than traditional grid-based resampling techniques do. Furthermore we also present a novel high-quality gradient estimation scheme, which is also based on the Filtered Back-Projection algorithm. Finally we introduce a hierarchical space partitioning approach for projection-based volumetric data, which is used to accelerate D^{2}VR.

CR Categories: I.4.5 [Image Processing]: Reconstruction—Transform Methods; I.4.10 [Image Processing]: Volumetric Image Representation; I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism

Keywords: Volume Rendering, Reconstruction, Filtered Back-Projection

*e-mail: cseb@iit.bme.hu
†e-mail: {rautek|grimm|bruckner|groeller}@cg.tuwien.ac.at

1 Introduction

Modern 3D scanning technologies, like Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), usually provide data values on rectilinear grid points. These data values are computed from measured projections by discrete tomographic reconstruction [19, 6]. The set of the reconstructed data values (or samples) can be interpreted as a discrete representation of the underlying continuous phenomenon. In order to authentically visualize the original continuous signal, it has to be accurately reconstructed from the discrete samples (note that such a signal reconstruction is differentiated from discrete tomographic reconstruction).

From a signal-processing point of view, the original signal can be perfectly reconstructed from discrete samples if it is band-limited and the sampling frequency is above the Nyquist limit [16]. Theoretically the perfect continuous reconstruction is obtained by convolving the discrete volume representation with the *sinc* function. The *sinc* function is considered to be the best reconstruction kernel, since it represents an ideal low-pass filter. In practice, however, it is difficult to convolve a discrete signal with the *sinc* kernel, because of its infinite support. Therefore practical reconstruction filters either approximate it or truncate it with an appropriate windowing function [11, 20]. Moreover, real-world signals can hardly be considered band-limited. As a consequence, practical resampling results in a loss of information.

Figure 2 shows the signal-processing approach of the traditional volume rendering pipeline (follow the red line). The first step is the discrete tomographic reconstruction of a rectilinear volume representation from the measured projections. Although there exist different algorithms for tomographic reconstruction, one of the most popular techniques is the Filtered Back-Projection algorithm. It first performs high-pass filtering on the measured projections and afterwards the samples at rectilinear grid points are computed by back-projecting the filtered signals. As the projections are acquired by measuring accumulated attenuation by a limited number of sensors, they are actually available as discrete representations of continuous projection functions. Therefore high-pass filtering is performed in discrete frequency domain, so the result is also a discrete function. In the back-projection phase, however, the rectilinear grid points are not necessarily projected exactly onto the discrete samples of the filtered projections. Therefore for back-projection resampling is necessary, which results in the *first* loss of information in the pipeline.

The obtained rectilinear volume can be visualized by different rendering techniques. Using indirect methods, like the classical Marching Cubes algorithm [9], an intermediate geometrical model of an isosurface is constructed from the volumetric model. This geometrical model is then interactively rendered by, for example, conventional graphics hardware. In contrast, Direct Volume Rendering (DVR) approaches, like ray casting [7] or splatting [24, 25] directly render the volumetric model without any intermediate representation. In both cases an interpolation technique is applied to define data values between the rectilinear grid points. In other words, a resampling of the discrete volume representation is required. This resampling results in the *second* loss of information in the traditional pipeline.

In order to minimize the loss of information we propose to modify the traditional volume-rendering pipeline by simply removing an unnecessary resampling step (follow the green line in Figure 2). To render the underlying continuous phenomenon, data samples at arbitrary sample points need to be defined, and for shading computation the corresponding gradients need to be determined. As it will be shown, both tasks can be solved using directly the filtered projections. This eventually leads to an alternative projection-based volume representation. Thus, there is no need to compute samples at regular grid points by discrete tomographic reconstruction, and as a consequence one resampling step (see Figure 2) is unnecessary. Traditional direct volume-rendering methods rely on such an intermediate grid representation, so in this sense they are in fact indirect. In contrast, we present DVR *directly* from the measured raw data. To distinguish from the common DVR the novel approach is referred to as D^2VR (pronunciation: [di-skwerd vi ar]).

In Section 2 we review previous work related to discrete tomographic reconstruction and volume resampling. In Section 3 our novel volume-rendering approach is introduced, where it is explained how to reconstruct data values and gradients directly from the filtered projections by using the Filtered Back-Projection algorithm. Furthermore, in Section 3.3, we also present a hierarchical space partitioning approach for projection-based volumetric data. Section 4 describes the implementation. Section 5 reports the results. Finally in Section 6 the contribution of this paper is summarized and ideas for future work are given.

2 Related Work

In most of the practical volume-rendering applications, especially in 3D medical imaging, the input data is usually generated from measured projections by using tomographic reconstruction [19, 6, 15]. The set of projec-

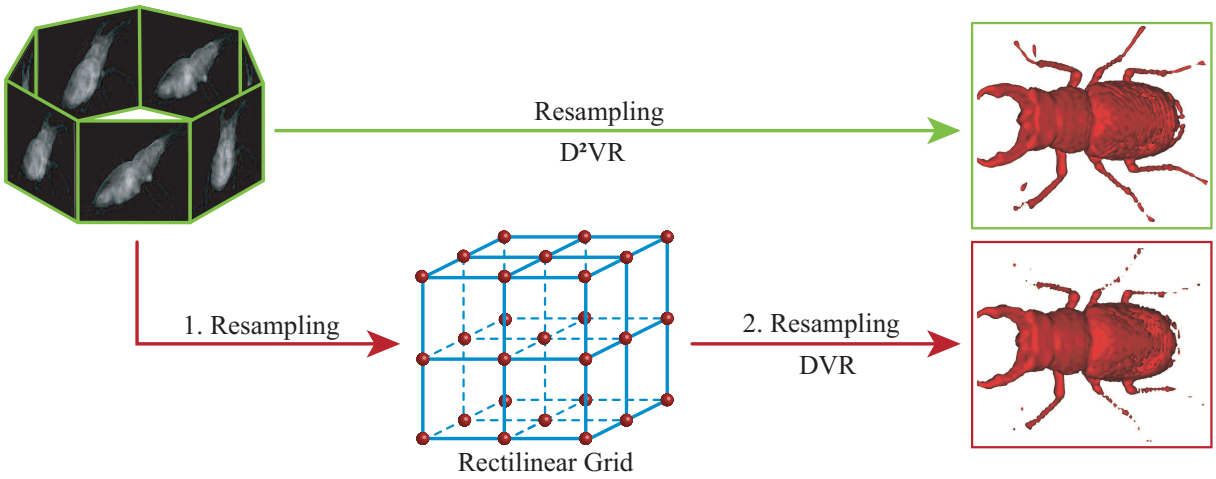

Figure 2: Data processing work flow of projection- and grid-based volume rendering. The red line corresponds to the traditional volume rendering pipeline. It requires two resampling steps in order to visualize the data. First an intermediate grid is resampled and then this grid is resampled again for rendering. The green line corresponds to the projection-based volume rendering pipeline; the lossy first resampling step is avoided.

tions is referred to as the Radon transform of the original signal. Therefore the tomographic reconstruction is, in fact, the inversion of the Radon transform. The inversion can be performed by using the classical Filtered Back-Projection [2] algorithm, which is based on the Fourier projection-slice theorem [6, 10]. Although there exist alternative tomographic reconstruction techniques like algebraic or statistical ones, Filtered Back-Projection is still the most popular method used in commercial CT scanners.

The output of tomographic reconstruction is a discrete (or sampled) representation of the underlying continuous phenomenon. The samples are conventionally generated on rectilinear grid points. The rectilinear grid has several advantages. For example, the sampled signal can be represented by 3D arrays, implicitly storing the locations of the samples. Furthermore, the neighborhood of a certain sample can be efficiently addressed, which is important for many volume-processing or volume-rendering algorithms.

Nevertheless, in order to render the underlying continuous 3D function, data values need to be defined also between the rectilinear grid points. The *sinc* kernel as ideal reconstruction filter is impractical because of its infinite extent. In practice it is approximated by filters of finite support [11, 20]. Generally, the wider the support of the reconstruction filter, the higher the quality of the reconstruction. On the other hand, the wider the support of the filter, the higher the computational cost of a spatial-domain convolution. Therefore several researchers analyzed different reconstruction filters, both in terms of accuracy and computational cost [12, 11, 13, 14]. As the practical filters only approximate the ideal low-pass filter they result in either aliasing or smoothing [11], which can be interpreted as a loss of information.

For frequent resampling tasks, like rotation, or upsampling, frequency-domain techniques can be alternatively applied [8, 1, 3, 4, 21, 22]. In frequency domain, it is exploited that a computationally expensive spatial-domain convolution is replaced by a simple multiplication. Although the frequency-domain resampling methods generally provide higher accuracy than spatial-domain methods do, they assume that the new samples to be computed are also located at regular grid points.

In order to avoid a lossy resampling step in the traditional volume-rendering pipeline, we directly use the tomographic inversion in order to reconstruct the underlying function at arbitrary sample positions. Therefore we do not generate an intermediate rectilinear volume representation, but we directly process the filtered projections as an alternative volume representation. Using this gridless or projection-based volume-rendering approach as a new paradigm, the same accuracy can be ensured at all the sample positions. In contrast, using the traditional grid-based approach, accurate samples are available only at the grid points, while the accuracy of intermediate samples depends on the quality of the applied imperfect reconstruction filter.

3 D^{2}VR

We present D^{2}VR based on a raycasting approach. In order to perform raycasting the underlying 3D volumetric function needs to be reconstructed at arbitrary resampling locations. In case the data is given on a rectilinear grid the reconstructed function value is computed from a close neighborhood of samples as shown in Figure 3a. In contrast to that, for raycasting directly performed on the filtered projections the reconstructed function value is computed from the filtered projections at the corresponding positions, see Figure 3b. Furthermore, gradients at these resample locations need to be determined in order to perform shading. The whole reconstruction mechanism is described in the following.

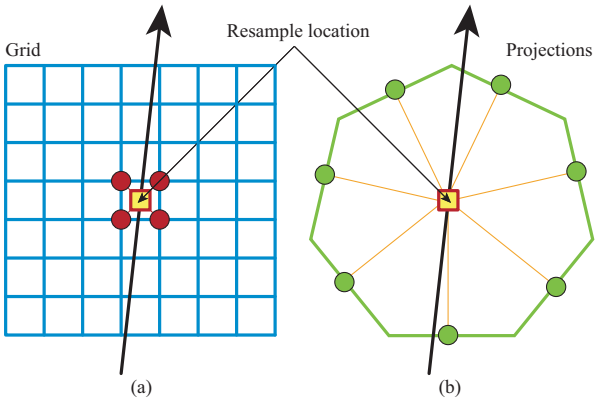

Figure 3: (a) Resampling along a ray on rectilinear volumetric data. (b) Resampling along a ray directly from the filtered projections

3.1 Data reconstruction

Data reconstruction from projection-based volumetric data requires Filtered Back-Projection. For simplicity we illustrate the Filtered Back-Projection in 2D based on a computed tomography scanning process using orthographic projection. Parallel projections are taken by measuring a set of parallel rays for a number of different angles. A projection is formed by combining a set of line integrals. The whole projection is a collection of parallel ray integrals as is given by $P_\theta(t)$ for a constant θ, see Figure 4. The line integrals are measured by moving an x-ray source and detector along parallel lines on opposite sides of the object.

The Filtered Back-Projection can be derived, using the Fourier projection-slice Theorem as follows:

The density function $f(x,y)$ can be expressed as:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}dudv$$

where $F(u,v)$ denotes the two-dimensional Fourier transform of the density function $f(x,y)$. By moving from a cartesian coordinate system in the frequency domain to a polar coordinate system, i.e., $u = w\cos\theta$, $v = w\sin\theta$, and $dudv = wdwd\theta$, we obtain:

$$f(x,y) = \int_0^{2\pi}\int_0^{\infty} F(w,\theta)e^{j2\pi w(x\cos\theta+y\sin\theta)}wdwd\theta$$

If we consider θ from 0 to π, the integral can be split as follows:

$$f(x,y) = \int_0^{\pi}\int_0^{\infty} F(w,\theta)e^{j2\pi w(x\cos\theta+y\sin\theta)}wdwd\theta + \int_0^{\pi}\int_0^{\infty} F(w,\theta+\pi)e^{j2\pi w(x\cos(\theta+\pi)+y\sin(\theta+\pi))}wdwd\theta$$

Since $F(w,\theta+\pi) = F(-w,\theta)$, the above expression can be written as:

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} F(w,\theta)\,|w|\,e^{j2\pi wt}dw\right]d\theta$$

where $t = x\cos\theta + y\sin\theta$. By substituting $S_\theta(w)$ for the two-dimensional Fourier transform $F(w,\theta)$ the above integral can be expressed as:

$$f(x,y) = \int_0^{\pi}\int_{-\infty}^{\infty} S_\theta(w)\,|w|\,e^{j2\pi wt}dwd\theta$$

According to the Fourier projection-slice theorem $S_\theta(w)$ is the Fourier transform of $P_\theta(t)$. Let us define:

$$Q_\theta(t) = \int_{-\infty}^{\infty} S_\theta(w)\,|w|\,e^{j2\pi wt}dw \qquad (1)$$

which is the inverse Fourier transform of $S_\theta(w)\cdot|w|$. As multiplication in frequency domain corresponds to a convolution in spatial domain, according to Equation 1, $Q_\theta(t)$ is obtained by high-pass filtering the measured projection $P_\theta(t)$. Other filters to reduce artifacts resulting from reconstruction can be applied, see [6].

In practice, the 2D density function $f(x,y)$ is discretely approximated by:

$$f(x,y) \approx \tilde{f}(x,y) = \frac{\pi}{K}\sum_{i=1}^{K} Q_{\theta_i}(x\cos\theta_i + y\sin\theta_i) \qquad (2)$$

where Q_{θ_i} are the filtered projections. Thus, according to Equation 2 the density function can be reconstructed

from a fixed number of projections. The Filtered Back-Projection algorithm is conventionally used for discrete tomographic reconstruction in order to obtain a rectilinear representation of the original density function. This intermediate representation is then usually resampled by many volume visualization algorithms. However, the formula in Equation 2 can also be considered as a resampling scheme to interpolate a density value at an arbitrary sample point. Therefore, it is unnecessary to generate an intermediate rectilinear representation by discrete tomographic reconstruction. Furthermore, each resampling step usually causes a loss of information. By avoiding the unnecessary intermediate grid representation, the quality of reconstruction can be improved. Previous reconstruction techniques assume that accurate samples are available at the grid points. In practice these samples are obtained by tomographic reconstruction. In order to maintain the same accuracy at any arbitrary sample location, we apply the Filtered Back-Projection to reconstruct the density value.

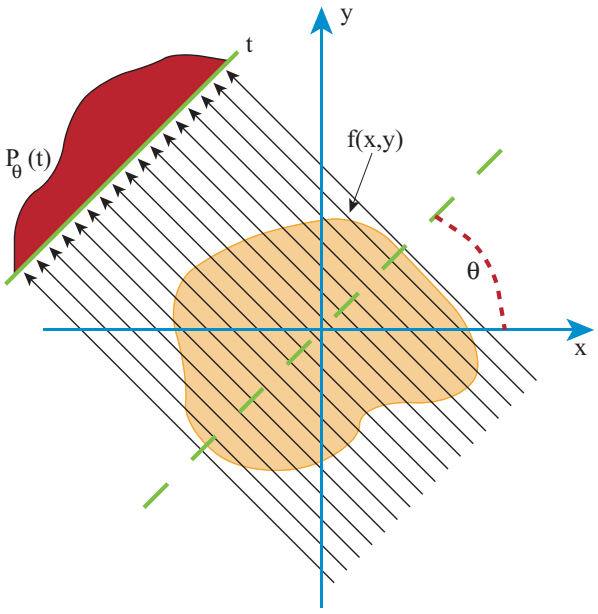

Figure 4: Parallel projection for a specific angle θ.

3.2 Derivative Estimation

In order to process or render volumetric data in many cases derivatives of the original density function are necessary. For example, for volume rendering the estimated gradients are used as surface normals to perform shading. In case of a grid based representation the straightforward way is to estimate the derivatives from a certain voxel neighborhood. To determine the gradient, common methods, such as intermediate difference gradient, central difference gradient, or higher order gradient estimation schemes are applied. In our case, computing the derivatives from a certain 3D neighborhood of samples requires to perform a large number of back-projections. Especially for higher order gradient estimation schemes, which need a large neighborhood of samples, the computational costs would be significantly high. However, the Filtered Back-Projection reconstruction scheme can also be exploited to compute derivatives.

For example the partial derivative $\tilde{f}_x$ according to variable x can be expressed by using Newton's difference quotient:

$$\tilde{f}_x = \frac{\partial \tilde{f}(x,y)}{\partial x} = \lim_{\Delta x \to 0} \frac{1}{\Delta x} \left(\frac{\pi}{K} \left(\sum_{i=1}^{K} Q_{\theta_i}(x\cos\theta_i + y\sin\theta_i) - \sum_{i=1}^{K} Q_{\theta_i}((x+\Delta x)\cos\theta_i + y\sin\theta_i) \right) \right)$$

Substituting $t_i := x\cos\theta_i + y\sin\theta_i$ we obtain:

$$\begin{aligned}
\tilde{f}_x &= \lim_{\Delta x \to 0} \frac{1}{\Delta x} \left(\frac{\pi}{K} \left(\sum_{i=1}^{K} Q_{\theta_i}(t_i) - \sum_{i=1}^{K} Q_{\theta_i}(\Delta x\cos\theta_i + t_i) \right) \right) \\
&= \lim_{\Delta x \to 0} \frac{1}{\Delta x} \left(\frac{\pi}{K} \sum_{i=1}^{K} Q_{\theta_i}(t_i) - Q_{\theta_i}(\Delta x\cos\theta_i + t_i) \right) \\
&= \frac{\pi}{K} \sum_{i=1}^{K} \lim_{\Delta x \to 0} \frac{1}{\Delta x} \left(Q_{\theta_i}(t_i) - Q_{\theta_i}(\Delta x\cos\theta_i + t_i) \right)
\end{aligned}$$

The term

$$\lim_{\Delta x \to 0} \frac{1}{\Delta x} \left(Q_{\theta_i}(t_i) - Q_{\theta_i}(\Delta x\cos\theta_i + t_i) \right) \qquad (3)$$

is the partial derivative of the projections Q_{θ_i}, but scaled with $\cos\theta_i$. We can therefore calculate the partial derivative $\tilde{f}_x$ directly as sum of scaled derivatives of the projection data. Analogously, taking the difference quotient with respect to y we obtain:

$$\tilde{f}_y = \frac{\partial \tilde{f}(x,y)}{\partial y} = \frac{\pi}{K} \sum_{i=1}^{K} \lim_{\Delta y \to 0} \frac{1}{\Delta y} \left(Q_{\theta_i}(t_i) - Q_{\theta_i}(\Delta y\sin\theta_i + t_i) \right)$$

It can be seen that applying Newton's difference quotient directly on the filtered projections is equivalent to applying Newton's difference quotient for the 2D density function $f(x,y)$. Moreover, any higher order derivative can be obtained by applying Newton's difference quotient multiple times.

Using Filtered Back-Projection for gradient estimation we expect higher accuracy than using the traditional gradient estimation schemes on the rectilinear grid. Consider central differences on the continuous reconstruction from a rectilinear representation. In order to calculate the gradient at an arbitrary sampling point six additional samples have to be interpolated. As interpolation usually causes loss of information, the introduced errors are accumulated in the estimated gradients. In contrast, using Filtered Back-Projection, the density values at the additional sample points are as accurate as the grid points. Therefore, no interpolation error is introduced.

3.3 Hierarchical Space Partitioning

For almost all volumetric processing approaches a hierarchical space partitioning data structure is essential for efficient processing. The performance gain which can be achieved with such a data structure mainly depends on its granularity. An octree is one of the most widely used data structure for organizing three-dimensional space. An octree is based on the principle of hierarchical space partitioning. Each node of the octree represents a cuboid cell. In a min-max octree for volumetric data each node contains the minimum and maximum of the enclosed data. The minimum and maximum value of an octree cell, in case of grid-based volumetric data, depends on the used data reconstruction method. The most widely used data reconstruction method is trilinear interpolation. The convex nature of trilinear interpolation ensures that all function values within a cuboid are bounded by the maximal and minimal values at the grid positions.

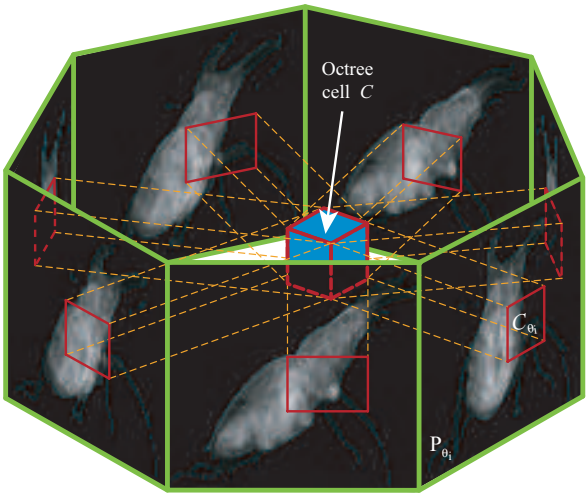

Figure 5: Projected octree cell C onto projection P_{θ_i}.

This convexity condition does not hold for reconstruction based on the Filtered Back-Projection. However, it is still possible to generate a min-max octree for projection-based volumetric data. Consider an octree cell C projected onto all filtered projections P_{θ_i}, see Figure 5. The resulting projections of C are referred to as C_{θ_i}. According to Equation 2 an upper bound of the maximum value contained in the octree cell C can be determined by:

$$\max C_{\tilde{f}} \leq \frac{\pi}{K} \sum_{i=1}^{K} \max C_{\theta_i \tilde{f}} \tag{4}$$

where

$$C_{\tilde{f}} = \left\{ v = \tilde{f}(\vec{x}) | \vec{x} \in C \subseteq \Re^3 \right\}$$

are all possible function values within the enclosing cuboid of cell C and

$$C_{\theta_i \tilde{f}} = \left\{ v = Q_{\theta_i}(\vec{t}) | \vec{t} \in C_{\theta_i} \subseteq \Re^2 \right\}$$

are all projection values within the enclosing rectangle of the projected cell C onto the projection P_{θ_i}. Analogously a lower bound of the minimum of the octree cell C can be determined by:

$$\min C_{\tilde{f}} \geq \frac{\pi}{K} \sum_{i=1}^{K} \min C_{\theta_i \tilde{f}} \tag{5}$$

The algorithm to generate an octree, using a bottom-up approach, for projection-based volumetric data is as follows:
Starting with a root cell, which encloses the entire volumetric function defined by the projections, we recursively, subdivide the octree cells. Once we reach a desired octree depth we compute the maximum respectively the minimum using Equation 4 and 5. These values are then propagated to the higher octree levels. Instead of propagating these minimums and maximums to the higher levels, it would have been also possible to compute the minimum and maximum directly for each of these higher levels using Equation 4 and 5. However, this would lead to a more conservative approximation. Although this space partitioning is a conservative approximation, it works very well in practice, as shown in Figure 6.

4 Implementation

In Section 3 we presented all the necessary components for D^{2}VR. First, in Section 3.1 we presented an approach for data reconstruction at arbitrary sample positions. Following, in Section 3.2 we showed how to

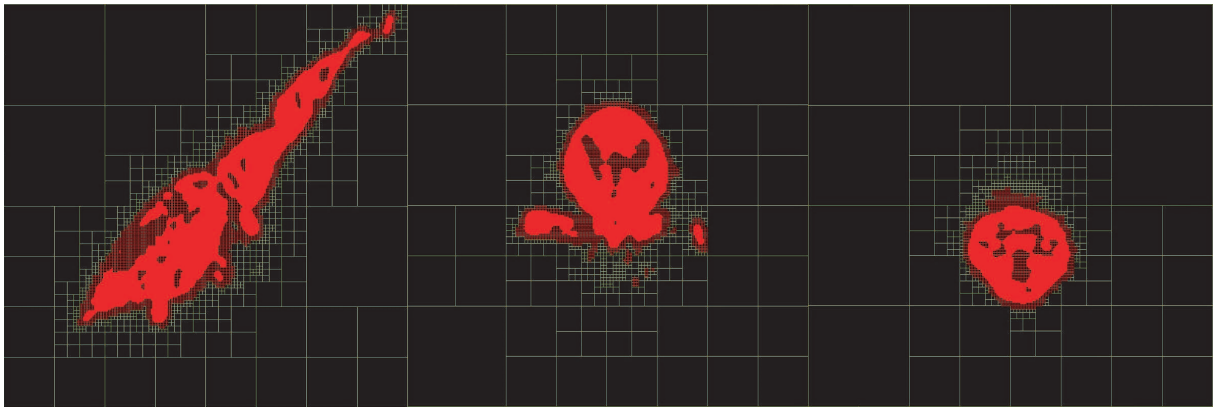

Figure 6: Octree up to depth 8 of the stag beetle. Red octree cells are classified as visible.

determine gradients directly out of the filtered projections at arbitrary sample positions. In Section 3.3 we presented a hierarchical space partitioning data structure for projection-based volumetric data, which is used in the following as acceleration structure for D^2VR. All these approaches do not employ an intermediate grid representation.

We implemented a CPU-based as well as a GPU-based prototype for orthographic and perspective projection. The CPU implementation is based on a raycasting approach. For each image pixel of the image plane, rays are cast through the volumetric space enclosed by the filtered projections. At each resample location the underlying 3D density function is reconstructed according to Equation 2 and gradient estimation is done using Equation 3. The final color and opacity of the image pixel is determined by the over-operator [18] in front-to-back order. The GPU version is implemented in C++ and Cg using OpenGL. We are aware that the current Cg compiler does not always produce optimal code. We did not manually optimize the code by using assembly as this is a proof of concept implementation. We implemented D^2VR on the GPU (NVidia GForce 6800 GT, 256 MB) utilizing texture-based volume rendering with view-aligned slices in combination with filtered projection textures. All the filtered projections are linearly stored in a 3D texture. Basically we compute texture slices parallel to the viewing plane. As the newest NVidia hardware supports dynamic branching, the slices are directly computed in a *for*-loop using Equation 2 from all the filtered projections. The same is done for the gradient computation using Equation 3. The volumetric space enclosed by the filtered projections is thereby cut by polygons parallel to the viewing plane. The polygons are then projected by the graphics hardware onto the image plane and composited using alpha blending. In order to accelerate the rendering process, we utilized the octree described in Section 3.3. We project all visible octree cells onto the image-plane. The resulting z-buffer image is then used for early z-culling, a capability of modern hardware, implementing empty space skipping. In average, for example, we measured for 128 projections (128^2 sized) using a 256^2 view-port four seconds per frame for an iso-surface rendering. The performance benefit depends on the granularity of the octree. Further optimization, such as early ray termination, are not employed due to the lack of graphics hardware support. In future we will investigate alternative acceleration approaches in order to provide fully interactive projection-based volume rendering.

5 Results

In order to show the differences between projection-based and grid-based data reconstruction and gradient estimation we simulated the computed tomographic scanning process. We scanned several different density functions, such as the Marschner & Lobb function, a stag beetle, and a carp. The Marschner & Lobb function is analytically defined. For the other data sets, an analytical representation is not given, therefore we took high resolution grids in combination with trilinear interpolation as ground truth. The Marschner & Lobb function is scanned taking 64 projections, each projection with a resolution of 64^2. From this projections a grid is reconstructed, using the same amount of samples (64^3). Additionally we also reconstructed a grid with eight times more samples (128^3). Furthermore we computed an iso-surface directly from the analytical

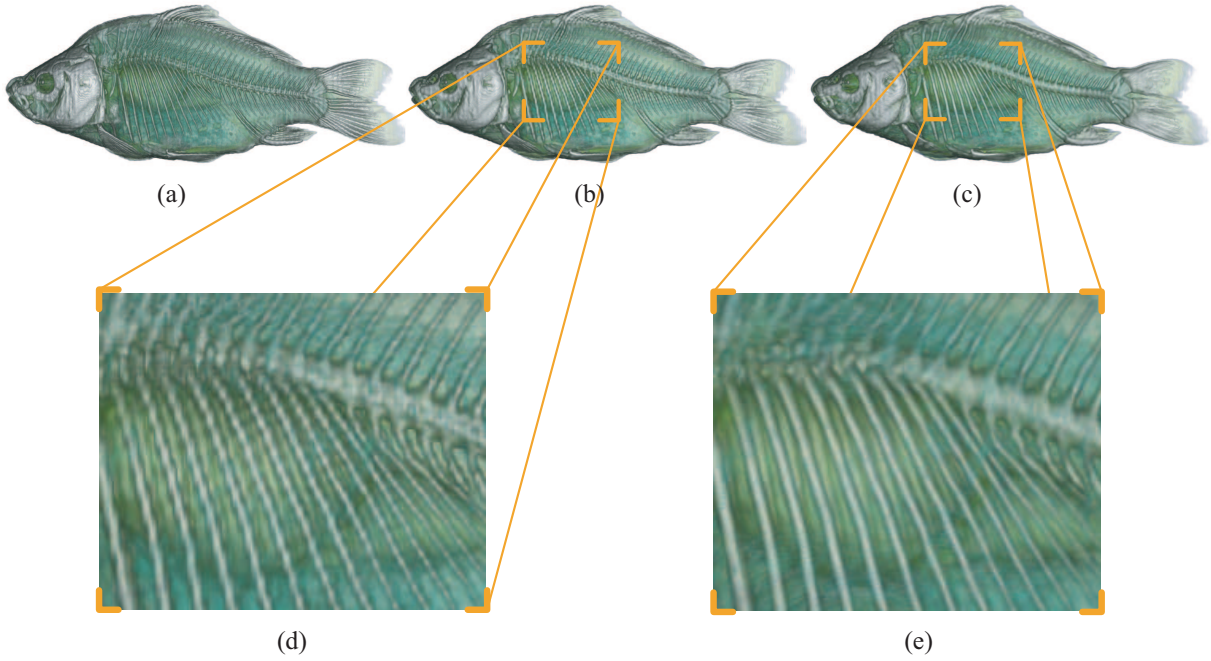

Figure 10: CT scan of Carp (256x256x512): (a) DVR of original grid. (b) DVR of a 128x128x256 grid, reconstructed from 128 filtered projections, each projection with a resolution of 128x256. (c) D^2VR of projection-based volumetric data (128 projections, each projection with a resolution of 128x256). (d) Zoom in of DVR. (e) Zoom in of D^2VR.

Marschner & Lobb function. Figure 7a shows the differences between the analytical value and the reconstructed value using trilinear interpolation on the grid (64^3) and Figure 7b shows the differences between the analytical value and the reconstructed value using Filtered Back-Projection. To encode the data reconstruction error on the iso-surface a color coding is applied. Green encodes low error, on the other hand red encodes higher errors. Figure 8a shows the differences in degrees between the analytically computed gradients and the estimated gradients using central difference gradient estimation. Figure 8b shows the differences in degrees between the analytically computed gradients and the estimated gradients using our new projection-based gradient estimation method. Figure 9 shows a comparison of iso-surface renderings of the Marschner & Lobb function: Figure 9a shows an analytical rendering. Figure 9b shows DVR of the 64^3 grid. Figure 9c shows DVR of the eight times bigger grid (128^3). And finally in Figure 9d our D^2VR from 64 filtered projections, each projection with a resolution of 64^2 is shown. Furthermore, we rendered a stag

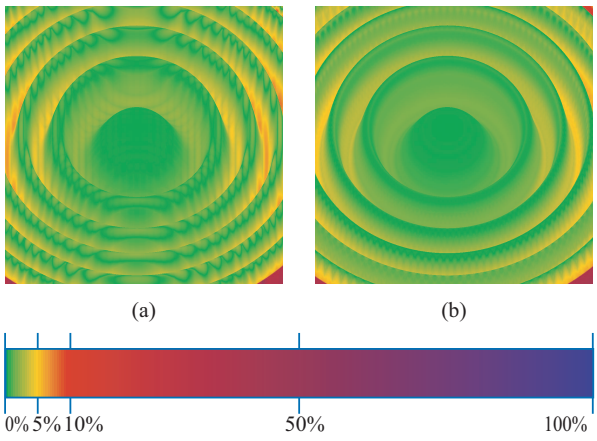

Figure 7: Color encoded differences between analytical value and (a) the reconstructed value using trilinear interpolation on the grid (64^3), (b) the reconstructed value using Filtered Back-Projection.

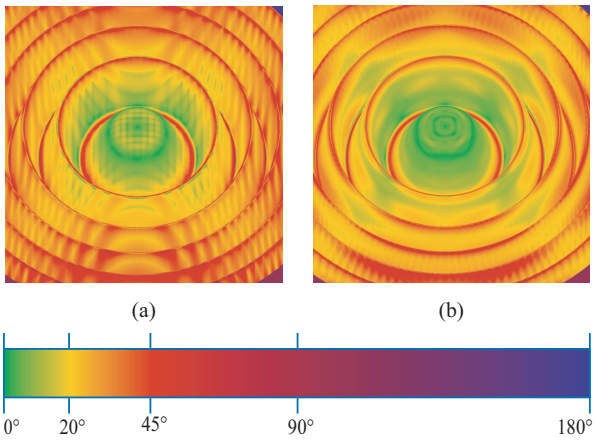

Figure 8: Color encoded differences in degrees between analytically computed gradients and (a) the estimated gradients using central difference gradient estimation on the grid, (b) the estimated gradients using our new projection-based gradient estimation method.

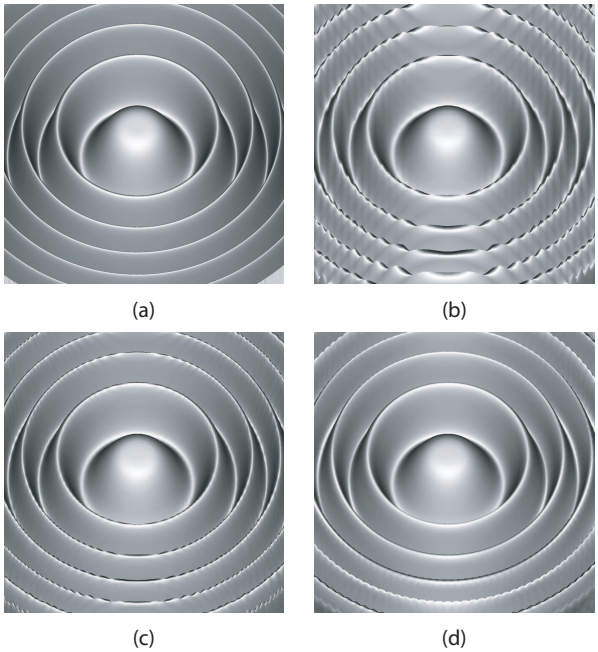

Figure 9: Comparison of an iso-surface of the Marschner & Lobb function: (a) Analytically computed. (b) DVR rendering of a 64^3 grid. (c) DVR rendering of an eight times bigger grid (128^3). (d) D^{2}VR from projection-based volumetric data (64 projections, each projection with a resolution of 64^2). Grids are reconstructed from 64 filtered projections, each projection with a resolution of 64^2.

beetle and a carp, In Figure 1, and Figure 10 the differences between rendering from projection-based and grid-based volumetric data can be seen. Additionally for all the shown data sets we reconstructed a 128^3 grid from the filtered projections. From this grid as well as from the filtered projections we also reconstructed a rotated grid. In Table 1 the root mean squared data reconstruction error with respect to the analytical function respectively to the corresponding high resolution grid in combination with trilinear interpolation is shown.

Data	RMS of grid	RMS of filtered projections	Ratio
Marschner & Lobb	0.0624	0.0537	1.1620
Fish	37.5486	26.9216	1.3947
Stag beetle	80.2559	77.8378	1.0310

Table 1: Root Mean Squared (RMS) data reconstruction error with respect to the analytical function respectively to the corresponding high resolution grid in combination with trilinear interpolation. Values for the Marschner & Lobb data set are between zero and one, and for the other data sets between zero and 4095. In column three the ratio of column one and two is shown.

6 Conclusion and Future Work

In this paper a new direct volume-rendering paradigm has been introduced. It has been shown that volumetric raw data measured as a set of projections can be directly rendered without generating an intermediate grid-based volume representation by using tomographic reconstruction. As our method avoids an unnecessary and lossy resampling step, it provides much higher image quality than traditional direct volume-rendering techniques do. Furthermore, our novel projection-based gradient estimation scheme avoids the accumulation of interpolation errors. Traditional methods ensure accurate samples at the grid points, while the accuracy of intermediate samples strongly depends on the quality of the applied interpolation method. In contrast, our approach provides an accurate data value for an arbitrary sample position.

In order to accelerate D^{2}VR we proposed a hierarchical data structure for empty space skipping. Furthermore, as the filtered projections can be interpreted as 2D textures, the conventional graphics cards can be exploited to efficiently accumulate the contributions of the filtered projections to view-aligned sampling slices. Inspite of these optimizations, our volume-rendering method is

still slower than previous ones. On the other hand, in the last two decades, a huge research effort was spent to accelerate traditional direct volume rendering, which was far from interactivity in the beginning. As our approach is a try to open a new research direction, the current performance of our technique is not yet comparable to that of a well developed technology.

Although our work was inspired by the practical tomographic reconstruction problem, its theoretical significance is the demonstration of an alternative image-based volume representation. In our future work we plan to explore other gridless volume representations, which are not necessarily related to the physical constraints of current scanning devices. For example, in order to achieve full uniformly distributed reconstruction quality, projection planes with uniformly distributed normals might also be applied. Although the adaptation of the Filtered Back-Projection algorithm to such a geometry requires further research, it would lead to a direction independent high-quality volume reconstruction scheme.

7 Acknowledgements

The work presented in this publication has been funded by the ADAPT project (FFF-804544). ADAPT is supported by *Tiani Medgraph*, Vienna (http://www.tiani.com), and the *Forschungsförderungsfonds für die gewerbliche Wirtschaft*, Austria. See http://www.cg.tuwien.ac.at/research/vis/adapt for further information on this project. The stag beetle has been provided by G. Glaeser, University of Applied Arts Vienna. Data scanning is courtesy of J. Kastner, Wels College of Engineering. The carp is courtesy of Michael Scheuring Computer Graphics Group, University of Erlangen, Germany.

References

[1] M. Artner, T. Möller, Ivan Viola, and Eduard Gröller. High-quality volume rendering with resampling in the frequency domain. In *Proceedings of Eurographics / IEEE VGTC Symposium on Visualization*, 2005. to appear.

[2] B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 91–98, 1994.

[3] Q. Chen, R. Crownover, and M.Weinhous. Subunity coordinate translation with Fourier transform to achieve efficient and quality three-dimensional medical image interpolation. *Med. Phys.*, 26(9):1776–1782, 1999.

[4] R. Cox and R. Tong. Two- and three-dimensional image rotation using the FFT. *IEEE Transactions on Image Processing*, 8(9):1297–1299, 1999.

[5] M. Frigo and S. G. Johnson. An adaptive software architecture for the FFT. In *Proceedings of ICASSP*, pages 1381–1384, 1998.

[6] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988.

[7] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications, Vol.8, No.3*, pages 29–37, 1988.

[8] A. Li and K. Mueller. Methods for efficient, high quality volume resampling in the frequency domain. In *Proceedings of IEEE Visualization*, pages 3–10, 2004.

[9] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics (Proceedings of SIGGRAPH '87)*, pages 163–169, 1987.

[10] T. Malzbender. Fourier volume rendering. *ACM Transactions on Graphics, Vol.12, No.3*, pages 233–250, 1993.

[11] S. Marschner and R. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of IEEE Visualization*, pages 100–107, 1994.

[12] D. Mitchell and A. Netravali. Reconstruction filters in computer graphics. In *Proceedings of SIGGRAPH*, pages 221–228, 1988.

[13] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. A comparison of normal estimation schemes. In *Proceedings of IEEE Visualization*, pages 19–26, 1997.

[14] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 143–151, 1998.

[15] L. T. Niklason, B. T. Christian, L. E. Niklason, D. B. Kopans, D. E. Castleberry, B. H. Opsahl-Ong, P. J. Slanetz C. E. Landberg, A. A. Giardino, R. Moore, D. Albagli, M. C. DeJule, P. F.

Fitzgerald, D. F. Fobare, B. W. Giambattista, R. F. Kwasnick, J. Liu, S. J. Lubowski, G. E. Possin, J. F. Richotte, C. Y. Wei, and R. F. Wirth. Digital tomosynthesis in breast imaging. *Radiology*, 205:399–406, 1997.

[16] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall Inc., Englewood Cliffs, 2nd edition, 1989.

[17] D. P. Petersen and D. Middleton. Sampling and reconstruction of wave-number-limited functions in n-dimensional euclidean spaces. *Information and Control*, 5(4):279–323, 1962.

[18] T. Porter and T. Duff. Compositing digital images. *Computer Graphics*, 18(3):253–259, 1984.

[19] J. Russ. *The Image Processing Handbook*. CRC Press, 1992.

[20] T. Theußl and E. Gröller. Mastering windows: Improving reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 101–108, 2000.

[21] R. Tong and R. Cox. Rotation of NMR images using the 2D chirp-z transform. *Magn. Reson. Med.*, 41:253–256, 1999.

[22] M. Unser, P. Thevenaz, and L. Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Transactions on Image Processing*, 4(10), 1995.

[23] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. *Computer Graphics (Proceedings of SIGGRAPH '98)*, pages 169–176, 1998.

[24] L. Westover. Footprint evaluation for volume rendering. *Computer Graphics (Proceedings of SIGGRAPH '90)*, pages 144–153, 1990.

[25] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. EWA volume splatting. In *Proceedings of IEEE Visualization 2001*, pages 29–36, 2001.