

# High-Quality Volume Rendering with Resampling in the Frequency Domain

Martin Artner<sup>†</sup>, Torsten Möller<sup>‡</sup>, Ivan Viola<sup>†</sup>, and Meister E. Gröller<sup>†</sup>

<sup>†</sup>Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria

<sup>‡</sup>School of Computing Science, Simon Fraser University, Vancouver, Canada

---

## Abstract

*This work introduces a volume rendering technique that is conceptually based on the shear-warp factorization. We propose to perform the shear transformation in the frequency domain. Unlike the standard shear-warp algorithm, we allow for arbitrary sampling distances along the viewing rays, independent from the view direction. The accurate scaling of the volume slices is achieved by using the zero padding interpolation property. Finally, a high quality gradient estimation scheme is presented which uses the derivative theorem of the Fourier transform. Experimental results show that the presented method outperforms established algorithms in the quality of the produced images. If the data is sampled above the Nyquist rate the presented method is capable of a perfect reconstruction of the original function.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

---

## 1. Introduction and Related Work

The resampling of discrete signals is an important part of any volume rendering algorithm. The numerically best implementation of volume rendering is the ray casting approach introduced by Levoy [Lev88]. For every pixel in the final image a viewing ray is cast into the scene. Numeric integration of the volume along the viewing rays is performed. This requires the computation of equidistant sample points along each viewing ray. The quality of the resulting image depends directly on the resampling filter used in this stage of the rendering process. Research has been done to improve the design of these spatial domain filters [MMK<sup>\*</sup>98], and to evaluate and compare the quality of the reconstruction [ML94].

As a basic principle all of these filtering methods are approximations of the *sinc* filter, which provides the ideal signal reconstruction. Unfortunately the *sinc* filter has infinite extend in the spatial domain, therefore it is usually dismissed as a theoretical solution. However, the frequency domain representation of the *sinc* filter is a simple box filter. This leads to the assumption that volume rendering with *sinc* filter

quality is possible if the resampling is done in the frequency domain. The Fourier transform was introduced to volume rendering by Dunne et al. [DNR90]. The proposed algorithm, later referred to as frequency domain volume rendering (FDVR), or Fourier volume rendering (FVR), was further established by Malzbender [Mal93], Levoy [Lev92] and Totsuka and Levoy [TL93]. The FVR method is based on the projection slice theorem of the Fourier transform, which states that projection in the spatial domain is equivalent to slicing in the frequency domain. If the size of the volume is  $N^3$ , the computational expense of FVR is  $\mathcal{O}(N^2 \log_2 N)$  as compared to  $\mathcal{O}(N^3)$  of other rendering methods. Therefore the computational complexity of frequency domain volume rendering is lower than that of other traditional volume rendering approaches.

Unfortunately even with the most recent improvements by Lee et al. [LDB96], Westenberger and Roederik [WR00] and Entezari et al. [ESMM02], which have added lighting effects, this method generates only “x-ray” like images. The lack of occlusion and support of transfer functions are the major drawbacks of this method.

In this paper we utilize the shifting theorem, the packing theorem (also known as zero-padding), and the derivative theorem [OS89]. With these theorems we are able to perform

---

<sup>†</sup> {artner | viola | meister} @cg.tuwien.ac.at

<sup>‡</sup> torsten@cs.sfu.ca

volume rendering using the ideal *sinc* filter. Our method is conceptually based on the shear-warp factorization introduced by Lacroute and Levoy [LL94]. The essence of shear-warp is a factorization of the viewing transformation into a permutation, a shear, and a warp transformation. In this work we propose a new method performing the shear transformation in the frequency domain. To further improve the quality of the resulting images, two additional modifications of the standard shear-warp approach are introduced. First, a method is proposed for resampling intermediate slices before the shear operation is applied. This ensures that we obtain a steerable and view-independent sampling distance along the viewing rays. Second, we introduce a technique to perform zooming in the standard object coordinate system as compared to zooming in the warping stage, which improves image quality significantly. Additionally a high quality gradient estimation scheme based on the derivative theorem of the Fourier transform is presented.

During the development of this algorithm a work by Li et al. [LME04] has been published. Their approach uses principles similar to our method, i.e., they perform the resampling of the volume in the frequency domain. As this is the mostly related work to our technique, we include a comparison in terms of complexity between these two methods. We assume that both algorithms are applied on a cubiform dataset with edge length of  $N$ . We also assume  $N$  to be a power of two, allowing a complexity of  $N \log_2 N$  for a 1D Fourier transform. The approach of Li et al. decomposes the transformation matrix into four shear operations, which are performed subsequently in the frequency domain. This requires four sets of 1D forward and backward Fourier transforms with a complexity of  $2N^3 \log_2 N$  each, leading to an overall result of  $8N^3 \log_2 N$ . In our approach the transformation matrix is factored according to the shear-warp factorization which requires only one shear operation to be executed in the frequency domain. To achieve this we need one 3D Fourier transform (complexity  $3N^3 \log_2 N$ ) to get the frequency domain representation of the volume. After the resampling along the principal viewing direction ( $k$  direction),  $N^2$  1D inverse Fourier transforms (complexity  $N^3 \log_2 N$ ) are applied to get slices in the frequency domain. Finally to get the slices for compositing  $N$  2D inverse Fourier transforms (complexity  $N^3 \log_2 N$ ) are used. This all adds up to a complexity of  $6N^3 \log_2 N$ , which is about  $\frac{3}{4}$  of the approach of Li. et al. This is not a very significant difference, but if the volume is resampled by the application of shear operations it is necessary to add sufficient spatial domain zero-padding to fully accommodate the rotated volume. The problem, that arises if the spatial domain zero-padding is too small, is that parts of the data volume pass over the border of the volume and through the periodicity of the dataset enter from the other side. This error is amplified by the consecutive shears. To allow arbitrary positions of the viewpoint, a symmetric spatial domain zero-pad of at least  $\frac{\sqrt{3}}{2}$  times the volume resolution (if the dataset is cubiform) has to be applied in each

direction. This creates a more than five times higher memory consumption as compared to our method that does not require spatial domain zero-padding of that amount. It should be mentioned that our method was especially designed for volume rendering whereas the method of Li et al. was primarily developed for the resampling of a regular 3D volume into another regular 3D volume. In this work we further introduce a gradient estimation scheme that takes advantage of the derivative theorem of the Fourier transform, which could be also applied to their work.

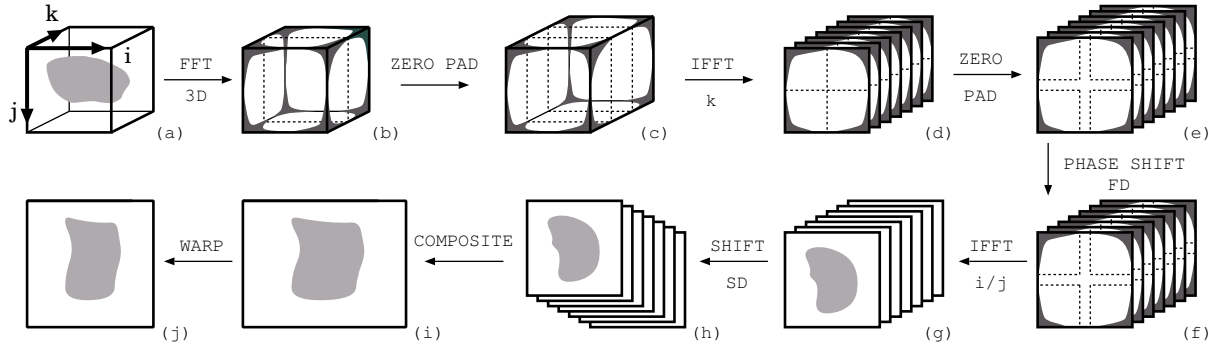
We introduce a new rendering pipeline in the following section. Our results are presented in Section 3 which are followed with conclusions in Section 4.

## 2. The Rendering Pipeline

The proposed algorithm is based on the shear-warp factorization introduced by Lacroute and Levoy [LL94]. The shear-warp technique is one of the fastest software based volume rendering algorithms. It gains its performance by factoring the projection matrix that transforms the volume from object space into image space. The transformation described by each of these matrices can be computed very effectively.

In our work we use the same matrix decomposition as in the original paper. However, our focus is on high reconstruction quality. The shear-warp factorization has four stages: permutation, shearing, compositing and warping. The permutation stage changes the storage order of voxels in memory in order to maximize cache coherency. During the shearing stage the volume is treated as a set of slices which are resampled on a sheared grid. The quality of this resampling process depends very much on the filter used for the reconstruction of the signal. Here we propose to perform the shear by applying the shifting theorem [OS89] in the frequency domain. The next stage, i.e., the compositing, is equal to a numeric integration along the viewing rays. A major drawback of the standard shear-warp technique is that the sampling distance for the numeric integration cannot be changed. This sampling distance is very coarse ( $\geq 1.0$ ) and view dependent. In our rendering pipeline we propose a resampling step that allows to perform the numerical integration along the rays with an arbitrary sampling distance, which is independent of the viewing direction. The last stage, i.e., the warping, transforms the intermediate image of the compositing stage into the final image. The warping in our method remains similar to the standard shear-warp warping. In order to maintain high quality a higher-order spatial domain filter is used. The quality loss through the resampling of the intermediate image in this stage does not create visible artifacts in the final image.

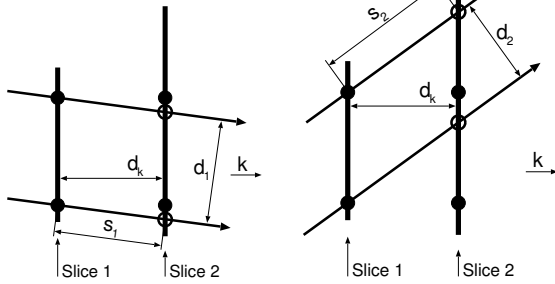
The adapted rendering pipeline for our new frequency domain based method is presented in Figure 1. The first stage of the rendering pipeline starts from the standard object coordinate system, that means the volume is already permuted



**Figure 1:** Rendering pipeline of the shear-warp factorization in the frequency domain.

such that the  $(i, j)$  plane is most perpendicular to the viewing direction. Through a 3D Fourier transform the next pipeline stage is reached (see Figure 1(b)).

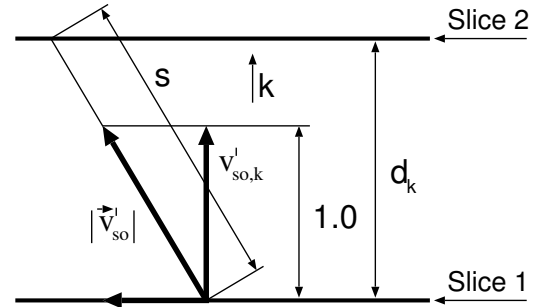
**Resampling in the Principal Viewing Axis ( $k$ ):** In the standard shear-warp factorization the number of volume slices along the principal viewing axis  $k$  is kept constant for performance reasons. Therefore the distance of the sampling points along the rays vary with the viewing direction. Figure 2 shows how the sampling distance ( $s_1$  vs.  $s_2$ ) varies according to the viewpoint settings. Viewing rays are cast from the left, through the samples of the first slice, to the right. The second sample on each ray is interpolated within the second plane (black circles). Only a two dimensional interpolation scheme is required, but an angle dependency of the sampling distance along the rays ( $s_1, s_2$ ) and the distance between the rays ( $d_1, d_2$ ) is introduced.  $d_k$  indicates the distance between two slices in  $k$  direction. The sample distance



**Figure 2:** The shear transform of the shear-warp factorization for two different viewing directions.

variation can create artifacts that are especially visible in animations. Resampling of the volume in  $k$  direction allows to select an arbitrary sampling distance along the rays independent of the viewing direction. Resampling creates additional volume slices and is done by exploiting the packing theorem, which is also known as zero-padding in the frequency domain. The packing theorem states that interpolation in

one domain corresponds to appending zeros in the other domain. To compute the necessary size of the zero-pad area, the sampling distance  $s$  along the viewing rays before resampling is calculated. The setup for this calculation is illustrated in Figure 3. The viewing vector  $\vec{v}_{so} = (v_{so,i}, v_{so,j}, v_{so,k})$  and the distance  $d_k$  between the volume slices along the  $k$  axis are needed.  $\vec{v}_{so}$  is normalized in  $k$  direction by dividing each component by  $v_{so,k}$  yielding  $\vec{v}'_{so} = (v'_{so,i}, v'_{so,j}, v'_{so,k})$  with  $v'_{so,k} = 1.0$ . The absolute length  $|\vec{v}'_{so}|$  of the vector  $\vec{v}'_{so}$  is the sampling distance if the volume slices are 1.0 apart. A multiplication with  $d_k$ , the real distance between the volume slices, gives the sampling distance before resampling.



**Figure 3:** Calculation of the sampling distance  $s$  for a given viewing vector  $\vec{v}_{so}$  and slice distance  $d_k$ .

The sampling distance before resampling  $s$  (Eq. 1) and the desired sampling distance  $s'$  are inversely proportional to  $K$  the number of slices in the  $k$  direction before resampling, and  $K'$  the number of slices in  $k$  direction after resampling (Eq. 2). Using Eq. 3 we calculate  $K'$ , the number of samples after zero-padding. The difference between  $K'$  and  $K$  is the amount of zero-padding necessary for the selected sampling distance  $s'$ .

$$s = d_k \sqrt{(v'_{so,i})^2 + (v'_{so,j})^2 + (v'_{so,k})^2} \quad (1)$$

$$\frac{s'}{s} = \frac{K}{K'} \quad (2)$$

$$K' = \frac{sK}{s'} \quad (3)$$

As zero-padding can only be applied in discrete steps,  $K'$  has to be rounded to the closest integer number  $K'_{pad} \in \mathbb{N}$ . This is typically not an issue, since for values of  $K'_{pad} > 50$  the deviation from the desired sampling distance  $s'$  is below  $\pm 1.0\%$ .

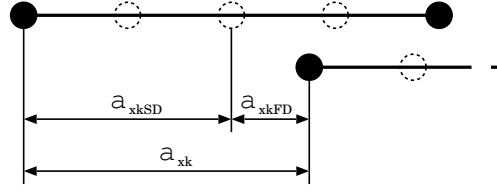
After the application of the zero-pad in  $k$  direction the pipeline stage Figure 1(c) is reached. The next stage is an inverse Fourier transform in  $k$  direction, while the  $i$  and  $j$  direction of the volume remain in the frequency domain (Figure 1(d)).

#### Resampling of Volume Slices in $i$ and $j$ Direction:

In the standard shear-warp factorization zooming is performed by scaling of the intermediate image. This approach leads to considerable blurring artifacts, especially for zoom factors greater than 2.0, as pointed out by Sweeney and Mueller [SM02]. In our method zooming is performed earlier, in the standard object space where the rescaling is applied to the volume slices. A desired slice resolution ( $I'$ ,  $J'$ ) is achieved by increasing the signal period of each slice by zero-padding in the frequency domain. As adding samples is only possible in discrete steps a particular zoom factor can only be achieved with limited precision. The maximum deviation from the desired amount of samples is  $\pm 0.5$  samples, i.e., the difference to the closest full sample count. Therefore, if  $I'_{pad} > 50$  and  $J'_{pad} > 50$  the deviation of the zoom factor is below  $\pm 1.0\%$  of  $I'$  and  $J'$ . This problem is also present for any computer screen, which only has pixel accuracy. Hence, this is a general drawback of any discrete storage-based pipeline. After the application of the zero-pad the render process reaches the stage in Figure 1(e).

**Shearing:** During shearing the volume is transformed from standard object space to sheared object space. This causes the viewing direction to be perpendicular to the slices of the volume. Performing the calculations explained in detail in Lacroute's thesis [Lac95] we acquire the shear coefficients ( $s_i, s_j$ ) and the translation values ( $t_i, t_j$ ). The values  $a_{ik}$  and  $a_{jk}$  describe the displacement of the  $k^{th}$  slice in  $i$  and  $j$  direction. For both directions the shifts by  $a_{xk}$  ( $x \in \{i, j\}$ ) are split into a multiple of the voxel lengths  $a_{xkSD}$  and the remainder, a fraction of a voxel length  $a_{xkFD}$  (see Figure 4).

The shift by  $a_{xkFD}$  is performed in the frequency domain, and the shift by  $a_{xkSD}$  is done in spatial domain. The shift in the spatial domain is actually only a movement of the slice in full voxel steps. The interpolation part is performed in the frequency domain. The reason for this split is to keep the shift in the frequency domain as small as possible to limit wrapping effects. These effects appear because in the Fourier transform the volume slices are assumed to be periodic in  $i$  and  $j$  directions. To limit the wrapping effects a symmetric



**Figure 4:** The shift of the signal by  $a_{xk}$  is split into a spatial domain ( $a_{xkSD}$ ) and a frequency domain ( $a_{xkFD}$ ) fragment.

spatial domain zero-pad of one voxel has to be added to separate the periodic replicas in the spatial domain.

The shifting theorem of the Fourier transform states that a signal can be shifted in the spatial domain by the application of phase shifts in the frequency domain. This theorem is used to perform the  $a_{xkFD}$  shift, and the rendering stage Figure 1(f) is reached. An inverse Fourier transform in  $i$  and  $j$  direction moves the rendering process to stage Figure 1(g). The  $a_{xkSD}$  part of the slice displacement is applied in the spatial domain, which completes the transformation of the volume to the sheared object space (see Figure 1(h)).

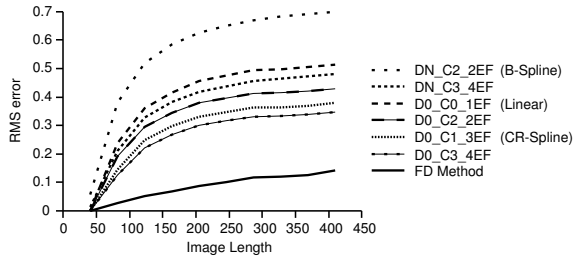
**Compositing:** During the compositing stage the resampled slices are blended into a 2D intermediate image along the  $k$  axis. Transfer functions can be applied to the volume, as well as other visualization techniques (e.g., shading, non-photorealistic effects, texturing, etc.).

During compositing the density information of each slice is transformed into an image. Each pixel in these images has a color and a transparency coefficient. These images are then composited using the “over” operator [PD84], which results in the non-warped intermediate image (see Figure 1(i)).

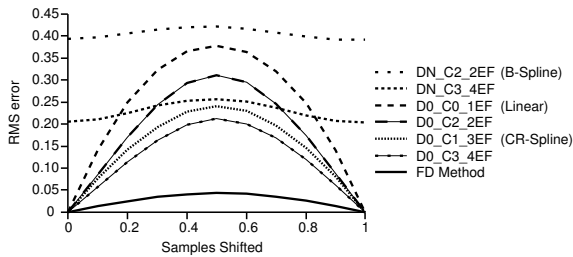
**Warping:** The 2D warping transformation applied to the intermediate image leads to the final image. The warping matrix transforms data points from the sheared object space into the final image space. This transformation compensates the view dependent scaling of the distances between the viewing rays (compare  $d_1$  and  $d_2$  in Figure 2), and performs the rotation component around the  $k$  axis.

If  $M$  is the maximal volume extension (maximum of  $i$ ,  $j$  and  $k$  resolution in standard object space), and  $z$  is the scale factor applied to the scene, then an  $N \times N$  image buffer with  $N = \sqrt{3} \cdot z \cdot M$  can accommodate the resulting image. This calculation is based on the assumption that each side of the volume has length  $M$  and the main diagonal of the volume is visible in its full length. The coordinates of every pixel of the image buffer in the image-coordinate space are transformed to the sheared-object space with the inverse of the warping matrix. The pixel values at the obtained coordinates are interpolated from the pixel values of the intermediate image. In order to maintain high quality, a higher-order spatial domain filter (D0 C3 4EF [MMK\*98] comparable to a Catmull-Rom spline [BBB88]) is used for the resampling.





**Figure 5:** Error when zooming a slice of  $\rho_{ml}$  with several spatial domain filters and by zero padding in the frequency domain.



**Figure 6:** Error when shifting a slice of  $\rho_{ml}$  for sub-pixels with several spatial domain filters and phase shifts in the frequency domain.

**Gradient Estimation:** In order to obtain surface normals for lighting calculations we introduce a high quality gradient estimation scheme. Since the gradient consists of the partial derivatives of the original function and ideal interpolation with the *sinc* filter will reconstruct that function, the gradient can be reconstructed exactly by using the derivate of the *sinc* as a reconstruction kernel [BLM96].

For the computation of the gradient vectors three copies of the original dataset are created and Fourier transformed in all three space dimensions. Each one of these volumes is used to calculate one component of the gradient vector. The derivative theorem of the Fourier transform states that the derivative of a signal in spatial domain can be computed by a multiplication in frequency domain. This theorem is used to derive each volume in one of the three space dimensions. Subsequently these gradient volumes are processed through the same rendering pipeline as the density volume. The gradient volumes are combined to a volume of gradient vectors at the compositing stage. These gradient vectors are used with the processed original data to compute the intermediate image.

### 3. Results

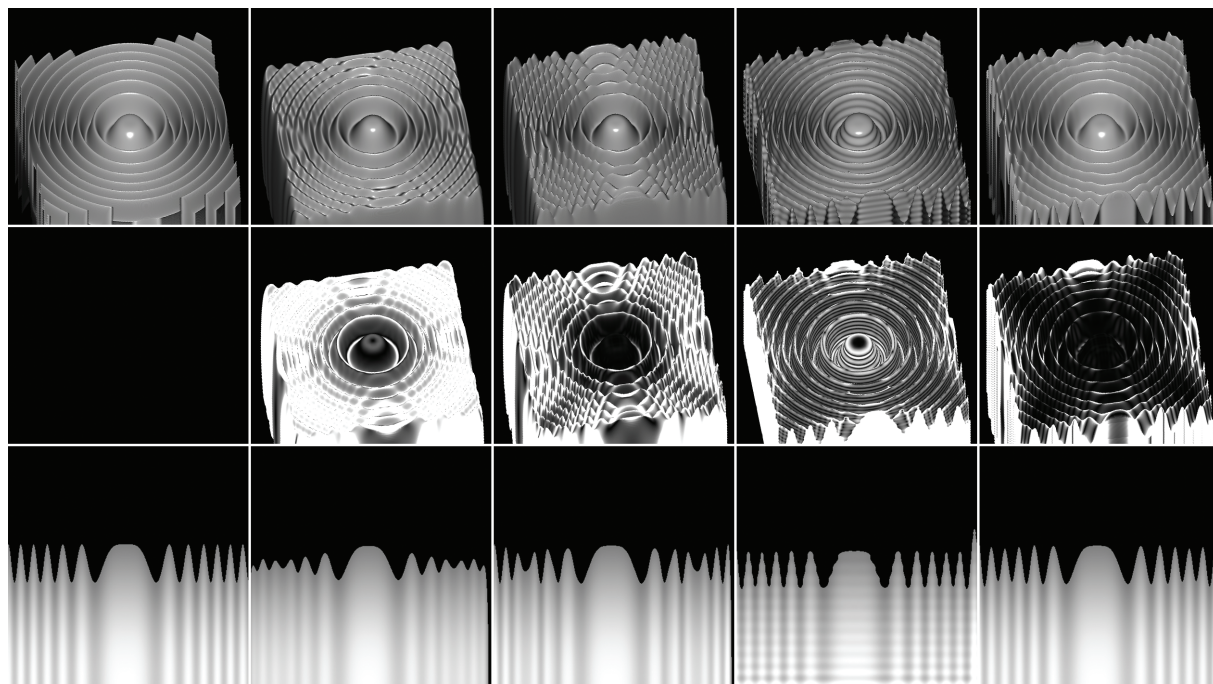
This section presents several experiments to show the advantages and disadvantages of the frequency domain based

techniques as compared to standard spatial domain filtering. 3D datasets of two categories are used for this purpose, synthetic datasets and CT-scans. The naming of the used spatial domain filters was introduced by Möller et al. [MMK\*98]. The convention is that DN identifies an approximation and D0 an interpolation filter. The C value indicates the smoothness of the filter, i.e., C0 would be a linear interpolation. Finally the EF value describes the error degree, i.e., 1EF is a linear and 2EF is a quadratic error filter.

To test the quality of the new interpolation method, the test function introduced by Marschner and Lobb [ML94] is used. In this work we refer to it as  $\rho_{ml}$ . This dataset is sampled almost according to the Nyquist criteria, i.e., 99.8% of the signal energy is captured. When using the discrete Fourier transform (DFT) the assumption is that the signal is periodic and discrete in the spatial and the frequency domain. Through the sampling of  $\rho_{ml}$  we get one period of this signal, which is half the period of a sine wave in  $z$  direction. In  $z$  direction, in the zone between two of these periods, a jump in the signal is present, which is a source of artifacts. In order to demonstrate the impact of this discontinuity, but to maintain compatibility with the original signal  $\rho_{ml}$  a second synthetic dataset was created. The sample points of  $\rho_{ml}$  are mirrored along the  $z = -1$  plane. This creates an extended version of the test dataset with a resolution of  $41 \times 41 \times 80$  samples. The extended test dataset is referred to as  $\rho_{ml\text{ext}}$ .

The original resolution the Stanford Bunny dataset, used in Figure 8, of  $512 \times 512 \times 360$  was downsampled to  $128 \times 128 \times 133$  by removing high frequency components in the frequency domain representation of the datasets. Further, a Hamming window was applied in all three space directions to reduce ringing artifacts. This rough downsampling helps to amplify the resampling artifacts for the different filter methods and therefore makes the quality differences more visible. A second issue is the high memory requirement of the Fourier domain based method, which can be handled easier with a smaller dataset.

**Zooming of 2D slices:** The purpose of this experiment is to compare the quality of zooming by zero-padding in the frequency domain to interpolation with spatial domain filters. As the newly introduced rendering algorithm is based on the shear-warp factorization, most of the rendering steps that are quality critical are performed on volume slices. Initially a  $41 \times 41$  slice of the  $\rho_{ml}$  function at the position of  $z = 0$  was taken. This slice was then zoomed by a factor  $f$  with  $f \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ , in  $x$  and  $y$  direction. The zooming of the volume slices by the factor  $f$  was done by zero-padding in the frequency domain and by resampling with standard spatial domain filters. The synthetic function  $\rho_{ml}$  with  $z = 0$ , was evaluated on an  $f$  times denser grid, to create a reference solution for this zooming operation. The reference solution was subtracted from the resampled slices. To mask out artifacts at the border, after scaling and subtraction of the reference image, a frame of 15 samples width was



**Figure 7:** Quality comparison of the analytic dataset  $\rho_{ml}$  with different reconstruction schemes. The first column shows reference images rendered by analytically evaluating the  $\rho_{ml}$  function. The images of the subsequent columns were rendered (from left to right) based on DN C2 2EF (B-spline) and D0 C0 1EF (Catmull-Rom spline) interpolation, and the new frequency domain based method applied to the  $\rho_{ml}$  and the  $\rho_{ml_{ext}}$  dataset. The first row displays an iso-surface extraction. The second row shows divergence images of the estimated gradient direction to the analytic reference gradient. The third row presents resampled center slices ( $x = 0$ ) of the  $\rho_{ml}$  and  $\rho_{ml_{ext}}$  datasets.

set to 0 in the resulting slices. Finally the root-mean-square (RMS) of these slices was used to draw Figure 5. We can observe that the frequency domain based method (FD method) has the lowest error. If the  $\rho_{ml}$  dataset would have been sampled exactly according to the Nyquist criteria, a perfect scaling with the frequency domain based method would be possible (RMS error of zero).

To visually demonstrate the quality of zooming with these methods, a slice ( $x = 0$ ) of the  $\rho_{ml}$  and the  $\rho_{ml_{ext}}$  dataset was zoomed by a factor of 10.0 in  $y$  and  $z$  direction. To emulate the iso-surface extraction used by Marschner and Lobb [ML94], the range of voxel values  $f(x, y, z)$  from 0.0 to 1.0 was mapped to the gray scale color range 0 to 255. Afterwards the color of the data points with a value of  $f(x, y, z) < 0.5$  was set to black.

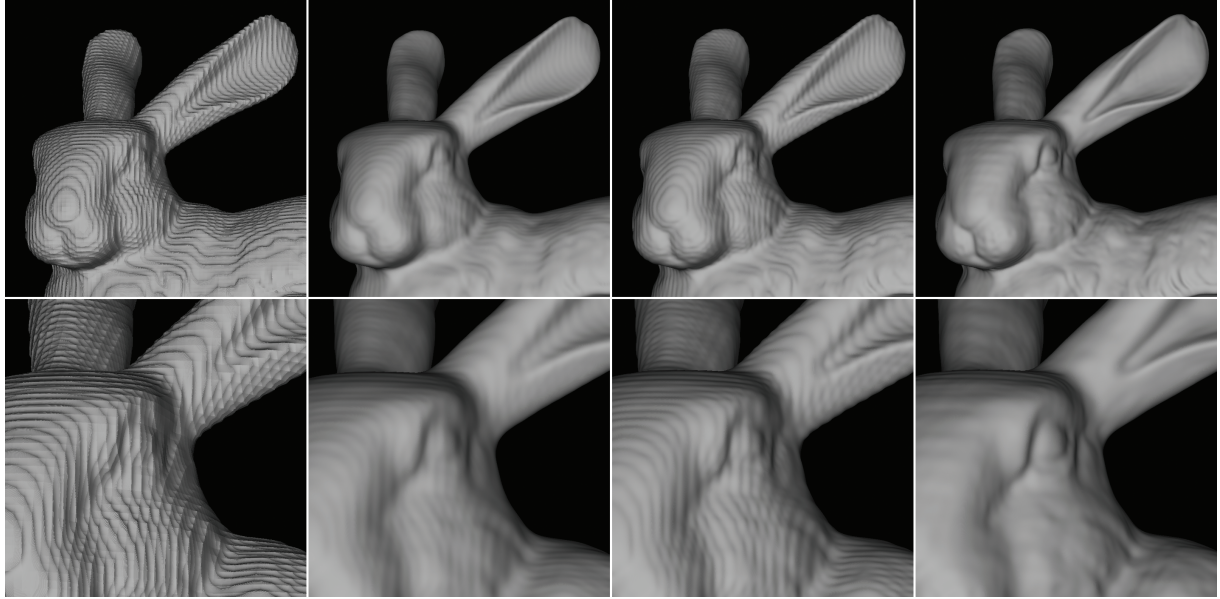
The  $\rho_{ml}$  dataset has a resolution of  $41 \times 41 \times 41$ , and the  $\rho_{ml_{ext}}$  dataset has a resolution of  $41 \times 41 \times 80$ . Therefore a slice taken at  $x = 0$  from  $\rho_{ml}$  has a resolution of  $41 \times 41$ , a slice taken at  $x = 0$  from  $\rho_{ml_{ext}}$  has a resolution of  $41 \times 80$ . The lower half of the  $\rho_{ml_{ext}}$  slice is just a mirror of the upper one. To create slices of the same size, for easier comparison,

the lower half of the slices created from  $\rho_{ml_{ext}}$  was removed after the zooming process (see third row of Figure 7).

**Displacement of 2D slices:** Initially a  $41 \times 41$  slice of the  $\rho_{ml}$  function, with  $z = 0$  was extracted. Sub-pixel shifts where applied in both  $x$  and  $y$  direction. As a reference the function  $\rho_{ml}$  was evaluated shifted by the same amount. To avoid disturbing influences of the border regions, a 7 samples wide border area was set to 0. This size was chosen because it is the biggest filter kernel size (6) used in the spatial domain plus one additional sample. Therefore no sample that was influenced by information outside the  $41 \times 41$  samples affects the result. The RMS of these final error images was then used to draw Figure 6. It is easy to see that the frequency domain based method has the smallest deviation of the synthetic reference result.

**Rendering of 3D datasets:** The resampling and derivative computations are performed with *sinc* filter quality. This filter allows perfect reconstruction of the original signal if the sampling was done according to the Nyquist criteria.

Figure 7 shows images of the synthetic datasets  $\rho_{ml}$  and  $\rho_{ml_{ext}}$ . The first row displays an iso-surface extraction with an iso value of 0.5, a sampling distance of 0.05 and a zoom



**Figure 8:** The Stanford Bunny rendered (from left to right) with D0 C0 1EF (linear), DN C2 2EF (B-spline), D0 C1 3EF (Catmull-Rom spline) interpolation, and the new frequency domain based method. The zoom factor in  $i$  and  $j$  direction for the images of the first row was 10.0 and for the second row 20.0.

factor of 10.0. The second row shows error images of the estimated gradient direction to the analytic reference gradient. The gray value of 255 represents an angular error of at least  $20^\circ$ . The third row presents resampled center slices ( $x = 0$ ) of the  $\rho_{ml}$  and  $\rho_{mlex}$  datasets. The first column shows reference images rendered by analytically evaluating the  $\rho_{ml}$  function. The images of the subsequent columns were rendered (from left to right) with DN C2 2EF (B-spline) interpolation, D0 C0 1EF (Catmull-Rom spline) interpolation, and the new frequency domain based method. For the calculation of the gradients analytic derivatives of interpolation filters in the spatial domain were used. As the  $\rho_{ml}$  dataset is not sampled alias-free, strong ringing artifacts are present in the images rendered with the new frequency domain based method. The  $\rho_{mlex}$  dataset is sampled closer to the Nyquist criteria and therefore the reconstruction quality is higher (compare the last two columns in Figure 7). The spikes are reproduced by the frequency domain method with much higher accuracy than with the spatial domain methods. We see that data, which is not alias-free (e.g.  $\rho_{ml}$ ) will show typical artifacts. Traditional frequency methods (such as windowing) or non-*sinc* filters, that include a smoothing step, should be applied in such cases.

The images in Figure 8 show rendering results of the Stanford Bunny dataset. The dataset is displayed with a 10.0 and 20.0 times zoom, with the sampling distance in  $k$  direction set to 0.05. The images were rendered (from left to right) with D0 C0 1EF (linear), DN C2 2EF (B-spline),

D0 C1 3EF (Catmull-Rom spline) interpolation, and the new frequency domain based method. This figure demonstrates that the rendering quality of the frequency domain based method is superior to standard spatial domain filtering.

We show further volume rendering results using our new frequency domain based method in Figure 9 (see color plates). These pictures demonstrate the practical results that can be achieved with real-world datasets. The result images except the Stanford Bunny image have been created through the application of standard density based transfer functions. An arbitrary volumetric procedural texture has been used to color the iso-surface rendered from the Stanford Bunny dataset.

**Performance:** In the current implementation the frequency domain resampling does not create images at an interactive rate. The main impact on the rendering performance is caused by the global gradient estimation scheme. Gradients are calculated for the whole volume, even if they are just used on a small subset of the voxels, like an iso surface. This leads to a considerable computational overhead that is not contributing to the final image.

In our approach, to perform a zoom of the final image every volume slice is resampled before the projection onto the intermediate image. This creates a noticeable impact on rendering speed, but leads to significant quality improvements in the resulting images. This superior image quality is bought with a computational effort that is proportional to the number of volume slices and is higher than zooming in

the warping stage. Currently, if images are zoomed, and only a small section of interest is to be displayed, the whole image has to be calculated.

#### 4. Conclusions and Future Work

We have developed a volume rendering algorithm that performs resampling using the ideal interpolation filter (*sinc*) and hence achieves the best theoretical quality. This has been achieved by exploiting signal processing ideas, such as the shifting theorem, the packing theorem and the derivative theorem. Using synthetic and real data sets, we have demonstrated that this method can render properly sampled volume data with higher quality than standard spatial domain resampling. We noted that special care is required for data whose periodic extension in the spatial domain would introduce discontinuities. This can always be fixed by a mirrored extension. Further, a high quality gradient estimation scheme that provides very accurate surface normals for lighting calculations has been introduced. Our current implementation has not been performance optimized. There are aspects that have to be addressed in order to improve the practicality of the frequency domain resampling. Li et al. [LME04] have demonstrated that an efficient implementation can outperform traditional spatial domain filtering. The first challenge is to reduce the memory consumption. This could be done by exploiting the fact that the data is given as a real function with a Hermitian Fourier transform. A Hermitian function has symmetries which allow half of the memory to be conserved. The impact on the separability of the Fourier transform on such a data structure has to be investigated. Zooming operations (for the visualization of detailed structures) could be accelerated by using partial inverse Fourier transforms. In order to achieve perspective projection, scaling of volume slices with arbitrary factors has to be developed.

#### Acknowledgments

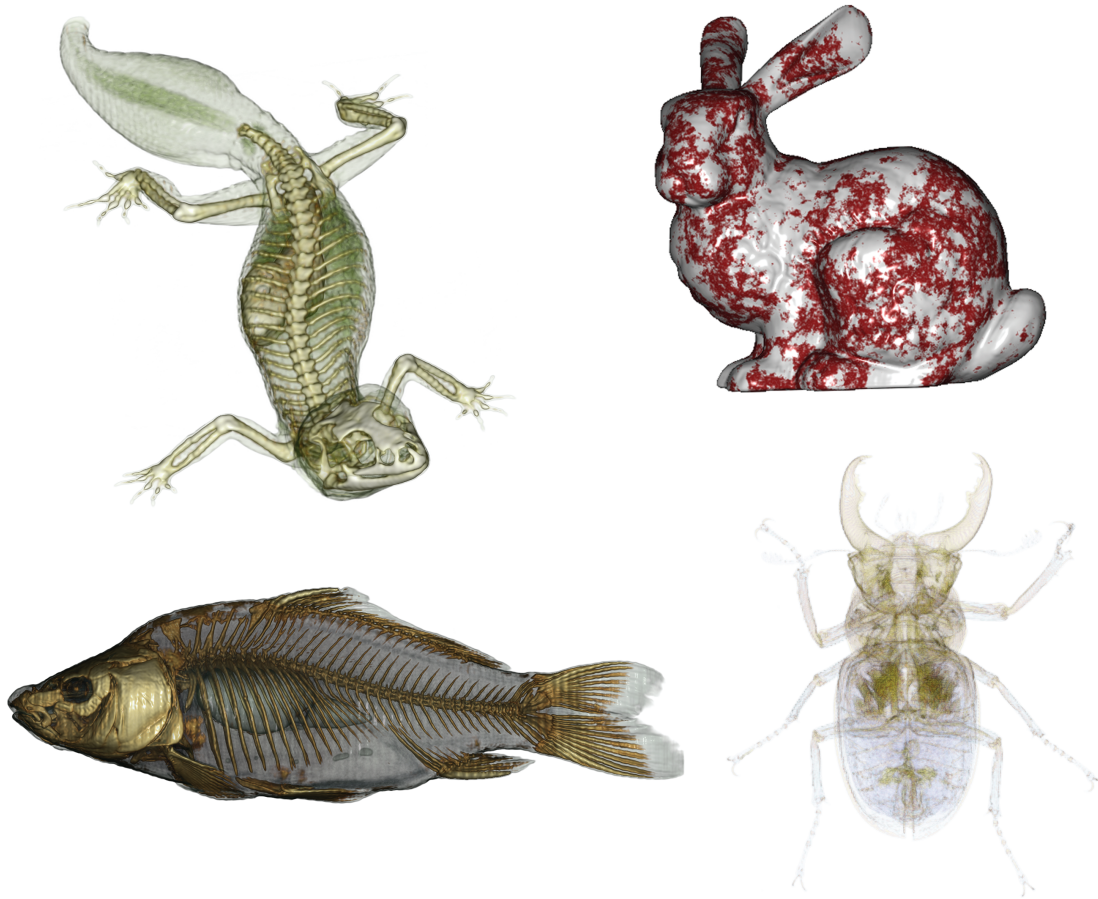
This work was performed during a research stay of the first author at Simon Fraser University. Partial funding of this research was provided by the British Columbia Advanced Systems Institute.

We would like to thank Alireza Entezari and Steven Bergner at the Graphics, Usability and Visualization Lab at Simon Fraser University for their helpful comments and fruitful discussion. We further thank Klaus Mueller at the Department of Computer Science at State University of New York for his remarks on the algorithm complexity.

The Leopard Gecko dataset is courtesy of University of Veterinary Medicine Vienna. The Carp dataset is courtesy of Michael Scheuring of University of Erlangen. The stag beetle has been provided by Georg Glaeser, University of Applied Arts Vienna. Data scanning is courtesy of Johann Kastner, Wels College of Engineering. The Bunny dataset is courtesy of the Stanford volume data archive.

#### References

- [BBB88] BARTELS R., BEATTY J., BARSKY B.: *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. Morgan Kaufmann, Palo Alto, CA, 1988.
- [BLM96] BENTUM M. J., LICHTENBELT B. B. A., MALZBENDER T.: Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 2, 3 (1996), 242–254.
- [DNR90] DUNNE S., NAPEL S., RUTT B.: Fast reprojection of volume data. In *Proceedings of the 1st International Conference on Visualization in Biomedical Computing* (1990), pp. 11–18.
- [ESMM02] ENTEZARI A., SCOGGINS R., MÖLLER T., MACHIRAJU R.: Shading for Fourier volume rendering. In *Proceedings of the 2002 IEEE Symposium on Volume Visualization* (2002), IEEE, pp. 131–138.
- [Lac95] LACROUTE P. G.: *Fast volume rendering using a shear-warp factorization of the viewing transformation*. PhD thesis, Computer Systems Laboratory, Department of Electrical Engineering and Computer Science, Stanford University, 1995.
- [LDB96] LEE Z., DIAZ P. J., BELLON E. M.: Analysis of Fourier volume rendering. In *IEEE Proceedings of the 1996 Fifteenth Southern Biomedical Engineering Conference* (1996), pp. 469–472.
- [Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Computer Graphics and Applications* 8, 3 (1988), 29–37.
- [Lev92] LEVOY M.: Volume rendering using the Fourier projection-slice theorem. In *Proceedings of Graphics Interface '92* (1992), pp. 61–69.
- [LL94] LACROUTE P. G., LEVOY M.: Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics* 28, Annual Conference Series (1994), 451–458.
- [LME04] LI A., MUELLER K., ERNST T.: Methods for efficient, high quality volume resampling in the frequency domain. In *Proceedings of the 2004 IEEE Visualization* (2004), IEEE, pp. 3–10.
- [Mal93] MALZBENDER T.: Fourier volume rendering. *ACM Transactions on Graphics* 12, 3 (1993), 233–250.
- [ML94] MARSCHNER S. R., LOBB R. J.: An evaluation of reconstruction filters for volume rendering. In *Proceedings of Visualization '94* (1994), IEEE, pp. 100–107.
- [MMK\*98] MÖLLER T., MUELLER K., KURZION Y., MACHIRAJU R., YAGEL R.: Design of accurate and smooth filters for function and derivative reconstruction. In *Proceedings of the 1998 IEEE Symposium on Volume Visualization* (1998), pp. 143–151.
- [OS89] OPPENHEIM A. V., SCHAFER R. W.: *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, USA, 1989.
- [PD84] PORTER T., DUFF T.: Compositing digital images. *Computer Graphics* 18, 3 (1984), 253–259.
- [SM02] SWEENEY J., MUELLER K.: Shear-warp deluxe: The shear-warp algorithm revisited. In *Proceedings of the 2002 Symposium on Data Visualisation* (2002), Eurographics Association, pp. 95–104.
- [TL93] TOTSUKA T., LEVOY M.: Frequency domain volume rendering. *Computer Graphics* 27, Annual Conference Series (1993), 271–278.
- [WR00] WESTENBERG M. A., ROERDINK J. B. T. M.: Frequency domain volume rendering by the wavelet x-ray transform. *IEEE Transactions on Image Processing* 9, 7 (2000), 1249–1261.



**Figure 9:** These images show results of rendering CT datasets. They have been created with the new frequency domain based method. These pictures demonstrate the practical results that can be achieved with real-world datasets. The result images except the Stanford Bunny image are created with the application of standard density based transfer functions. An arbitrary volumetric procedural texture was used to color the iso-surface rendered from the Stanford Bunny dataset.