Adaptive GPU Ray Casting Based on Spectral Analysis

Stefan Suwelack, Eric Heitz, Roland Unterhinninghofen, Rüdiger Dillmann

Adaptive GPU Ray Casting Based on Spectral Analysis

Stefan Suwelack[1], Eric Heitz[1] Roland Unterhinninghofen[1], and Rüdiger Dillmann[1]

Institute of Anthropomatics (IFA) Humanoids and Intelligence Systems Laboratories (HIS), Karlsruhe Institute of Technology (KIT), Germany,
`suwelack@kit.edu`

Abstract. GPU based ray casting has become a valuable tool for the visualization of medical image data. While the method produces high-quality images, its main drawback is the high computational load. We present a novel adaptive approach to speed up the rendering. In contrast to well established heuristic methods, we use the spectral decomposition of the transfer function and the dataset to derive a suitable sampling criterion. It is shown how this criterion can be efficiently incorporated into an adaptive ray casting algorithm. Two medical datasets, which each represent a typical, but different material distribution, are rendered using the proposed method. An analysis of the number of sample points per ray reveals that the new algorithm requires 50% to 80% less points compared to a non-adaptive method without any quality loss. We also show that the rendering speed of the GPU implementation is greatly increased with reference to the non-adaptive algorithm.

Keywords: volume rendering, gpu raycasting, adaptive sampling, spectral analysis

1 Introduction

The 3D visualization of computer tomographic (CT) data by means of real-time volume rendering techniques is widely used in clinical diagnostics and medical education [10] [12]. In order to obtain meaningful and visually pleasant 3D images, the gray values of the CT dataset are assigned different optical properties (color values) through the so called transfer function. The final image is then obtained by evaluating the rendering integral based on the output of the transfer function. A very powerful and popular approach to do so is to perform ray casting on the graphics processing unit (GPU) [7]. This method allows the interactive rendering of moderately sized datasets on a standard desktop PC.

Many different approaches have been proposed to increase the efficiency of GPU based ray casting. In order to reduce the memory consumption on the GPU side Gobetti et al. presented an out-of-core technique based on a view-dependent octree data structure [4]. A similar approach is used by Crassin et al. to render

gigavoxel datasets [2]. The computational load can be reduced by adapting either the number of rays (image based adaptivity) or the number of sample points for each ray (object based adaptivity). While image based adaptive methods can be effective in combination with level-of-detail methods [8] we will focus here on object based adaptive sampling.

Even very early GPU based ray casting algorithms employed empty space skipping and early ray termination to reduce the sample points per ray [7]. Subsequent implementations such as the vertex shader based approach from Scharsach et al. improved the performance of the empty space skipping technique [13]. A full adaptive sampling algorithm based on a so-called importance volume was proposed by Roettger et al. [11]. Recently presented ray casting engines often rely on multi-resolution data structures and therefore naturally embed full adaptive sampling. However, very little attention has been paid to the criteria that control the sampling distance. The sampling distance is either chosen according to a given multi-resolution representation of the volume [8] or heuristic properties such as the gradient of the dataset [6].

Mathematically sound adaptivity criteria can be obtained by applying a Fourier analysis [1]. The sampling distance then follows through the Nyquist-Shannon sampling theorem. If non-linear transfer functions are used (which is the case in medical applications), it is not sufficient to apply the spectral decomposition just to the scalar data $f(x)$ [5]. In order to capture high-frequency characteristics of the transfer function $g(x)$ the complete signal $(g \circ f)(x) = g(f(x))$ has to be analyzed instead. However, the transfer function usually changes during the rendering to highlight different anatomical regions of interest. Therefore, an online Fourier decomposition of the whole dataset would be necessary in order to determine the Nyquist rate as soon as the transfer function changes. This is not feasible due to the large computational effort associated with this analysis.

In order to alleviate this constraint, several authors proposed a technique called pre-integration of the transfer function [3] [14]. Based on the assumption that the scalar function is linear between the two sample points s_1 and s_2, the volume integral is determined analytically in this interval. The integral values are pre-computed and can be accessed via a lookup table for all values of s_1 and s_2. In this way high frequency behavior of the transfer function can be absorbed and the sampling rate can be reduced [5].

Unfortunately, the approach does not reduce the sampling rate where small changes of g actually reduce the maximal frequencies of the complete signal $g \circ f$. Furthermore, the signal reconstruction with the ray casting technique introduces further errors that deteriorate the quality of the pre-integrated solution. This issue will be discussed in detail in the next chapter. Bergner et al. showed a way to overcome these limitations [1]. Drawing from well established results in the field of signal processing they derive a mathematically rigorous estimation for the spectral decomposition of $g \circ f$. The key result establishes an estimate for the Fourier analysis of $g \circ f$ based on the Fourier analysis of g and the local gradient $f'(x)$.

We build upon the work of Bergner et al. and present a GPU based ray casting algorithm that incorporates object-space adaptive sampling based on a spectral analysis of $g \circ f$. In chapter 2 we describe the errors that are introduced when applying ray casting for 3D image reconstruction. We subsequently explain the relation between the Nyquist rate and the sampling frequency. In chapter 3 we show how different local behavior of the transfer function can be incorporated in the approach. Most importantly, we present how dynamic transfer functions can be used in the adaptive framework with minimal computational effort. A memory efficient implementation of the method is presented in chapter 4. We also analyze the 3D rendering of typical medical CT datasets. The performance of the new method is compared to the theoretical optimal adaptive algorithm and to a non-adaptive approach. We conclude with a summary of key contributions of this paper and give a short outlook for future work in chapter 5.

2   Application of sampling theory to ray casting

A CT dataset can be regarded as a set of samples that have been taken from a band limited, continuous signal. According to the Nyquist-Shannon sampling theorem, the sampling frequency must be at least twice as high as the highest frequency of the original signal. Thus, for an isotropic sample distance of T between the voxels, the maximal frequency v_f that is represented in the dataset is given by $v_f = 1/2T$. The signal can be perfectly reconstructed by sampling the dataset once per voxel and then performing a convolution with a sinc filter kernel. However, additional errors are introduced, if ray casting is used to reconstruct the signal. These errors are discussed in the following paragraph by means of 1D examples. However, all results can be easily generalized to the 3D case.

The ray casting method essentially solves the rendering integral by adding the volume densities along each viewing ray. Usually tri-linear interpolation is employed to calculate the density values at the sample points. From a mathematical perspective, this approach corresponds to a reconstruction with a linear ('tent') filter kernel [9]. The errors that occur due to the reconstruction with the simplified filter kernel are referred to as filtering artifacts . We note two important properties of this error source. First of all, the filtering artifacts cannot be reduced by raising the sampling rate. This is quickly illustrated for a 1D signal (see fig. 1). If the samples are always taken exactly in the center of each voxel, there is no additional benefit from sampling the signal with a higher frequency than once per voxel. Secondly, in contrast to the sinc filter, which has an infinite support, the tent filter's support is limited to adjoining voxels. Thus, the required sampling rate only depends on the local behavior of the material density. This is a fundamental property that permits the use of local frequency based criteria to control the adaptive sampling.

In the previous discussion we assumed that the sample points are always placed in the centre of each voxel. However, during ray casting the samples are taken along the viewing ray at different positions that not necessarily coincide with the midpoint of the voxels. Fig. 1 illustrates the error that emanates from
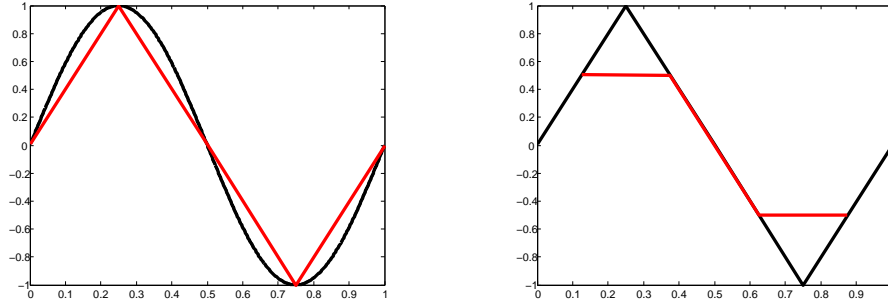
Fig. 1. A sinusoid signal (black) is reconstructed with a ray casting method (red). The filter kernel creates filtering artefacts (left). Samples taken with an offset introduce another error source (right), which can be reduced by performing oversampling.

this procedure. The size of the error depends on the frequency of the signal, the sampling rate and the actual sample point, which cannot be known a-priori. Although the error scales with the Nyquist frequency of the signal, the simple 1D example (fig. 2) already suggests that it is often not sufficient to sample with the Nyquist rate. This is in fact the case for volume rendering of medical datasets. Previously published results suggest that an oversampling by a factor of two to four with respect to the Nyquist frequency is enough to remedy the problem [11]. In accordance with these results we use an oversampling by a factor of two in our implementation.

In order to generate visually pleasant and meaningful images, a transfer function g is used to assign optical properties to the Hounsfield values stored in the CT dataset. Thus, the sampling frequency does not only depend on the data values f, but rather on the composite function $g \circ f$. This is especially important, if the transfer function has high frequency components. In this case, a high sampling frequency is necessary even if there is only little variation in f. A method to overcome this problem is to use pre-integration as previously introduced. However, it has to be noted that the sampling error discussed in the previous paragraph can be considerably amplified by a high-frequency transfer function. Even with the use of pre-integration, small errors in the sampling of f can lead to large errors for the composite function $g \circ f$. A good frequency based sampling criterion can therefore not be based on the spectral analysis of f alone, but rather on the analysis of $g \circ f$. A detailed analysis of the influence of transfer function on the sampling frequency can also be found in [14].

3   Spectral Analysis

The direct spectral decomposition of the composite function $g \circ f$ (e.g. by the discrete Fourier transform) is a very computational intensive procedure. If the transfer function does not change during the rendering this analysis can be done in a pre-processing step. However, in medical applications the transfer function

is usually changed during the rendering to browse through different anatomical structures and regions of interest. In this section we present an efficient approximation for the spectral decomposition that can be evaluated with very little computational effort. This allows the use of frequency based adaptivity criteria even for transfer function that change dynamically during the rendering.

3.1 Spectral analysis of the composite function

Bergner et al. showed that the maximal frequency v_h of the composite function g(f(x)) can be approximated by

$$v_h = v_g \max_x |f'(x)|,\tag{1}$$

where v_g denotes the maximum frequency of g [1]. This approximation is not tight which means that the actual maximal frequency might be higher than estimated. However, Bergner et al. showed that the energy of the signal decays exponentially for frequencies above this estimation. In accordance with these results our experiments showed that no visible artifacts occur even when no additonal oversampling is used.

While the estimate above holds for all volume rendering techniques, it can be significantly enhanced for the ray casting technique due to the locality of the reconstruction filter. When approximating the frequency to determine the step size from sample point s_1 to sample point s_2 it is obvious that the maximal gradient of $f'(x)$ should be determined over the interval $[s_1, s_2]$ and not over the whole volume. Additionally, we propose to determine the maximal frequency of g not over the entire range of Hounsfield values, but rather over these values $I = [s_{\min}, s_{\max}]$ that actually occur in $[s_1, s_2]$. The benefits of this approach become quickly apparent when looking at the transfer functions that are typically used for medical visualization (see fig 2. There are large areas where the transfer function has minimal variations whereas it changes substantially around a certain threshold value. The step size can therefore be reduced if all values of f in $[s_1, s_2]$ only induce little changes in the transfer function. Using the approximation 1 the sampling period is given by

Fig. 2. A typical non-stationary opacity transfer function used for medical visualization (left). The transfer function is shifted along the x-axis to browse through different anatomical structures of a human hand (middle, right).

$$T = \frac{1}{2v_g(I) \cdot \max |f'(x)|} \tag{2}$$

where $v_g(I)$ is the maximal frequency of g for all values in I. This approximation can be very efficiently implemented as will be detailed in chapter 4. We will discuss next how the maximal frequencies of g are calculated.

3.2 Dynamic transfer functions

The spectral decomposition of g is obtained by applying a fast Fourier transform (FFT). We define the cut-off frequency that contains a pre-defined percentage of the signal's energy as the maximal frequency of g. If the transfer function changes arbitrarily during the rendering, the FFT has to be applied again. Although the FFT is a very efficient algorithm, this cannot be done in real-time for high-resolution transfer functions. To overcome this problem, we make use of the fact that in medical applications the transfer function is only changed in a very restricted way during user interaction.

Medical imaging workstations (e.g. Vital Vitrea, OsiriX Imaging Software) typically allow the user to change a 1D transfer function with simple mouse gestures. In order to highlight different anatomical structures (see fig. 2) all four RGBA components of the transfer function $g(x)$ are shifted along the x-axis or multiplied by a constant factor. This behavior can also be modelled by defining the affine map

$$t \mapsto \Psi(t) = a_\Psi t + b_\Psi \tag{3}$$

and using this map to adjust the material density f before the transfer function g is applied. To determine the maximal frequency for a modified transfer function we have to analyse the function composition $g \circ \Psi \circ f$ instead of $g \circ f$. By noting that

$$(\Psi \circ f)' = (a_\Psi f + b_\Psi)' = a_\Psi f' \tag{4}$$

we can define the new estimate

$$T = \frac{1}{2v_g(I) \cdot |a_\Psi| \max |f'(x)|} \tag{5}$$

to incorporate the change of the transfer function into the approximation. This allows to approximate the maximal frequency of the modified transfer functions with minimal computational effort by using the spectral decomposition of the original transfer function. A FFT on g has to be performed only if a new transfer function is chosen, but not if the transfer function is altered in the way described above. It is important to remark that this perfectly fits the typical use case for medical visualization. A completely new transfer function is usually chosen only in the beginning of the examination after loading the dataset. At this stage the delay caused by the calculation of the FFT is negligible. The change of the transfer function during the interactive rendering can then be modelled as described above.

4 Implementation and Results

4.1 Implementation

The ray casting algorithm was implemented as a module for the visualization toolkit (VTK) using the C++ and OpenGL/GLSL programming languages. We use pre-integration and store the necessary pre-computed data in a 2D texture. In order to implement the adaptive sampling using the frequency based criterion, two additional textures are used. A 2D texture holds the maximal frequency $v_g(I)$ of g in the interval $I = [s_{\min}, s_{\max}]$. As detailed above these frequencies can be calculated in a pre-processing step by performing a spectral decomposition of g using the FFT.

The gradients $f'(x)$ are stored in an importance volume that is of the same size as the dataset. For each voxel the gradient of f is determined in each spatial direction. It is important that the adaptivity criterion estimates the maximal frequency of the whole region that is covered in the next sampling step. In order to achieve this, the gradients are filtered with a max-filter with a width of $2T_{\max}$ and the result is stored in the importance volume. Here, $T_{\max}$ denotes the maximal step size. Please note that this procedure essentially implements an isotropic sampling criterion. Further speed-up could be achieved if the direction of the ray was taken into account.

In order to estimate the step size we also need the values $s_{\min}$ and $s_{\max}$ in the region covered by the ray in the next sampling step. Although it is possible to store these values in an importance volume similar to the gradients $f'(x)$, this approach would be very memory intensive. To circumvent this problem, we approximate f with its Taylor expansion

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \tag{6}$$

around the sample point x_0. The upper and lower bound for $s_{\min}$ and $s_{\max}$ can then be calculated using the relations

$$s_{\min} = f(x_0) - f'(x_0)T_{\max} \tag{7}$$

$$s_{\max} = f(x_0) + f'(x_0)T_{\max} \tag{8}$$

and the gradients stored in the importance volume.

We conclude the description of the implementation by summarizing the additional computational effort that is associated with the adaptive method. The algorithm requires some additional floating point operations to calculate the interval $I = [s_{\min}, s_{\max}]$ values according to equation 8. Due to the fact that the transfer function changes during the rendering, the interval has to be shifted as described by the relation 3. Finally, the evaluation of the step size using the frequency approximation (see eqn. 5) has to be performed. Two additional texture fetches have to be carried out in order to obtain the gradient f' and the local maximal frequency of v_g. Most significantly, the adaptive method uses more texture memory to store the importance volume. Our implementation uses an importance volume that is of the same size as the original dataset. However, it has to be pointed out that it contains a lot of redundancy due to the use of the max-filter and can therefore be easily compressed.

4.2   Results

In this section the rendering performance of the algorithm for two typical medical datasets is outlined. The first one is a CT angiography of a human hand (fig. 3) featuring a resolution of 300x300x300 voxels. The second one is a section of an abdominal CT scan with a resolution of 512x512x101 voxels (fig. 4). The test system is equipped with a NVIDIA GeForce GTX 280 graphics card with 1GB of video memory. We use a maximal step size of $T_{\max} = 8$ and an oversampling by a factor of two as discussed in chapter 2.
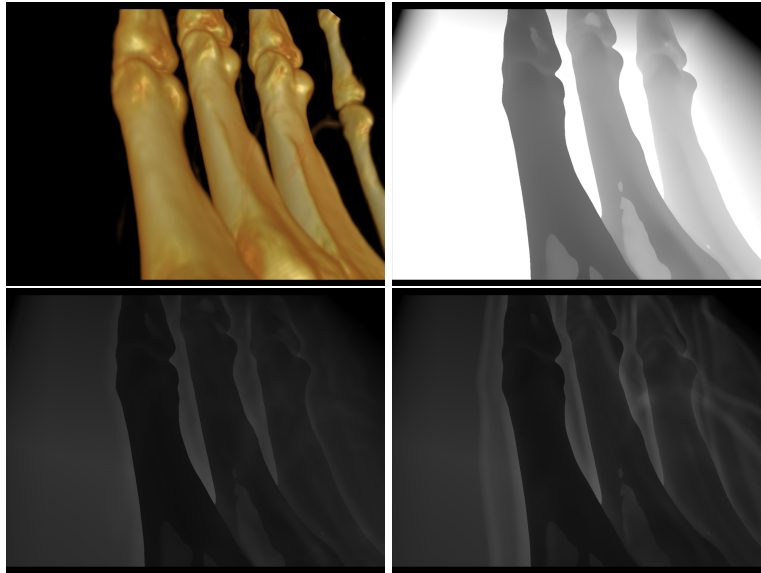

Fig. 3. Close-up view of the human hand dataset (upper left). Black and white images indicate how many samples are taken per ray if regular sampling (upper right), direct sampling (lower left) or the proposed sampling criterion (lower right) is used.

The regular, non-adaptive algorithm is used to render a reference solution. The transfer function is not changed during the rendering. This allows to compute the spectral decomposition of $g \circ f$ in a pre-processing step using the FFT. The solution that is obtained using the pre-computed frequencies is referred to as direct sampling. It serves as the theoretical upper bound for the average step size. As explained in chapter 3, direct sampling cannot be used when the transfer function changes during the rendering. In order to evaluate the performance of the adaptive method we compare the average number of sample points per ray as well as the number of frames per second (FPS). Additionally, the number of sample points for each ray are plotted as black and white images (see fig. 4, 3). Many sample points are used within white regions, whereas darker regions indicate that fewer samples have been taken.
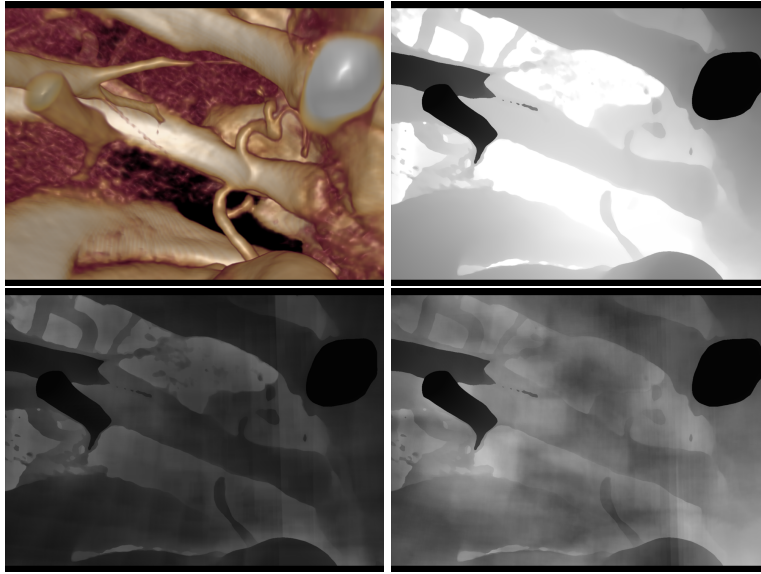
Fig. 4. Close-up view of the abdomen dataset (upper left). Black and white images indicate how many samples are taken per ray if regular sampling (upper right), direct sampling (lower left) or the proposed sampling criterion (lower right) is used.

No visible difference is observed between the images rendered with the different methods. The average number of sample points is reduced from 430 to 88 samples per pixel for the rendering of the hand dataset (see table 1). This is only slightly more than the optimal solution (75 samples/pixel). A detailed analysis (see fig. 3) reveals that most of the speed-up can be attributed to empty space skipping around the bones of the hand. In contrast to this scenario, the image generated from the abdominal CT scan does not contain much empty space. The adaptive method significantly reduces the number of sample points in all image areas which results in an average redution of more than 50% (fig. 4). However, the approximation still needs twice as many sample points as the optimal solution.

Sampling type	regular	direct	approx.
Samples/pixel	430	75	88
FPS	1.2	6.5	5.5

Sampling type	regular	direct	approx.
Samples/pixel	752	197	404
FPS	5	8.1	6.2

Table 1. Performance of the proposed approximation in terms of samples per ray (less is better) and frames per second (FPS) for the hand dataset (left) and the abdomen dataset (right). Results using regular sampling and direct sampling are given as reference values.

The adaptive method significantly increases the rendering speed. The first example shows that the acceleration factor perfectly corresponds to the reduction of the sample points. This shows that the additional computations for the adaptive method only have a very small impact. In the second example the number of samples per pixel does not scale linearly with the rendering speed which can be attributed to the parallel execution on the GPU. This suggests that an implementation wich allows a finer control over the hardware (e.g. using NVIDIA CUDA) could further accelerate the algorithm.

5 Conclusion and Outlook

We have presented a novel adaptive ray casting algorithm. In contrast to well established heuristic methods, we use the spectral decomposition of the transfer function and the dataset to derive a suitable sampling criterion. The analysis of the number of sample points per ray reveals that the new algorithm requires 50% to 80% less points compared to a non-adaptive method without any quality loss. Furthermore, the approximation of the maximal frequency of $g \circ f$ performs well compared to the optimal solution. Because of the numerical efficiency of the method, the rendering speed of the GPU is greatly increased with reference to the non-adaptive algorithm. The presented adaptivity criterion can be effectively used in a framework that uses a hierarchical data structure to perform adaptive ray casting (e.g. Gobetti et al. [4]). Such a data structure also allows to efficiently store the dataset gradients which reduces the necessary texture memory, thereby alleviating the only major disadvantage of the method. We employed the fast Fourier transform to obtain the spectral decomposition over certain regions of g. It could be beneficial to employ a different transform that has a better spatial resolution such as a Wigner-Ville transform. Finally, the criterion could be further improved further by considering anisotropic behaviour of the dataset.

References

1. S. Bergner, T. Moller, D. Weiskopf, and DJ Muraki. A spectral analysis of function composition and its implications for sampling in direct volume visualization. *IEEE transactions on visualization and computer graphics*, 12(5):1353–1360, 2006.
2. C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann. Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 15–22. ACM, 2009.
3. K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, page 16. ACM, 2001.
4. E. Gobbetti, F. Marton, and J.A. Iglesias Guitián. A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets. *The Visual Computer*, 24(7):797–806, 2008.

5. M. Kraus. Direct volume visualization of geometrically unpleasant meshes. 2003.

6. M. Kraus, M. Strengert, T. Klein, and T. Ertl. Adaptive sampling in three dimensions for volume rendering on GPUs. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 113–120, 2007.

7. J. Kruger and R. Westermann. Acceleration techniques for GPU-based volume rendering. *IEEE Visualization, 2003. VIS 2003*, pages 287–292, 2003.

8. P. Ljung. Adaptive sampling in single pass, GPU-based raycasting of multiresolution volumes. In *Proceedings Eurographics/IEEE Workshop on Volume Graphics 2006*, pages 39–46, 2006.

9. S.R. Marschner and R.J. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of the conference on Visualization'94*, pages 100–107. IEEE Computer Society Press, 1994.

10. F. Rengier, S. Doll, H. von Tengg-Kobligk, J. Kirsch, H.U. Kauczor, and F.L. Giesel. Integrated teaching of anatomy and radiology using three-dimensional image post-processing. *European Radiology*, 19(12):2870–2877, 2009.

11. S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart hardware-accelerated volume rendering. In *Proceedings of the symposium on Data visualisation 2003*, page 238. Eurographics Association, 2003.

12. GD Rubin, CF Beaulieu, V. Argiro, H. Ringl, AM Norbash, JF Feller, MD Dake, RB Jeffrey, and S. Napel. Perspective volume rendering of CT and MR images: applications for endoscopic imaging. *Radiology*, 199(2):321, 1996.

13. H. Scharsach. Advanced GPU raycasting. *Proceedings of CESCG*, 5:67–76, 2005.

14. JP Schulze, M. Kraus, U. Lang, and T. Ertl. Integrating pre-integration into the shear-warp algorithm. In *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, page 118. ACM, 2003.