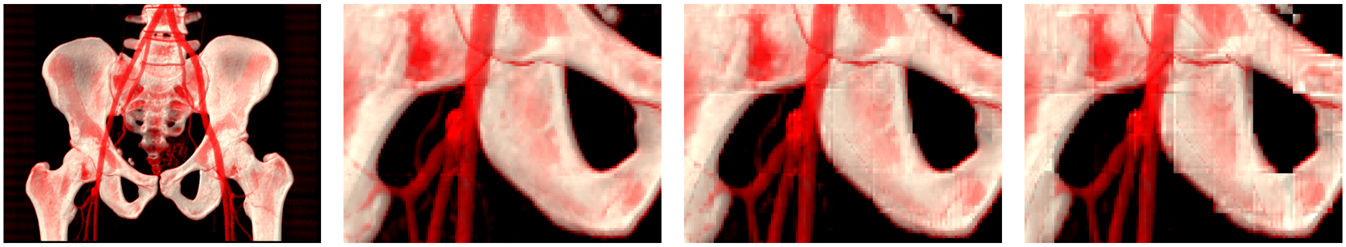
Transfer Function Based Adaptive Decompression for Volume Rendering of Large Medical Data Sets

Patric Ljung* Claes Lundström† Anders Ynnerman* Ken Museth*

*Norrköping Visualization and Interaction Studio, Linköping University
†Center for Medical Image science and Visualization, Linköping University, and Sectra-Imtec AB



1. Original volume losslessly compressed to 54 MB (ratio 2.7:1).

2. Rendered volume using 11% of full data size (16 MB, ratio 9:1).

3. Rendered volume using 1.4% of full data size (2.0 MB, ratio 72:1).

4. Rendered volume using 0.3% of full data size (0.44 MB, ratio 326:1).

Figure 1: Data reduction effects using Adaptive Decompression with our High Quality scheme and occlusion compensation. CT data set of abdomen (144 MB, $512 \times 512 \times 384$ @ 12 bits / voxel). Images 2 – 4 show a zoomed in part of respective rendered image at varying data reduction levels, i.e. a de facto compression of the original data set.

Abstract

The size of standard volumetric data sets in medical imaging is rapidly increasing causing severe performance limitations in direct volume rendering pipelines. The methods presented in this paper exploit the medical knowledge embedded in the transfer function to reduce the required bandwidth in the pipeline. Typically, medical transfer functions cause large subsets of the volume to give little or no contribution to the rendered image. Thus, parts of the volume can be represented at low resolution while retaining overall visual quality. This paper introduces the use of transfer functions at decompression time to guide a level-of-detail selection scheme. The method may be used in combination with traditional lossy or lossless compression schemes. We base our current implementation on a multi-resolution data representation using compressed wavelet transformed blocks. The presented results using the adaptive decompression demonstrates a significant reduction in the required amount of data while maintaining rendering quality. Even though the focus of this paper is medical imaging, the results are applicable to volume rendering in many other domains.

CR Categories: E.4 [Coding and Information Theory]—Data compaction and compression; I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms; I.4.2 [Image Processing and Computer Vision]: Compression (Coding)—Approximative coding;

Keywords: Volume compression, Volume rendering, Wavelet transform, Multiresolution, Transfer function, Adaptive decompression, Medical imaging, Image quality measures

1 Introduction

Volumetric data sets routinely captured in medical imaging are rapidly increasing in size due to improved geometric resolution and decreased examination times in state-of-the-art imaging modalities. Since increasingly precise information can be gathered, improved diagnostic procedures are possible and new types of examinations can be developed, replacing invasive methods with non-invasive methods to minimize patient risk and discomfort. However, as data sizes increase there is a need for improved tools that enable interactive exploration of the data sets. Potentially, the most important visualization method for medical diagnostic work on these GB-sized volumes is Direct Volume Rendering (DVR) but it is a difficult task to introduce DVR into the diagnostic work flow [Andriole 2003]. Technical limitations in terms of memory and bandwidth pose challenges for the visualization pipeline, making interactive frame rates hard to reach and maintain. To address these problems a method that can reduce the required bandwidth and memory usage for retrieval, unpacking and rendering of these data sets is urgently needed.

This paper contributes to the solution of the above problems by significantly reducing the amount of data to be processed by the DVR pipeline. Our approach is based on the central role that the Transfer Function (TF) plays in DVR. When a TF is applied, large subsets of the volume will give little or no contribution to the rendering, even if those regions have high energy in the original volume. A typical medical TF for CT volumes makes tissue with attenuation lower than fat completely transparent. This usually means that more than 50% of the voxels do not contribute in the rendering. The major idea is to make use of the knowledge encoded in the TF to select a level-of-detail (LOD) that reduces the data for retrieval, reconstruction and rendering. First, the volume is divided into blocks that are passed through a compression scheme, enabling several LODs for each block. This multi-resolution feature is then exploited in order to give significant blocks high resolution, and vice versa. The selection of LOD for each block is performed adap-

*{plg,andyn,kenmu}@itn.liu.se
†clalu@imv.liu.se

tively using a significance priority scheme during decompression.

We present results for two versions of TF-based LOD selection, a High Quality scheme based on an exhaustive analysis and a Fast scheme with a simplified approach. The former is intended to explore the potential of adaptive decompression whereas the second scheme aims for use in a real-time visualization pipeline. As can be seen in the results section, the Fast scheme is usually able to achieve results comparable to the High Quality scheme.

Furthermore we have introduced a distortion metric based on the distortion in the rendered images rather than the distortion in the reconstructed volumes. Primarily the distortion in the reconstructed volumes has a reduced correspondence to the final result after application of the TF. Secondly a view dependent LOD also makes a distortion measure on the volume data after application of a TF less feasible. Thus, we propose a quality measure based on the distortion of the rendered images using the perceptually adapted CIELUV color space.

The major advantage of our method is that it exploits the data reduction potential of feeding back the TF into the decompression stage. Accordingly, no *a priori* information of TFs or other rendering parameters is needed at compression time. Since this data reduction can benefit all stages of the visualization pipeline from retrieval through decompression to rendering we have defined it as a *de facto compression*. The proposed adaptive decompression scheme provides a significant performance improvement for both lossless and lossy compression. The scheme can also readily be applied on top of other DVR schemes working with multi-resolution representations, enhancing existing compression ratios. This paper focuses on DVR for medical imaging, nevertheless, the presented methods are applicable in many other application domains for DVR.

The paper is organized as follows: Section 2 reviews related work. The LOD selection process is described in detail in section 3. Section 4 presents some implementation issues of our visualization pipeline. Section 5 contains the test results. Concluding remarks are given in section 6, together with ideas for future work.

2 Related work

Direct volume rendering (DVR) techniques [Kaufman 1991] have been the focus of vast research efforts in recent years. Our work attempts to reduce the amount of data to process in the DVR pipeline. This goal is shared by several other researchers. Westermann [1994] has presented a multi-resolution framework for DVR where ray-casting rendering with adaptive sampling frequency is performed directly on the wavelet transformed data. Schneider and Westermann [2003] proposed a compression and rendering scheme for DVR based on vector quantization. An advantage of the approach is the ability to both decompress and render on the graphics hardware. As in this paper, Guthe et al. [2002] achieve a multiresolution representation through a blocked wavelet compression scheme. In the decompression stage an LOD selection occurs, prioritizing block resolution partly according to the reconstruction error of different LODs.

In the following paragraphs we will in some detail review work that is most related to this paper, i.e. the research targeting use of the transfer function or other visualization features in the decompression stage.

Bajaj et al. [2001] explore the use of voxel visualization importance in the compression process. Voxel weights are defined, e.g. for DVR on the basis of transfer functions. Each wavelet coefficient is then modulated by the maximum weight in the voxel set that contributed to the coefficient. This aims to give coefficients with most visual importance the largest magnitude. Bajaj et al. show that applying a threshold to weighted coefficients yields higher quality than using unweighted ones. A significant drawback with this scheme is that the important visualization features need to be known at compression time. In addition, a limitation for the resulting image quality is the use of the simple Haar wavelet. To introduce a more advanced wavelet would make the weighting less precise, since each coefficient will depend on many more voxels if the wavelet filter support increases.

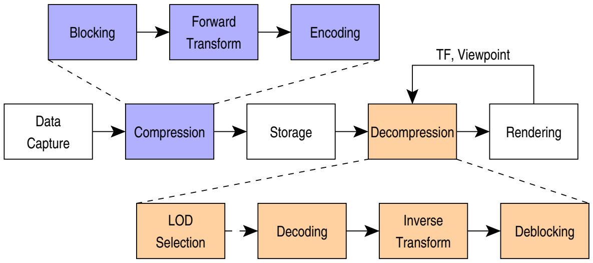


Figure 2: Schematic visualization pipeline. Input to the LOD selection process is the current TF and viewpoint from the rendering stage.

In line with the ideas of our work, Sohn et al. [2002] suggest the use of volumetric features to guide the compression, in their case applied to time-varying volumes. The features are defined in terms of iso-surface values or intensity ranges. The data is first passed through a block-based Haar wavelet compression stage. Blocks that have little or no contribution to the selected features are discarded. The wavelet coefficients can also be thresholded depending on their contribution to the features. Again, a major limitation is that the features must be selected before compression occurs. Use of the multi-resolution data for an LOD selection has not been exploited in his work.

The work by Li and Shen [2002] aims to achieve constant frame rates for volume rendering. The volume is divided into subvolumes of varying size, where coherent regions result in larger subvolumes. A multi-resolution pyramid for each subvolume is created by straight-forward averaging. Rendering time budgets are allocated to subvolumes according to an importance value which can be controlled, among other factors, by the maximum opacity in the subvolume. The budgets then guide an LOD selection. The transfer function feed-back constitutes only a minor part of Li and Shen's paper.

In our opinion, the potential of using transfer function feed-back to enhance the de facto compression has not been fully explored. A comparison with the schemes presented above shows that our method provides a number of advantages such as not requiring transfer function knowledge at the compression stage, and the ability to be an add-on to other lossless or lossy visualization schemes.

3 LOD selection

The aim of our work is to decrease the amount of data sent for decompression and rendering with minimal impact on visual quality. Our approach is an adaptive decompression method using a TF to guide an LOD selection process. Thus, the LOD selection is the core of our method and is described in detail in this section. An overall view of our visualization pipeline is shown in figure 2. The pipeline provides the ability to reconstruct the volume with an individual LOD for each block.

We use $\mathcal{T} : \mathbb{R} \rightarrow \mathbb{R}^4$ to denote a TF and $\mathcal{T}_\alpha$ refers to the alpha component. Consider a block, containing a set of values $\mathbf{V}_b$. The *TF content* for the block is the set obtained by applying the TF to each value, i.e. the set $\mathcal{T}(v), v \in \mathbf{V}_b$. The LOD selection for a block depends on TF content as follows:

1. No TF content, $\mathcal{T}_\alpha(v) = 0, \forall v \in \mathbf{V}_b$: The block can be discarded without introducing distortion.

2. Non-varying TF content, $\mathcal{T}(v) = \mathbf{C}, \forall v \in \mathbf{V}_b$, where $\mathbf{C}$ is a vector constant: The block can be reduced to a single average value without introducing distortion.

3. Varying TF content, $\exists u, v \in \mathbf{V}_b$, such that $\mathcal{T}(u) \neq \mathcal{T}(v)$: Low distortion is achieved by letting the LOD depend on the derivative of the TF in the range of $\mathbf{V}_b$, a high derivative implies high resolution.

Each time the selected LOD is less than full resolution for a block a de facto compression is obtained. In the cases of no or non-varying TF content the LOD reduction results in a lossless compression. Our adaptive decompression component can readily be added to traditional compression schemes. The only prerequisite is a multi-resolution data representation, for example wavelet transformed data. In this way our method can at decompression time enhance an existing compression scheme without counteracting it.

We use 16^3 blocks and thereby have five LODs corresponding to different resolutions, cubes with side 16, 8, 4, 2, and 1. We will refer to the resolution levels as L_{16}, L_8, etc. A sixth possibility is that the block is completely discarded.

Two LOD significance schemes have been developed, a High Quality version and a Fast version. The intent of the High Quality scheme is to explore the full potential of TF based LOD selection, with little use outside the research context. Its results also serve as a high-end benchmark for the Fast scheme, which is designed to be usable in a real-time visualization pipeline. The Fast scheme is intended to achieve results close to the High Quality scheme while reducing processing time.

The components of the two schemes are presented in the following subsections. First their respective significance measures are described. Then their LOD priority scheme components are explained. Finally, we describe how the measures can be adjusted for occlusion effects.

3.1 LOD significance measures

The High Quality (HQ) significance measures $s_{\mathrm{HQ}}(\lambda)$ are derived for each level λ of each block through an exhaustive analysis: Find the TF content for each LOD and then calculate the distortion compared to the TF content at full resolution. The distortion measure used is the ΔE, defined in appendix A. In this comparison each LOD is represented by a 16^3 block, achieved by wavelet expansion with zero coefficients as described in section 4.1. Equation 1 describes how $s_{\mathrm{HQ}}(\lambda)$ is derived. The voxel value for level λ at position p is denoted by $v_{p,\lambda}$.

$$s_{\mathrm{HQ}}(\lambda) = \sum_p \Delta E\Big(\mathcal{T}(v_{p,\lambda}), \mathcal{T}(v_{p,16})\Big) \qquad \lambda = 8, 4, 2, 1 \tag{1}$$

The exhaustive analysis needed for the s_{HQ} is very slow and must be performed on decompressed data and, thus, is of little use in a real pipeline. The Fast significance measure s_{F} employs a less time-consuming approach based on block specific meta-data acquired at compression time: the average $\bar{x}$, the root mean square wavelet coefficient $C_{\mathrm{RMS}}(\lambda)$ for each level λ, and a simplified histogram. Their respective use is described below. The memory overhead introduced by the meta-data is minor and it is also straightforward to give it a condensed representation.

The simplified histogram reduces the full histogram of a block to a number of piece-wise constant segments. The value range of the block is divided into small segments, in this paper an initial width of 10 is used. Each segment height is set as the average of the original histogram in the segment range. Pairs of segments are then iteratively merged until just a few remain. At each step the pair with least height difference is merged. This method preserves the shape of the original histogram well, since segments at steep slopes will not be merged. In this paper we have used a limit of 12 segments, as for the example in figure 3.

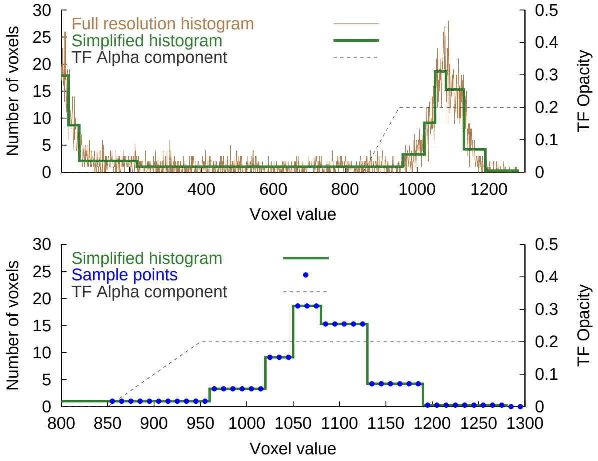


Figure 3: Top: A block histogram from a medical CT volume and its piece-wise constant approximation using 12 segments. Bottom: Retrieving TF content from the simplified histogram by sampling (blue dots) at small intervals in the TF range.

Having collected the meta-data during compression, $s_{\mathrm{F}}(\lambda)$ is derived for all levels λ at the decompression stage. The first step is to calculate $s_{\mathrm{F}}(1)$, the reduction of TF content obtained when going from L_{16} to L_1. The simplified histogram is sampled at small intervals in the TF range, as displayed in the bottom part of figure 3. For each sample the TF is applied. The ΔE distortion for the interval is retrieved by comparing with $\mathcal{T}(\bar{x})$, where the block average $\bar{x}$ is used to approximate the L_1 TF content. The distortion must then be multiplied by the number of voxels in the interval. Finally, all interval distortions are added to obtain $s_{\mathrm{F}}(1)$, as described in equation 2. A histogram interval has an index i, a midpoint x_i, a height h_i, and all intervals have the same width w. The interval width used in this paper is 10, the same as the minimum segment width.

$$s_{\mathrm{F}}(1) = \sum_i \Delta E\Big(\mathcal{T}(x_i), \mathcal{T}(\bar{x})\Big) \cdot h_i \cdot w \tag{2}$$

The overall TF content reduction of a block is described by $s_{\mathrm{F}}(1)$. The next step is to derive how it is distributed over the intermediate levels, such as $s_{\mathrm{F}}(8)$ corresponding to the L_{16} to L_8 transition, etc. A large difference in TF content roughly corresponds to large changes of the voxel values between the levels, i.e. in large wavelet coefficients. Thus, by comparing the root mean square coefficients $C_{\mathrm{RMS}}(\lambda)$ for all levels λ, an appropriate portion of the total distortion can be assigned to each level. The derivation is described in equations 3 and 4.

$$C_{acc}(\lambda) = \sum_{\lambda' > \lambda} C_{\mathrm{RMS}}(\lambda'), \qquad \lambda = 8, 4, 2, 1 \tag{3}$$

$$s_{\mathrm{F}}(\lambda) = s_{\mathrm{F}}(1) \cdot \frac{C_{acc}(\lambda)}{C_{acc}(1)}, \qquad \lambda = 8, 4, 2 \tag{4}$$

3.2 Priority schemes

The High Quality and Fast LOD selection also have separate versions of priority schemes, based on the significance measures s_{HQ}

and s_{F}, respectively. The common outline of the schemes is as follows. First all blocks with no significance are removed from further pipeline processing. The remaining blocks are initially registered as L_1. A priority queue is created, containing all valid LOD transitions for all blocks. The queue is then sorted according to transition efficiency (explained below). Finally, the queue is traversed from the top, updating block LODs until the desired compression ratio is reached.

The measure for the efficiency of a block LOD transition is the relative significance, Δs. It is derived in a similar way for both schemes, as shown in equation 5. Consider the two levels involved in a block LOD transition. The relative significance is the difference in significance divided by the difference in size. s is either s_{HQ} or s_{F}, and N_i is the size of level λ_i.

$$\Delta s(\lambda_a, \lambda_b) = \frac{s(\lambda_b) - s(\lambda_a)}{N_a - N_b}, \qquad \lambda_a > \lambda_b \tag{5}$$

The High Quality scheme is implemented as an iterative solution. A block will only have one valid LOD transition at a time, the one with the highest Δs connected to the current LOD. For instance, a block at L_2 will find the valid transition among $L_2 \rightarrow L_4$, $L_2 \rightarrow L_8$, and $L_2 \rightarrow L_{16}$. Thus the size of the priority queue is the same as the number of significant blocks. When the block at the top of the queue registers its transition, a new valid transition maximizing Δs is found and reinserted into the queue.

For performance reasons the Fast scheme is not iterative. The priority queue is populated with all possible transitions, i.e. several instances for each block. The Δs for each transition is calculated initially. Some transitions depend on others, e.g. $L_2 \rightarrow L_8$ cannot occur unless $L_1 \rightarrow L_2$ has occurred. Incorrectly ordered dependent transitions are handled by setting their Δs just below the value of their predecessor, putting them lower in the priority queue. Another situation to handle is when a transition invalidates a later one, which is solved by always ignoring transitions that do not start at the current LOD of the block. From tests we learned that the L_2 level was rarely used in the resulting LOD selections. Therefore, this level is removed from the Fast priority scheme to increase performance. This simplification reduces the possible LOD transitions from 10 to 6, which in turn reduces the size of the priority queue by 40%.

Note that a block can skip intermediate LOD transitions in both schemes. If only the next higher level would be considered, many blocks would erroneously remain at a lower level. For example, if L_1 and L_2 have roughly the same significance, $s(1) \approx s(2)$, the block would not be likely to ever get to L_2 even if $s(8)$ were very high.

To achieve a close to lossless rendering, either priority scheme is by-passed by setting all blocks with non-zero significance directly to L_{16}. A perfect rendering is not achieved, since small errors in some cases occur when a ray crosses a block boundary in the rendering. The test results in table 2 show, however, that the resulting distortion is not perceivable, which is why we refer to this setting as *virtually lossless*.

3.3 Accounting for occlusion

In the case of a rendering with high opacity, large parts of the volume will be completely or partially obscured. Even if an occluded block has, in itself, TF content, this will never reach the viewer. Therefore, using the TF to estimate occlusion effects enables LOD reduction possibilities in addition to the significance measures described in section 3.1. In fact, occlusion adjustment is an essential LOD selection component for many TFs.

Our model for occlusion simulates a low resolution ray-casting renderer. Input to the ray-caster is the current viewpoint and the simplified histograms for each block. A block is occluded if the incoming light intensity is low, therefore this value is noted during the simulated rendering. The occlusion is accounted for by weighting the significance measures, s_{HQ} or s_{F}, with the average incoming intensity for each block.

The occlusion footprint (the outgoing light intensity) for each block is obtained by simplifying the discrete rendering equation (eq. 6, no emission factor). I_{in} is the incoming light intensity into a block, I_{out} is the outgoing, and α_i is voxel opacity.

$$I_{out} = I_{in} \prod_i (1 - \alpha_i) \tag{6}$$

Since we have the simplified histogram from section 3.1, we can calculate an estimated average opacity, $\bar{\alpha}$, of each block. A naïve simplification would be to replace each voxel opacity by the average, i.e. $I_{out} = I_{in}(1 - \bar{\alpha})^{\bar{n}}$, where $\bar{n}$ is the average number of ray samples through the block. More precision is obtained if only the non-zero opacities are considered which introduces $\bar{\alpha}_{nz}$ and $\bar{n}_{nz}$.

However, two main error sources need to be handled. The average opacity will cause a overestimation of I_{out}. As a simple example, consider two voxels along the ray with opacities 0.8 and 0, resulting in a reduction of incoming light of 80%. The average approach approximates this by two voxels of opacity 0.4, making the reduction only 64%. The second error source is the average number of ray samples, underestimating I_{out}. Rays with fewer samples than average will contribute more to the outgoing intensity in relation to rays with more samples. Consider two rays with 1 and 3 samples, all with opacity 0.5. The intensity reduction will be 50% and 88% for the rays, an average of 69%. However, the reduction for the ray sample average of 2 is as high as 75%. These observations lead to the enhanced approximation model described in equation 7. Empirical tests have led us to use $c_\alpha = 1.3$, $c_n = 0.5$ for the abdomen data set, and $c_\alpha = 1.2$, $c_n = 0.1$ for the heart data set. A more advanced approach would automatically adapt these constants to the properties of each data set.

$$I_{out} = I_{in}(1 - c_\alpha \bar{\alpha}_{nz})^{c_n \bar{n}_{nz}} \tag{7}$$

4 Implementation

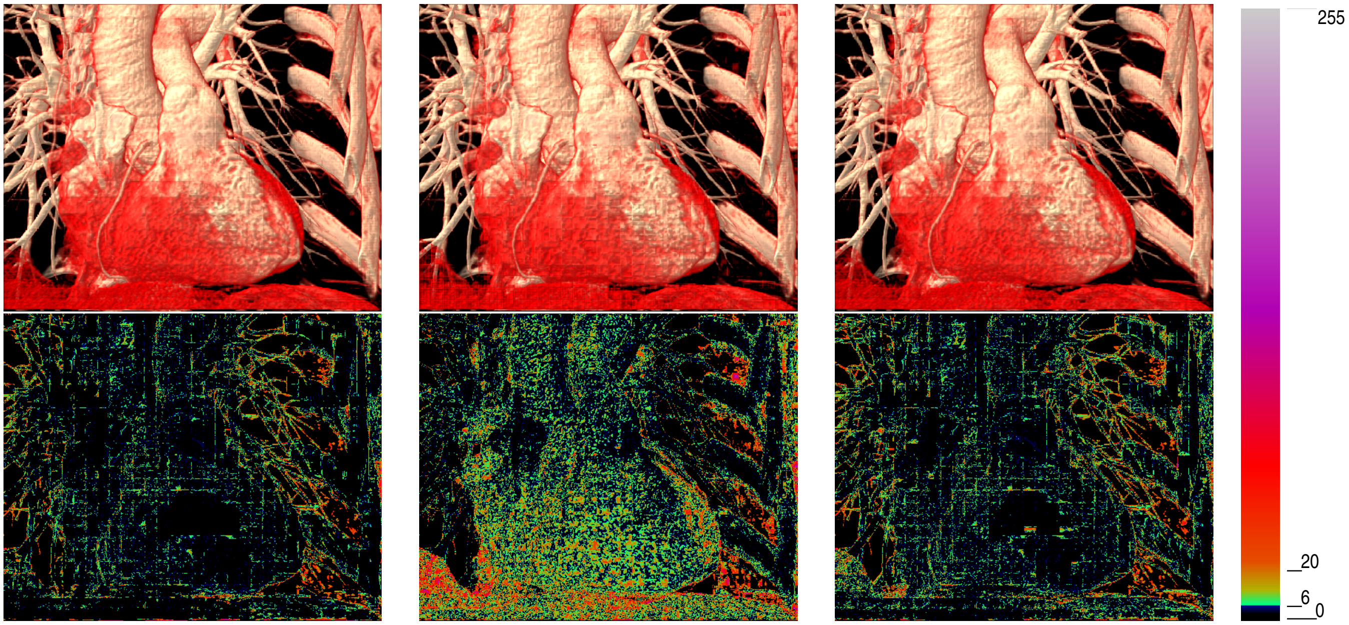
The methods described in section 3 have been implemented in a visualization pipeline, outlined in figure 2. This section describes the details of the pipeline, involving the block-based wavelet compression scheme and the ray casting volume renderer, as well as the distortion metric.

4.1 Pipeline details

For the results reported in this paper, the pipeline uses the 5/3 wavelet, a symmetric biorthogonal spline wavelet supporting lossless compression [Calderbank et al. 1996]. At block boundaries, symmetric extension has been employed [Brislawn 1995]. The 5/3 filter has a relatively small kernel while achieving good decorrelation. This wavelet has previously been used in blocked compression schemes for visualization [Guthe et al. 2002] and it is also part of the JPEG-2000 standard [Adams 2001]. The transforms have been implemented using the lifting scheme [Sweldens 1996].

Block LODs below L_{16} are constructed by feeding a stream of zero-valued coefficients to the inverse wavelet transform for the remaining levels up to full resolution. The 5/3 filter performs linear interpolation except on the boundary where the sample value is instead repeated. For the interior interpolated samples this scheme emulates typical trilinear interpolation performed in volume rendering.

We use a Huffman encoder [Huffman 1952] to achieve fairly high compression ratios with fast decompression, as demonstrated by Guthe et al. [2002]. To benefit from the many short zero sequences in the coefficients from our lossless wavelet transform, we



1. HQ (4.20, 12.9%)   2. WQ$_3$ (4.32, 34.0%)   3. HQ+WQ (3.65, 14.7%)   Color map

Figure 4: Comparisons between our High Quality (HQ) adaptive decompression scheme (left), traditional wavelet quantization (WQ) compression (middle) and the combination of both schemes (right) using the heart data set. The compression ratio for all images is 28:1. Bottom row contains difference images based on the rendered distortion compared to full quality, ΔE_{RMS} and ΔE_6 within parenthesis. A color map has been applied to the ΔE values. Black and blue is below ΔE_6.

introduce a few extra Huffman symbols for seven, four, and two zeros in sequence.

In order to achieve measurements with low quantization noise we have developed a software ray caster based on floating point arithmetic. The transfer function has full resolution in the sample range, 12 bits, and the rendered images are saved at 15 bits per component, i.e. 60 bits per RGBA pixel. The transfer function is pre-integrated using an approximation of the rendering integral [Engel et al. 2001] and stored in a look-up table.

4.2 Distortion Metric

The common distortion metric for volume compression schemes is to measure the accuracy of the reconstructed volume compared to the original, referred to as *reconstruction distortion*. The two main measures used are the peak signal to noise ratio (PSNR) [Nguyen and Saupe 2001] and signal to noise ratio (SNR) [Schneider and Westermann 2003], both based on root mean squared error (RMSE). These measures do not take the TF into account, thereby being of limited use to evaluate quality in the rendered image, which is the essential measure in our work. The TF-based LOD selection may cause a large reconstruction distortion while retaining visual fidelity. Thus, we propose a distortion measure applied to the rendered image, *rendered distortion*. The pixel-wise difference, ΔE, is defined in the CIELUV color space (see appendix A). Comparing two images, ΔE is computed for each pixel. As a measure of average rendered distortion we define ΔE_{RMS}, the root mean square of the pixel ΔE values.

A limitation of ΔE_{RMS} is the inability to pick up individual pixels with high distortion. To enhance the fidelity of the measurement, a second measure ΔE_6 is proposed, defined as the ratio of pixels with $\Delta E > 6.0$. This threshold is feasible since pixel distortion below it is unlikely to cause misinterpretations of the image data, see appendix A.

5 Results

Tests have been performed on three medical CT volumes, a chest, an abdomen, and a heart, of dimensions 512^3, $512 \times 512 \times 384$, and $512 \times 448 \times 416$ voxels, respectively. The compression ratio measurements are calculated based on the actual data range of 12 bits per voxel. The presented ratios refer to the total de facto compression, i.e. combining the entropy encoding effect at compression time with the LOD selection effect.

The data sets and TFs are from actual clinical use at CMIV[1], the TFs are presented in appendix B. The horizontal lines in the heart are artifacts due to the fact that the capturing process extends over several heart cycles, one volume slab being captured at a certain phase of each cycle. Since the chest volume is rendered with a low opacity TF, no occlusion correction has been used. Occlusion correction is not used in virtually lossless settings for any data set.

5.1 Potential of adaptive decompression

In order to roughly evaluate the potential of TF guided adaptive decompression, we have compared our HQ scheme with a traditional compression scheme based on quantization of wavelet coefficients (WQ). The quantization is level dependent, with larger step size for higher resolution coefficients, and the zero sequence enhanced Huffman encoding is used for the traditional scheme as well.

Table 1 presents the results for the heart data set, being a difficult case for an LOD selection approach with many narrow features. At the lowest compression ratio, the HQ scheme results in much lower rendered distortion than WQ. For higher ratios the performance in terms of ΔE_{RMS} is more equal, but HQ consistently has a significantly lower ΔE_6, i.e. fewer high distortion pixels. The results also demonstrate that high reconstruction distortion (PSNR) does

[1] Center for Medical Image science and Visualization, Linköping University

Ratio	Method	Rendered dist		Rec'd dist
		ΔE_{RMS}	ΔE_6	PSNR (dB)
5.73:1	HQ	0.06	0.02%	25.1
	WQ$_1$: 1, 1, 2, 4	0.63	2.15%	63.9
15.6:1	HQ	2.72	4.88%	24.7
	WQ$_2$: 2, 4, 8, 16	2.36	17.1%	52.1
28.2:1	HQ	4.20	12.9%	24.6
	WQ$_3$: 4, 8, 16, 32	4.32	34.0%	46.4
	WQ$_1$ + HQ	3.65	14.7%	24.8

Table 1: High Quality (HQ) adaptive decompression versus traditional wavelet quantization compression (WQ) for the heart data set. HQ yields similar or lower rendered distortion at equal compression ratios. WQ and HQ can be combined with good results. The traditional schemes are denoted along with their quantization step size for each level (L_2, L_4, L_8, L_{16}).

not necessarily lead to high rendered distortion (ΔE_{RMS}, ΔE_6). Finally, we have also combined the HQ scheme with the lossy compression WQ, demonstrating that the two schemes work well together. The corresponding images are found in figure 4. The difference images demonstrate the ability of HQ to prioritize image quality where TF content is high, which the quantization scheme cannot achieve.

We also compared HQ to other LOD selection schemes. A low quality benchmark is *Uni* consisting of a uniform LOD scheme where the blocks are all reduced to L_8 yielding 8:1 compression. A second reference is *RecE*, an LOD selection based on the distortion in the reconstruction according to the $\mathbb{L}_2$ norm (an RMSE), an approach partly used by Guthe et al. [2002].

The results are given in table 2. The distortion for the chest data set is generally lower, since the rendered image is more transparent. At equal ΔE_{RMS}, the HQ scheme gives 4.3–6.9 times better compression ratio than Uni and 2.4–3.8 times higher than the RecE scheme. ΔE_6 is slightly higher for HQ than the other schemes at moderate ΔE_{RMS} values. These high distortion pixels are mainly caused by block border artifacts. For the chest data set, HQ yields a virtually lossless de facto compression ratio of 7.71:1, for the abdomen data set 8.89:1. The lossless compression ratios our Huffman variant achieves for the wavelet transformed blocks (including meta-data) are 2.15:1 and 2.64:1, respectively. Thus, the adaptive decompression enhances the existing lossless compression ratio by a factor of 3.4–3.6. Some corresponding images are presented in figures 1 and 5. As for the heart data set, HQ yields low distortion for the regions with high TF content, whereas the distortion of the Uni and RecE schemes is more evenly spread.

Using the TF as input for an occlusion factor estimation proves efficient for high-opacity TFs as shown in table 3. The ΔE_{RMS} distortion is reduced by up to 40%.

5.2 Fast scheme

The goal of Fast adaptive decompression scheme is to get as close as possible to the qualitative results of the High Quality scheme (HQ) with minimum computational effort. Results are given in table 2. At similar ΔE_{RMS} distortion, the Fast scheme yields 0.3–1.0 times the HQ compression ratio and 1.3–2.7 times the RecE ratio. The performance of the Fast scheme relative to HQ increases with decreasing compression ratio, ending up at equal rendered quality in virtually lossless mode. Some corresponding images are presented in figure 5.

An example of LOD distribution for our proposed schemes are shown in table 4. Both schemes use all valid levels (L_2 is invalid for the Fast scheme), but HQ leaves more blocks at level L_1, prioritizing transitions from intermediate to high levels.

Data set	Method	ΔE_{RMS}	Ratio	ΔE_6
Chest	HQ	0.06	7.71	0.00%
	Fast	0.06	7.71	0.00%
	HQ	0.55	15.1	0.09%
	Fast	0.57	13.4	0.13%
	RecE	0.55	5.15	0.05%
	HQ	1.02	31.8	0.31%
	Fast	1.02	18.7	0.56%
	RecE	1.02	13.1	0.22%
	HQ	1.35	66.8	0.66%
	Fast	1.34	25.4	0.96%
	RecE	1.29	18.5	0.48%
	Uni	1.31	15.5	0.54%
Abdomen	HQ	0.16	8.89	0.21%
	Fast	0.16	8.89	0.21%
	HQ	0.52	17.5	1.29%
	Fast	0.53	14.1	2.14%
	RecE	0.49	5.29	0.92%
	HQ	2.03	40.3	8.69%
	Fast	2.04	23.7	11.1%
	RecE	2.01	10.6	11.0%
	HQ	4.16	128.4	20.5%
	Fast	4.16	43.7	27.2%
	RecE	4.20	34.4	34.8%
	Uni	4.14	18.5	33.9%

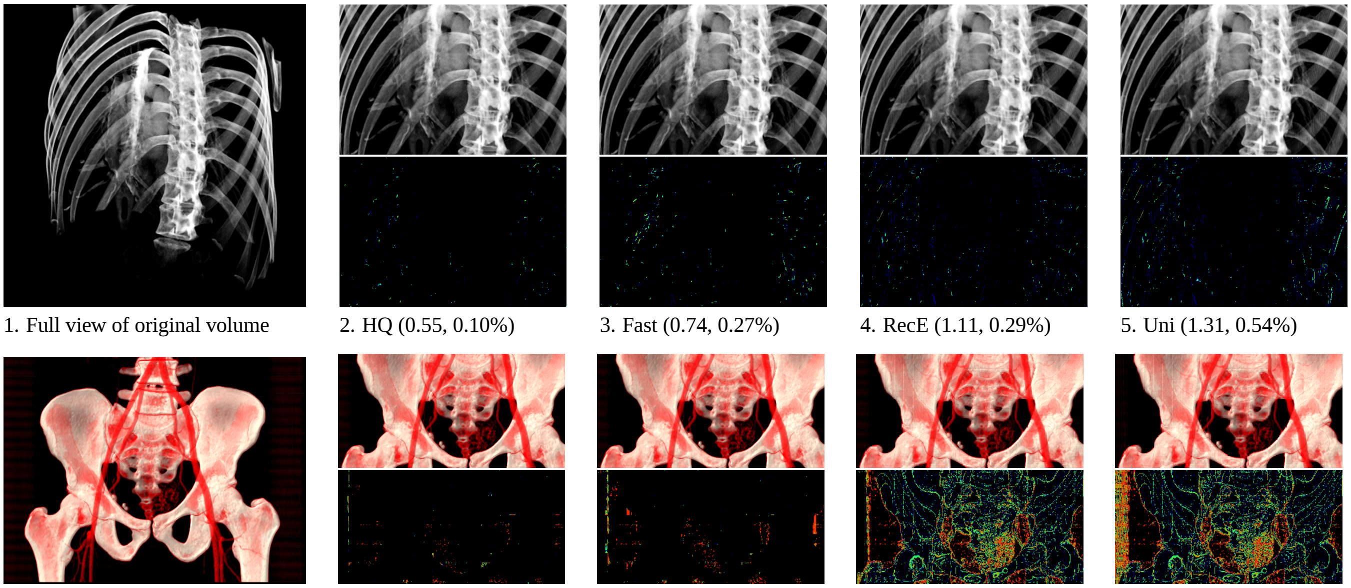
Table 2: High Quality (HQ) and Fast adaptive decompression compared to low quality benchmark (Uni) and reconstructed error based significance (RecE). The top two rows for each data set are virtually lossless settings.

Data set	Ratio	Method	ΔE_{RMS}	ΔE_6
Abdomen	17.5	HQ, occl	0.52	1.29%
		HQ, no occl	0.84	3.33%
	40.3	HQ, occl	2.03	8.69%
		HQ, no occl	2.85	13.8%

Table 3: High Quality (HQ) adaptive decompression with and without occlusion correction.

Test	Method	L_{16}	L_8	L_4	L_2	L_1	Insign.
Chest, 10:1	HQ	6451	818	29	2	1131	24337
	Fast	6349	1580	476	0	26	24337
Chest, 35:1	HQ	945	5464	130	12	1880	24337
	Fast	1115	3912	2210	0	1194	24337

Table 4: LOD distribution of High Quality (HQ) and Fast adaptive decompression at different compression ratios. The *Insign* column contains discarded blocks



1. Full view of original volume

2. HQ (0.55, 0.10%)

3. Fast (0.74, 0.27%)

4. RecE (1.11, 0.29%)

5. Uni (1.31, 0.54%)

6. Full view of original volume

7. HQ (0.52, 1.3%)

8. Fast (1.06, 5.2%)

9. RecE (2.71, 18%)

10. Uni (4.14, 34%)

Figure 5: Our High Quality (HQ) and Fast adaptive decompression schemes, compared to reconstructed error based LOD (RecE) and uniform LOD (Uni). Both rows: Images 1 and 6 show full quality renderings, images 2 – 5 and 7 – 10 show a part for renderings based on each scheme, where the ΔE_{RMS} and ΔE_6 values are given. Top row: Chest data set, images 2 – 5 are rendered using approximately 7% of full data size (ratio 15:1). Bottom row: Abdomen data set, images 7 – 10 are rendered using approximately 6% of full data size (ratio 18:1). The same color map is used for the ΔE images as in figure 4. Black and blue is below ΔE_6.

The important performance measure of this paper is the throughput of the Fast scheme. It has the potential to fit in a real-time visualization pipeline. Running a 2.4 GHz PC, the Fast LOD selection (i.e. calculating significance and running the priority scheme) excluding Huffman decoding and inverse transform for the 192 MB chest data set took 0.37 s for a 15.2:1 compression. This is the delay needed when the TF is changed. When only the viewpoint is changed, only the occlusion factors need to be recalculated, which is 0.06 s of the time above.

6 Conclusions

In this paper we have explored adaptive decompression based on putting the Transfer Function (TF) at the core of the visualization pipeline. Our High Quality scheme shows the great potential of using TF information for LOD selection, achieving high de facto compression while retaining visual quality. In virtually lossless mode the High Quality scheme achieves a de facto compression ratio of about 8:1. For the whole span of lossy settings tested, the method by far outperforms a LOD selection scheme based on reconstructed distortion.

We have also presented a Fast LOD selection scheme that is appropriate for use in a real-time visualization pipeline. In a virtually lossless setting it performs as well as the High Quality scheme. For increasing compression ratios, the Fast scheme performance relative to the High Quality scheme decreases. Even though this gap probably can be diminished through further refinement of the scheme, these results clearly demonstrate that a fast yet powerful scheme is feasible in practice.

A major reason for the qualitative limitations of the Fast scheme is the extensive simplification of small values in the block histograms. If the main content of the histogram is outside the TF range, the LOD selection is very sensitive to simplification errors for the remaining minor parts. A more detailed histogram approximation would reduce this effect but it would also lower the LOD selection performance.

We have shown that measuring reconstruction distortion, using for example PSNR, does not necessarily correspond to visual fidelity. The combination of our proposed measures for rendered distortion, ΔE_{RMS} and ΔE_6, has been shown to register significant artifacts and give a reliable measure on image quality. Medical imaging is especially sensitive to structured artifacts, which may potentially affect the diagnosis. In our future work we intend to explore the use of distortion metrics to extract structural information allowing the identification of such artifacts.

A well-known challenge for multi-resolution schemes is the block border artifacts due to inter-block interpolation difficulties. In our future work we will investigate the integration of methods from LaMar et al. [1999] and Weiler et al. [2000] into our schemes. These methods could operate directly on the multi-resolution blocks, using the LOD information to achieve an interpolation across block borders to reduce the rendered distortion.

Acknowledgements

This work has been funded by the the Swedish Research Council, grant 621-2001-2778 and 621-2003-6582, and the Swedish Foundation for Strategic Research, grant A3 02:116.

A CIELUV color space and metrics

The CIE 1976 $L^*u^*v^*$ (CIELUV) is a standardized color space for luminous colors, i.e. color monitors and television. It approximately incorporates perceptual aspects of the human eye. Although advanced color appearance models exist [Fairchild 1998], the CIELUV color space is adequate for difference measures of the work presented in this paper.

Conversion of R,G,B components to the $L^*u^*v^*$ components is performed in two steps. First, the RGB colors are transformed into CIE XYZ tristimulus coordinates using equation 8. Unfortunately, the RGB color components used in computer graphics do not refer to any particular standardized color space. Therefore, we have approximated the RGB colors to be the standardized sRGB colors (RGB_{709}) [Gen 1990; Poynton 1997].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} \quad (8)$$

In the second step, the tristimulus XYZ are converted to $L^*u^*v^*$ using equations 9 through 13 [Fairchild 1998]. The white-point, (X_n, Y_n, Z_n), in the CIE XYZ color space is computed from $\mathrm{RGB}_{709} = (1, 1, 1)$. Using X_n, Y_n, and Z_n in equations 10 and 11 give u'_n and v'_n, respectively.

$$L^* = 116(Y/Y_n)^{1/3} - 16 \quad (9)$$

$$u' = \frac{4X}{X + 15Y + 3Z} \quad (10)$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \quad (11)$$

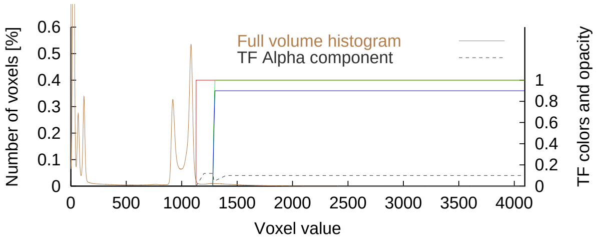
$$u^* = 13L^*(u' - u'_n) \quad (12)$$

$$v^* = 13L^*(v' - v'_n) \quad (13)$$

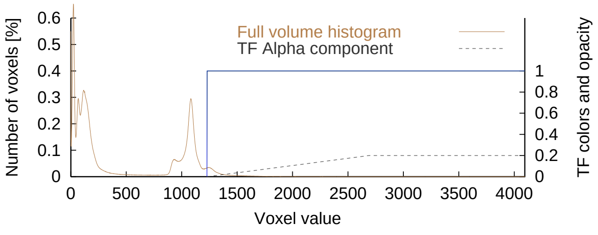
The CIE76 ΔE color-difference is defined in the CIELAB color space. We have adopted CIE76 ΔE to the CIELUV color space, as defined in equation 14. Individual pixel errors with ΔE around and below 1 are likely to be invisible to the human eye. In common practice, errors below a few units in ΔE are considered good quality and commonly not noticed by observers. It should be noted that ΔE in this paper refer to ΔE^*_{uv}.

$$\Delta E^*_{uv} = \left(\Delta L^{*2} + \Delta u^{*2} + \Delta v^{*2}\right)^{1/2} \quad (14)$$

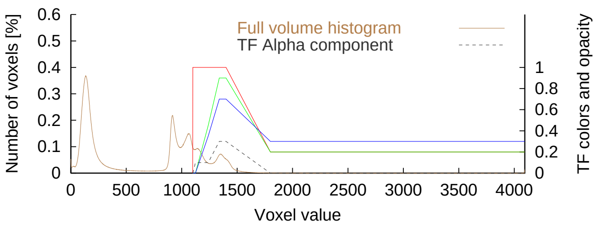
B Transfer functions and histograms



The abdomen data set



The chest data set



The heart data set

References

ADAMS, M. D. 2001. *The JPEG-2000 Still Image Compression Standard*. ISO/IEC (ITU-T SG8), September. JTC 1/SC 29/WG 1: N 2412.

ANDRIOLE, K. P., 2003. A position paper from the scar trip(tm) subcommittee. http://www.scarnet.org/pdf/TRIPwhitepaper1103.pdf, November. Acquired March 2004.

BAJAJ, C., IHM, I., AND PARK, S. 2001. Visualization-specific compression of large volume data. In *Proceedings Ninth Pacific Conference on Computer Graphics and Applications 2001*, 212–222.

BRISLAWN, C. M. 1995. Preservation of subband symmetry in multirate signal coding. *IEEE Transactions on Signal Processing 43*, 12 (December), 3046–3050.

CALDERBANK, A. R., DAUBECHIES, I., SWELDENS, W., AND YEO, B.-L. 1996. Wavelet transforms that map integers to integers. Tech. rep., Department of Mathematics, Princeton University, August.

ENGEL, K., KRAUS, M., AND ERTL, T. 2001. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Eurographics/SIGGRAPH Workshop on Graphics Hardware*, 9–16.

FAIRCHILD, M. D. 1998. *Color Appearance Models*. Addison Wesley Longman, Inc.

GENEVA: ITU. 1990. *ITU-R Recommendation BT.709: Basic Parameter Values for the HDTV Standard for the Studio and for International Programme Exchange (1990)*. Formerly CCIR Rec. 709.

GUTHE, S., WAND, M., GONSER, J., AND STRASSER, W. 2002. Interactive rendering of large volume data sets. In *Proceedings IEEE Visualization 2002*, 53–60.

HUFFMAN, D. A. 1952. A method for the construction of minimum-redundancy codes. In *Proceedings IRE*, vol. 40, 1098–1101.

KAUFMAN, A. 1991. *Volume Visualization (Tutorial)*. IEEE Computer Society Press.

LAMAR, E. C., HAMANN, B., AND JOY, K. I. 1999. Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings IEEE Visualization 1999*, 355–362.

LI, X., AND SHEN, H.-W. 2002. Time-critical multiresolution volume rendering using 3d texture mapping hardware. In *Proceedings IEEE Visualization 2002*.

NGUYEN, K. G., AND SAUPE, D. 2001. Rapid high quality compression of volume data for visualization. *Computer Graphics Forum 20*, 3.

POYNTON, C., 1997. Frequently asked questions about color. http://www.poynton.com/PDFs/ColorFAQ.pdf, March. Acquired January 2004.

SCHNEIDER, J., AND WESTERMANN, R. 2003. Compression domain volume rendering. In *Proceedings IEEE Visualization 2003*.

SOHN, B.-S., BAJAJ, C., AND SIDDAVANAHALLI, V. 2002. Feature based volumetric video compression for interactive playback. In *Proceedings IEEE Visualization 2002*.

SWELDENS, W. 1996. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Applied and Computational Harmonic Analysis*, 3, 186–200.

WEILER, M., WESTERMANN, R., HANSEN, C., ZIMMERMAN, K., AND ERTL, T. 2000. Level–of–detail volume rendering via 3d textures. In *Proceedings IEEE Volume Visualization and Graphics Symposium 2000*, 7–13.

WESTERMANN, R. 1994. A multiresolution framework for volume rendering. In *1994 Symposium on Volume Visualization*.