EUROGRAPHICS '97 / D. Fellner and L. Szirmay-Kalos
(Guest Editors)

Volume 16, (1999), Number 3

A Multiscale Approach to Integrated Volume Segmentation and Rendering

Rüdiger Westermann and Thomas Ertl

{wester,ertl}@informatik.uni-erlangen.de
Computer Graphics Group, University of Erlangen

Abstract

A number of techniques have been proposed for rendering volumetric scalar data sets. Techniques have also been proposed for analyzing the three dimensional information contents of the underlying domain, but traditionally the data analysis part is left as a post-processing step which only involves the rendered two dimensional images. In this paper, we describe a visualization method for scalar volume data which integrates explicit knowledge of the underlying domain into the rendering process. The key of this approach lies in a hierarchical description of the discrete signal, which is decomposed into a sequence of multiscale representations. We describe a technique for the analysis of structures within the data. This allows for the segmentation and classification of the relevant features and can be used to improve their visual sensation. We also address the problem of accelerating the final rendering pass by integrating the extracted object space information into the ray traversal process.

1. Introduction

During the last decade several approaches to direct volume rendering have been developed and explored extensively. Traditionally, these methods do not take into account the structure of the underlying data. Explicit knowledge of the objects characteristics is not integrated, and consequently, the data analysis part is left as a post-processing step which is mainly performed in image space.

Obviously, the results will be more meaningful if additional work is spent on the direct analysis of the three dimensional structures. This also helps to optimize the complete analysis/synthesis cycle. The key lies in the integration of the extracted information into the final rendering pass. Thereby, the gap between data analysis and image synthesis can be closed. Different stages of the whole visualization process (see Figure 1) may be considered simultaneously to generate the desired effects.

In this paper, we propose a new technique that performs a structural analysis of the data and incorporates the outgoing information into the imaging process. Basically, there are the following advantages to this strategy.

- ***Feature Extraction:*** The dominant features within the volume data are extracted and can be perfectly discriminated from white noise distributions.

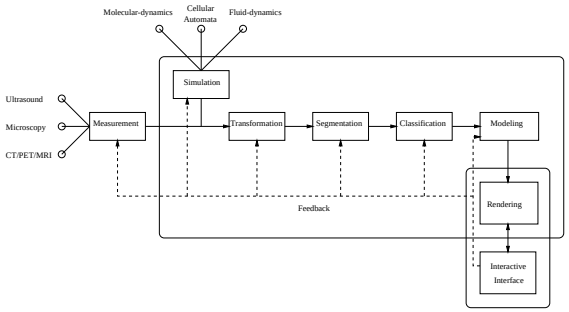


Figure 1: Volume visualization pipeline.

- ***Data Enhancement:*** Since the relevant structures are determined, their visual appearance can be enhanced. This improves the information contents of the rendered images and optimizes the data analysis cycle.
- ***Performance Tuning:*** The rendering times can be accelerated to some extent by integrating the obtained information into the visualization process.

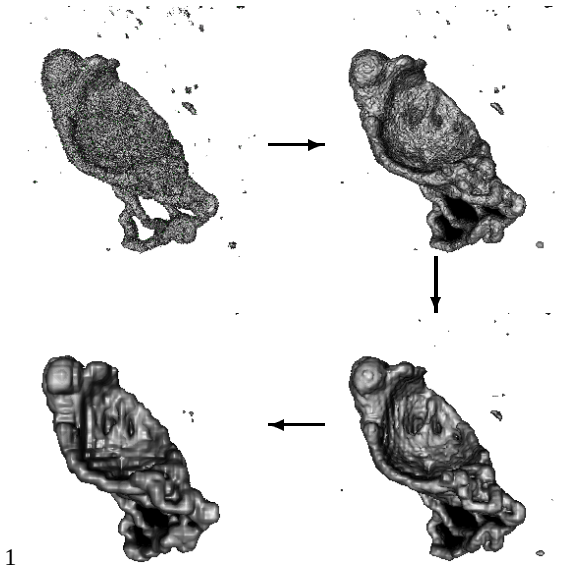
The forerunners to our technique can be classified in terms of the used data structures and the way in which they try to analyze the characteristics of the object. As in traditional im-

age processing the data analysis is commonly performed in a preprocessing step. It is used to generate a modified version of the original signal in which the desired effects are rendered prominent.

While a number of 2D techniques can be straight forwardly extended to 3D, specific approaches for analyzing volumetric data sets have been proposed.

In [6] it was claimed to estimate the consistency of volume cells from material dependent probability distributions. A very popular and efficient shading technique was introduced in [17]. This method combines the local grey-scale gradient and a material threshold to determine sharp boundary regions of interest. Then, the material opacity within these regions is scaled up accordingly in order to enhance the extracted structures.

On the other hand, several acceleration techniques have been explored [16, 31, 4, 7, 8, 23] which try to integrate the composition of the volume data into the rendering process. Frequently utilized data structures are octrees, which store multiple copies of the signal with decreasing resolution (see Figure 2).



1

Figure 2: Octree reconstruction from different resolution levels.

Recently, wavelet based multiresolution techniques for volume visualization have been examined in more detail [25, 26, 29, 10]. Wavelet transforms describe an effective tool to compress multi-dimensional discrete signals, and also to analyze their local characteristics. Furthermore, the original domain can be locally reconstructed from its multiresolution decomposition. By taking advantage of the sparseness of the representation the rendering process can be accelerated. The major limitation of wavelet based rendering techniques is the additional time they spent on reconstructing the data during the rendering pass. Only for very sparse representations the resultant times come close to those that can be achieved by traditional acceleration techniques.

On the other hand, wavelet based techniques allow one to precisely analyze the data at different scales. Thus, we recommend to take advantage of the representation by trying to extract the relevant structures.

As wavelet transforms preserve local difference information between successive averaging levels (see Figure 3), sharp variation points are still retained at distinct scales. Additionally, the evolution of the detail information across scales characterizes the original signal [11, 12, 22, 3]. In particular, it is possible to estimate the local smoothness from the decay of the wavelet coefficients maxima along different scales.

This multiscale behavior was exploited for the processing and visualization of time-resolved volume data [30], and also for the determination of critical regions within volumetric data sets [9]. A similar approach was presented by Sakas [28]. He argued intuitively that features within the volume must be retained on the coarse approximations. Consequently, the extraction of structures which remain dominant across different scales should be the main concern.

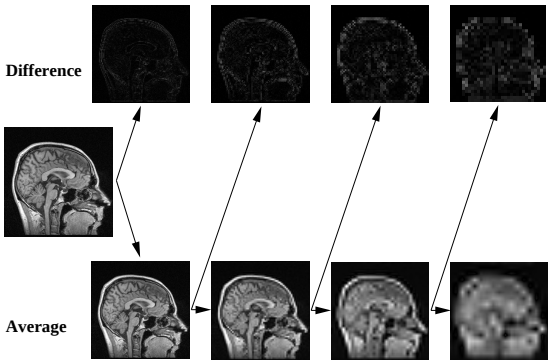


Figure 3: Schematic representation of the multiresolution hierarchy in two dimensions.

In the remaining sections we discuss the application of wavelet based multiscale techniques for the extraction and analysis of the dominant features from 3D scalar data sets. Once we have exemplified the extraction process, we describe how to efficiently integrate the found structures in order to accelerate the rendering process. Before we start with a detailed explanation of the proposed method, we first give an example which describes the key ideas.

2. The Experiment

A very popular method for visualizing 3D data sets in scientific applications is to determine material boundary regions and to render them in a prominent manner. This procedure is quite similar to the traditional edge detection techniques in image processing. Since edges carry the most important

information, it often suffices to visualize these structures in order to completely extract the information content.

In contrast to 2D techniques where the edge finding process only relies on the local gradient of the signal, the common approach in volume visualization is to determine surfaces of equal material properties, so called iso-surfaces. Either this is performed in object space by constructing geometric models which approximate the surface [18], or the extraction process takes place during the ray traversal in standard ray-casting techniques [17]. The left image in Figure 4 shows the visualization of an iso-surface according to a given threshold. The extracted surface belongs to the boundary region between the head and the outer air in a MRI-Scan.

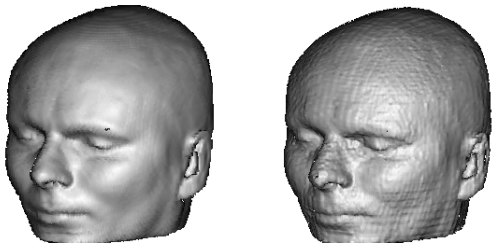


Figure 4: Iso-surface extraction from a human head MRI-Scan.

Now let us assume that we are only interested in extracting and visualizing this specific iso-surface. In order to demonstrate our method we decompose the original data set into a multiscale hierarchy, that is, starting with the original data set an averaging and a differencing step is performed at each level of the hierarchy. This process yields a pair of low-pass and high-pass filtered versions of the original signal at each stage. The same procedure is repeatedly applied to the low frequency part at each level.

Let us have a closer look on the hierarchical representation that is constructed. The difference information at each level describes the information that is lost due to the averaging step. Since the difference information carries the high frequency parts, high magnitude values are concentrated near sharp boundary regions. In empty or homogeneous regions the values are rather low.

But how much of the difference information do we really need to reconstruct the iso-surface? Obviously, large regions within the data can be removed since they do not contribute to the relevant features. This is outlined in the right image of Figure 4. The iso-surface was reconstructed from a multiscale hierarchy that was reduced about 95% of the original data.

Note that only a very small fraction of the original amount of memory is used, but we are still able to determine the principle location of the iso-surface. Of course, we loose information due to the reduction process, but even from the sparse representation those regions can be extracted which enclose the relevant features.

The major challenge to our approach is the development of a unified framework for finding the boundary regions in volume data sets and for integrating these structures efficiently into the rendering pass. Assuming that we have already found these regions and that we have marked them for further use, then the accuracy of the final rendering pass can be adjusted according to the structures we are seeking for.

Figure 5 demonstrates the key idea. From the structural analysis of the data we obtain skeletons which indicate the boundary regions. They are shown on a 2D slice from the MRI-scan. In order to visualize a certain boundary surface we just have to determine the intersection points with the corresponding structures. A nice feature then is that the rendering process can be accelerated significantly if the extracted information is integrated into the rendering pass.

Above all, note that this method does not rely on a specific material threshold. Only real boundary surfaces are rendered.

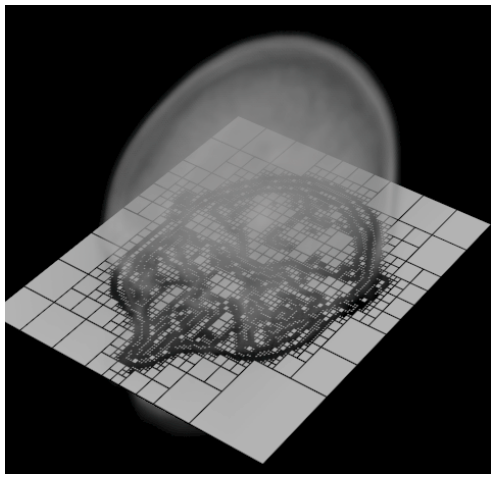


Figure 5: Octree based adaptive refinement.

In the following sections we discuss in more detail the mathematical theory and the major issues of our approach.

3. Multiscale Feature Extraction

A popular technique in signal processing and image analysis is the detection of irregular structures or edges which often carry most of the relevant information. Typical multiscale techniques try to decompose the original signal into a hierarchy which includes copies of the data at increasing scale. The size of details that can be detected within these copies scales up accordingly.

The major advantage of multiscale techniques is the ability to investigate and visualize the original signal at different

levels of detail. Moreover, from the evolution of the signal across scales its local characteristics can be determined.

One key idea is that if the original signal $f(x)$ is convolved with a smoothing function $\theta(x)$, then the zero crossings of $f * \theta''(x)$ can be used to detect the local sharp variation points of the blurred version of f [19]. Small fluctuations in the original signal are removed in the blurred one, while the dominant edges are retained during the convolution. Performing different passes with scaled versions of the same smoothing function yields the edge information at increasing scale [2]. The multiscale information can be gathered across different levels to determine the relevant features at the original scale.

In general, it is not necessary that the smoothing kernel is a Gaussian. The edge detection process can be improved if the kernel function has certain other properties [21, 22]. Assuming θ to be the integral over a wavelet Ψ[†]

$$\theta(x) = \int_{-\infty}^{x} \Psi(t)\, dt, \tag{1}$$

and setting $\theta^j(x) = 2^{j/2}\theta(2^j x)$ and $\Psi^j(x) = 2^{j/2}\Psi(2^j x)$, then the wavelet transform at scale 2^j can be written as

$$W^j f(x) = f * \Psi^j(x) = f * (2^j \frac{d\theta^j}{dx}) = 2^j \frac{d}{dx}(f * \theta^j)(x).$$

The wavelet transform at scale s, which is restricted to the dyadic scale $2^j, j \in \mathcal{Z}$, in the actual implementation, is thus proportional to the derivative of the original signal smoothed by $\theta^j(x)$.

Since the type of inflection point can be distinguished from the absolute value of $W^j f(x)$, the second derivative does not account for the determination of the signals edges. As a matter of fact, the multiscale edges of f can be fully specified by repeatedly convolving Ψ^j with the smoothed version $f * \theta^j(x)$ and determining the local maxima of $W^j f(x)$.

In order to extend the approach to three dimensions derivative wavelets $\Psi_1(\vec{x}), \Psi_2(\vec{x})$ and $\Psi_3(\vec{x})$ have to be constructed, which themselves are the partial derivatives along x, y and z of a three dimensional smoothing function $\theta(\vec{x})$. This process is defined with respect to images in [21][22]. Apparently we followed the same ideas for volumetric applications in this work. As in the one dimensional case the gradient vector at scale 2^j can then be written as

$$\vec{\nabla}(f * \theta^j)(\vec{x}) = \begin{pmatrix} W_1^j f(\vec{x}) \\ W_2^j f(\vec{x}) \\ W_3^j f(\vec{x}) \end{pmatrix} = 2^j \begin{pmatrix} \frac{\partial}{\partial x}(f * \theta^j)(\vec{x}) \\ \frac{\partial}{\partial y}(f * \theta^j)(\vec{x}) \\ \frac{\partial}{\partial z}(f * \theta^j)(\vec{x}) \end{pmatrix}.$$

Finally, at every position it is checked whether the gradient magnitude is a local maximum or not. This is done by determining the largest gradient magnitude along the direction the gradient is pointing to. The magnitude can be computed from the squares of the gradient components as

$$|\vec{\nabla} f| = \sqrt{(f * \Psi_1(\vec{x}))^2 + (f * \Psi_2(\vec{x}))^2 + (f * \Psi_3(\vec{x}))^2}.$$

A local maximum is found if the magnitude is strictly larger than its neighbor along the gradient direction and larger or equal than the neighbor in the opposite direction, and vice versa.

In practice the direction vector $[f * \Psi_1(\vec{x}), f * \Psi_2(\vec{x}), f * \Psi_3(\vec{x})]$ must be discretized into a finite number of directions. Otherwise every grid point is classified as a maximum since no gradient vector points exactly to another grid point. The number of directions into which the gradients are discretized influences the connectivity of the structures that are found to some extent. The more directions are used, the more local maxima will be found. A discretization into the 26 directions to the nearest neighbors within the three dimensional neighborhood has been chosen in the actual implementation.

Figure 13 shows the extracted gradient maxima at different scales from a 3D chromosome data set. The voxel intensity scales up according to the magnitude of the gradient maxima. Red spots emphasize small magnitude while large ones shift to green. Note that the magnitudes increase close to the dominant features at the coarser scales while they fall off in noisy regions. Particularly, the number of maxima decreases drastically from fine to coarse scale.

3.1. Derivative Wavelets

So far, nothing has been said about the wavelets which are involved in the edge detection process. The only condition is that they have to be the derivatives of a smoothing function $\theta(x)$. In general, equation (1) does not hold for the scaling function Φ and the wavelet Ψ which define a multiresolution analysis [20, 3, 13].

Mallat et al. [21] developed pairs of wavelets and smoothing functions in Fourier space from which derivative wavelets with arbitrary approximation properties can be obtained. The scaling function Φ and the wavelet Ψ with n vanishing moments are defined in the frequency domain by

$$\hat{\Phi}(\omega) = \left(\frac{sin(\omega/2)}{\omega/2}\right)^{2n+1}, \quad \hat{\Psi}(\omega) = i\omega\left(\frac{sin(\omega/4)}{\omega/4}\right)^{2n+2}.$$

The corresponding smoothing function is defined by

$$\hat{\theta}(\omega) = \left(\frac{sin(\omega/4)}{\omega/4}\right)^{2n+2}.$$

It satisfies condition 1. Throughout the implementation we chose n equal to 1. This yields the quadratic spline wavelet and the cubic spline as the smoothing function as shown in Figure 6. The finite impulse response filters that are nec-

[†] Ψ is said to be a wavelet if $\int_{-\infty}^{+\infty}\Psi(x)dx = 0$, and Ψ has exponential fast decay at infinity.

essary to perform the discrete wavelet transform are completely specified in [21].

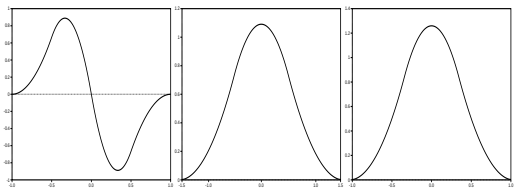


Figure 6: Quadratic spline wavelet Ψ with one vanishing moment, scaling function Φ, and cubic spline smoothing function θ.

3.2. Structural Analysis

Summarizing the results we conclude that the dominant structures remain present all over the multiscale versions of the 3D data set. On the one hand, since the gradient maxima are computed from the low frequency parts at every level the entropy of the extracted information slightly decreases. Less dominant structures disappear at one of the coarser scales. On the other hand, it is just the evolution of the gradient maxima across scales which allows for the characterization of the original signal. Therefore the maxima information from different scales has to be combined. This is described in the next paragraph.

From the asymptotic decay of the wavelet coefficients to the finest scale the smoothness of the signal can be estimated. A common measure for the smoothness is the Hölder Lipschitz regularity‡ [12, 22].

If the signal has an isolated singularity at x_0, then there is a path, or a chain, from the coarsest scale to the point x_0 at the finest scale along which only local maxima exist. Two locations x_1^j and x_2^{j+1} at different scales along the path are restricted to differ only slightly in space, that is, $|x_1^j - x_2^{j+1}| \leq A$, where A depends on the used wavelet. From the decay along these maxima chains the Hölder exponent can be estimated. The signal is uniformly Lipschitz regular with Hölder exponent α over an interval Ω if there exists a constant K such that for all $x \in \Omega$ the decay along the chains behaves as

$$|W^j f(\vec{x})| \leq K \cdot |2^j|^\alpha. \tag{2}$$

Thus, all singularities of the signal can be detected by connecting those maxima which proceed from a certain scale to the next coarser one. Recall from (2) that in case of sharp variation points the maxima at all scales have to exist. Consequently, instead of estimating the Hölder exponent numerically it suffices to classify points in terms of the number of maxima that are pointing to them. Note that all gradients along a chain must have the same direction within some tolerance. As will be described later it can be used to enhance the connectivity between maxima from different scales.

Figure 7 shows a 2D example. The pixel brightness directly corresponds to the gradient magnitude. Note that the brightness along the edges increases from scale to scale while between them it is reduced or extinguished.

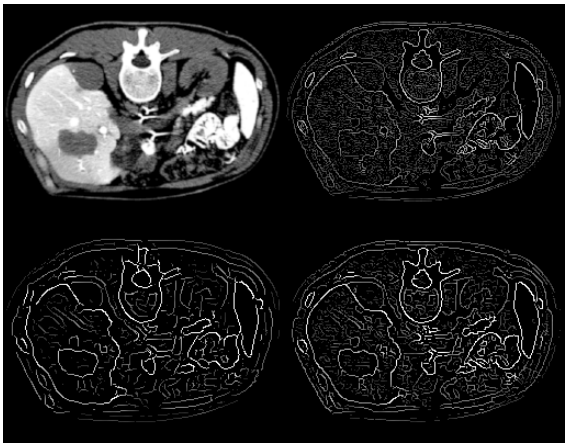


Figure 7: Gradient magnitude at different scales.

Figure 14 outlines the principal behavior in more detail. In both examples we show the original data set and the locations at the finest scale where a closed maxima chain is pointing to. In the left images of the vessel structures and the chromosome, respectively, the voxel opacity was linearly adjusted to the gradient magnitude, while in the right images a constant value was used for all voxels. This is quite similar to the volume gradient shading method proposed in [17].

The rendering methods we used differ substantially from the first to the second example. 3D texture based slice projection was chosen for the chromosome, while for the vessel structures a standard ray-casting algorithm was applied. The reader might have noticed that on the right only the boundary regions are present while on the left also structures in between the boundaries can be seen. This is explained in more detail in the next chapter.

It can be clearly realized that the dominant features remain visible across scales. Fuzzy or noisy structures are removed since they do not contribute to closed chains, i.e., they disappear at one of the coarser scales.

In the presence of stationary white noise this phenomenon can be formally explained. The expectation value of the squared wavelet transform of white noise with variance σ, $E(|Wn(s,x)|^2)$, equals $\| \Psi \|_2 \, \sigma^2/s$. It is thus inversely proportional to the scale s. The decay of the wavelet coefficients behaves contrarily to that, which can be observed near edges. Specifically, all scale-space maxima curves which disappear

‡ A function $\in L^2(\mathcal{R}^3)$ is said to be uniformly Lipschitz on a set $\Omega \subset \mathcal{R}^3$ with Hölder exponent α and $0 < \alpha < 1$, if there exists a constant K so that $\forall \vec{x}_0, \vec{x}_1 \in \Omega : |f(\vec{x}_0) - f(\vec{x}_1)| \leq K|\vec{x}_0 - \vec{x}_1|^\alpha$.

at one of the coarser scales do not belong to the dominant features.

We should mention here that we only use the maxima information to detect and indicate the relevant structures. We do not try to reconstruct the original data set from its maxima, as proposed in [21, 8].

3.3. Optimization

In this section two major problems of the outlined approach should be further discussed. First, how can we compute the multiscale information without increasing the memory requirement too much? Unfortunately, computing the information at all scales in advance is not possible. Second, how can we avoid undesirable gaps or holes which basically arise from the discretization of the wavelet transform and gradients?

The first problem can be solved quite easily. This is due to the compact support of the involved finite impulse response filter kernels: the wavelet transform and the gradient maxima at a certain level and at a certain position can be computed locally. All that has to be stored is one additional copy of the data. Each entry holds an unsigned integer value in which the gradient direction and its magnitude are saved. The quantized direction is coded in the 16 least significant bits whereas the other bits are used to code the magnitude. After processing a certain scale the maximal gradient magnitude which appeared in that position together with its direction remain.

The second problem originates from the discretization of the continuous wavelet transform. Since we proceed on a dyadic scale and also discretize the gradients into a small number of directions, it it almost impossible to obtain closed maxima chains across different levels. From one level to another local maxima slightly change their position and shift in space. Depending on the support of the used basis functions all maxima chains which point to a certain location $\vec{x}_0$ at the finest scale lie within a cone of influence defined by $|\vec{x}-\vec{x}_0| \leq A \cdot s$. Therefore, a maximum at scale 2^j and position $\vec{x}_0$ possibly influences points at scale 2^{j-1} within a neighborhood $|\vec{x}-\vec{x}_0| \leq A'$. The constant A' depends on the support of the involved basis functions and determines the range within which we seek for maxima at the next coarser scale.

Additionally, standard image processing techniques which help to close small gaps within a level and try to extend chains as long as possible can be applied. Figure 8 shows parts of a data set before and after these corrections. In the right image small gaps are removed by trying to enlarge segments which are broken into disconnected pieces. The way we proceed is to look for structures which do not have direct neighbors that belong to the same segment, but which have neighbors two steps away that do. Once we have found these structures, we put additional maxima with the same direction into the data at this level.

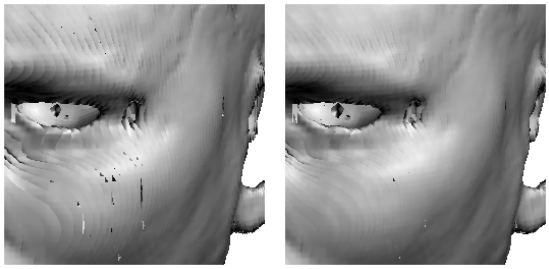


Figure 8: Structural enhancement by closing small gaps and holes.

Obviously, the gradients at these next neighbors must point into the same direction. Notice that the tolerance within which two gradients are said to have the same direction influences the connectivity of the found structures to some extent.

3.4. Edge Representation

Finally, we must store the extracted multiscale edge representation in such a way that allows its direct integration into the rendering process. During the structural analysis an 8 bit field is generated for each voxel. It indexes the scale at which a multiscale edge passes through that voxel. The number of bits used equals the number of scales that are processed.

If bit j is set at position $\vec{x}$, then a maximum is located at that position at level 2^j. Each time a maximum at location $\vec{x}_j$ and scale 2^j is pushed down to the next coarser scale, 2^{j+1}, to position $\vec{x}_{j+1}$, then bit $j+1$ is set in the bit mask at $\vec{x}_{j+1}$. From each bit pattern or tag it is now easy to determine the multiscale edges which pass through the corresponding voxel. Basically, 4 different patterns can be distinguished (see Figure 9).

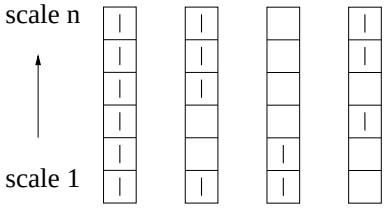


Figure 9: Categories of bit pattern within each tag.

In particular, we seek for those structures which are coded by the first tag in Figure 9: a maxima chain points exactly to the corresponding location. Since it proceeds across all levels we will call it a closed maxima chain hereafter. These are the sharp variation points that are preserved at every scale. The second and third pattern correspond to small fuzzy or

noisy fluctuations which are removed at one of the coarser scales. The last one localizes maxima at the coarser scales which are shifted into other voxels but do not define closed maxima chains.

Since in practice we only compute the first n levels, where n is small (3-4), we also store that level in the bit field at which a maximum at a certain position was originally detected. Now we have the complete edge information at multiple scales available.

4. The Rendering Pass ...

From the structural analysis of the discrete volume data we obtain an additional data set of equal resolution in which the multiscale edge information is coded. Each bit pattern indicates whether a closed maxima chain is pointing to the corresponding location in the original signal, and also whether a maximum was found at this location originally. Now, we seek a method that allows us to integrate the edge information into the rendering process. Recall that the bit field defines some kind of skeleton which separates different materials from each other: it builds segments which span the material boundary regions.

In direct volume visualization algorithms the amount of light impinging on the view plane at a certain position is simulated by evaluating the well known volume rendering integral [27, 16, 14]

$$I(t_0,t_1) = \int_{t_0}^{t_1} q(t) e^{-\int_{t_0}^{t} \sigma(s)ds} dt$$

along each ray of sight (see Figure 10). It sums up the contributions of the volume emission $q(t)$ along the ray, which is scaled by the optical depth according to the volume absorption $\sigma(s)$.

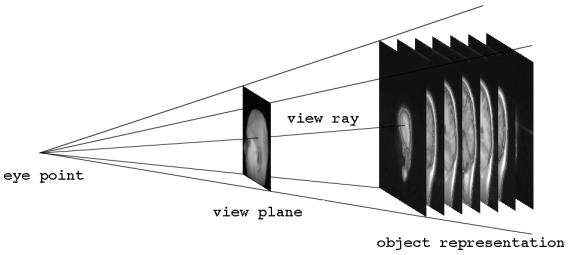


Figure 10: *Volume ray-casting.*

Traditionally, the evaluation of the integral is performed using an Eulerian sum: the ray is split into segments of equal length over which the source term and the opacity are assumed to be constant. The continuous integral evaluates into a discrete sum over the segments along each ray:

$$I(t_0,t_1) \approx \sum_{k=0}^{n} q_k \sigma_k \prod_{i=0}^{k-1} (1-\sigma_i). \qquad (3)$$

The complexity of the rendering process is proportional to the number of samples that have to be reconstructed. Commonly, $O(n^3)$ samples are evaluated by tri-linear or higher order interpolation of the original voxel values. At every imaging step the whole volume is re-sampled. The complexity can be improved by adaptively choosing the sampling frequency according to the importance and the range of the underlying data [16, 4].

The first approach to deal with the multiscale edge representation is to use it as a binary classifier which determines the visible structures. Equation (3) is evaluated only if the ray hits a voxel that belongs to a bit pattern of interest. Otherwise the segments opacity is set to zero.

The major limitation of this approach is that the rendered images only show the boundary regions, but on the other hand it has two major advantages. First, non boundary regions can be traversed very quickly. This is accomplished by using ray-generators [1, 24] (see Figure 11) which rely on a uniform subdivision of the underlying domain. All hits between a ray and the volume cells can be found incrementally. If an intersection with one of the edges is determined, then equation (3) is evaluated on the original data set.

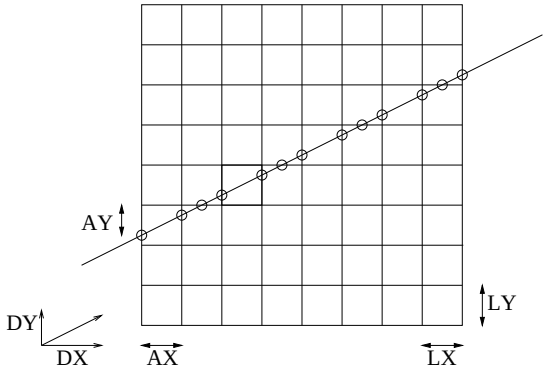


Figure 11: Uniform subdivision and cell traversal process in two dimensions. *LX* and *LY* are the distances from plane to plane. *DX* and *DY* specify the normalized ray direction. *AX* and *AY* are the distances from the entrance point of the ray to the planes parallel to the axes. The distances along the ray from the entrance point to the planes are *PX* and *PY*. Given $TX = \frac{LX}{DX}$, $TY = \frac{LY}{DY}$, and $PX = \frac{AX}{DX}, PY = \frac{AY}{DY}$, then the minimum of the distances *PX* and *PY* is the actual distance to the exit point. The new values of *PX* and *PY* can be computed from their old values and the values of TX and TY.

The second feature is the possibility to render the 3D edges at arbitrary scales. Since the number of gradient maxima decreases drastically at the coarse scales the rendering pass can be accelerated. Obviously, we only obtain the information at low resolution, but often it suffices for fast preview purposes. The first three images in Figure 15 demonstrate both acceleration techniques. For each of the proposed methods the percentage of samples which were reconstructed from the original domain is given.

The first image shows the original data set which was rendered in full resolution. Thresholding was applied during resampling to highlight the principal structure. Images 2 and 3 show the 3D edges at the finest scale and at one of the coarser scales rendered with the described ray-casting technique. Thresholding was applied only if a hit with a structure was determined. Note that a very large number of samples is saved while we already obtain the dominant structures.

The described rendering techniques are particularly useful to visualize the boundary regions of the volume data set. On the other hand, since we already store the original data in full resolution it is also possible to visualize material soft tissue. Therefore, the reconstruction process is performed not just around the edges but also in between them. In general, volume rendering acceleration techniques which try to adapt the integration step size according to the material properties can be applied. In order to ensure that all edges are hit the modified step sizes are only used between the intersection points.

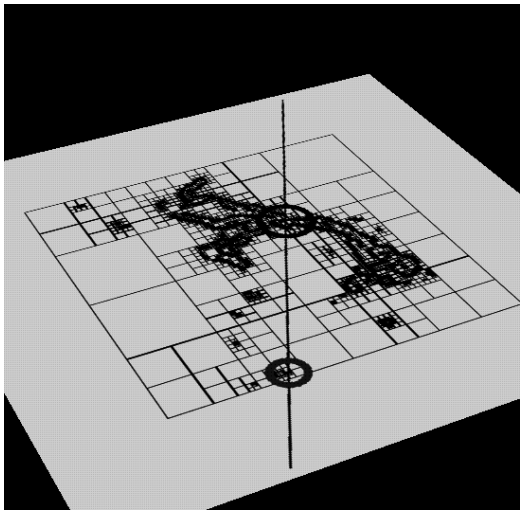


Figure 12: Quadtree based ray traversal.

5. ... on Sparse Representations

A major limitation of the proposed rendering approaches is the amount of memory that is used. In addition to the original data the multiscale edge representation must be stored. It takes at least an 8 bit field of equal resolution as the original one. The necessity to store the information in a less memory intensive representation becomes a dominant goal.

In order to accommodate this requirement we propose to generate an octree data structure in which the material is coded. Therefore, a criterion must be chosen where to refine the octree. The condition under which a refinement takes place is precisely described by the edge skeleton, but additionally, the local variance within certain regions must be considered.

New nodes are always constructed if the parent node includes at least one edge. If the parent node does not include an edge, then the local variance within this node is used to decide whether it has to be split or not. Figure 12 demonstrates the basic idea in flatland. The subdivision is performed just around the edges of a 2D slice out of the 3D chromosome data set. Note that the quadtree allows the accurate decoding of regions close to the edges, while in homogeneous regions a less accurate approximation is obtained.

Of course, the traversal process gains complexity since each ray must be passed through the octree representation, but on the other hand, a huge amount of memory is saved. The rightmost image in Figure 15 shows an example where the chromosome data set was rendered on the reduced pointered octree data structure with about 30% of the original memory. It is obvious that the basic features remain unchanged while only between the boundaries the reconstruction error is increased.

6. Results

We have implemented the complete system described in this paper on a modern single processor system. All results were computed on a R4400 processor running with 75 Mhz. Both test data sets have a resolution of 256^3. The first example shows a chromosome data set scanned by a confocal microscope, the second example shows a mr-angiographic scan of human vessel structures. One intermediate unsigned integer field has to be allocated during processing, and one additional byte field is needed for the final rendering pass.

The most expensive task is the computation of the gradient maxima at each level. In practice we experienced that it suffices to incorporate the first three levels since structures become too fragmented on the next coarser ones. It took roughly 14 minutes to process each level separately, and additional 9 minutes to detect the closed maxima chains.

Figure 13 clearly demonstrates the decreasing number of local gradient maxima from fine to coarse scale. Each maxima is highlighted by increased opacity of the corresponding voxel. Note that maxima which result from noise are almost completely removed at one of the coarser scales.

The extraction of relevant structures is outlined in Figure 14. It would be quite difficult to achieve the same results without thresholding the original data set. The distinction between different kinds of features by only considering the original data seems to be rather impossible. The multiscale analysis enables a precise characterization of the visible structures.

An important issue is that our approach does no longer depend on a user defined threshold to reconstruct the boundary regions. Only those regions which really correspond to the boundaries between different materials are considered.

Figure 15 clearly demonstrates the performance tuning of our method. The complexity of the rendering process is reduced to considerable extent: the number of samples that have to be reconstructed is proportionally related to the number of closed maxima chains that have been found. Of course, there is a slight overhead because we also have to sample the edge field, but on the other hand, this only involves a test with the cell neighbors. Ray-casting the finest scale edges took 32 % of the time needed to render the original data, for the coarse scale edges it took 11 %.

A speed up of some factors thereby achieving similar visual quality is what we suppose in case of typical data sets. Comparable visual results will be obtained by applying Levoys gradient shading method. But in contrast to our approach the whole data set must be re-sampled. We should mention that there is no reason why we should not use a density interval to further classify the found edges. Another degree of freedom is added in this case.

7. Discussion and future work

We have described a multiscale method to detect and enhance structures from volumetric data and to integrate this information into the rendering process. The proposed technique copes excellently with certain noise distributions and thus allows the discrimination of the relevant structures. Furthermore, the data analysis is speed up to some extent since the object space multiscale representation is integrated into the imaging process. The user can arbitrarily switch between different levels of detail in order to speed up the rendering pass.

A less memory intensive octree representation can be constructed from the original signal by taking into account the edge information in conjunction with the local variance within the data. Good compression ratios are obtained while achieving sufficient rendering quality.

In most cases the proposed method is significantly faster than fixed step rendering techniques without deteriorating the final image quality. In the contrary, the dominant features are extracted as they are and can be rendered prominent.

Obviously, our method does not differ from other methods in the fact that the structures within the volume strongly influence the rendering times and the compression ratios. However, referring to the statements of other volume rendering specialists, who argue that almost 80% of typical volumes are empty or almost homogeneous, our method decreases the rendering times and the used memory to considerable extent.

In the future we plan to spend additional work on a detailed investigation of the extracted structures. The important question is how to extend them in order to build closed segments. Techniques from image processing like morphological operations, region growers or ballooning methods should be tried.

Another still open question is how we can distinguish between different structures in terms of their multiscale representation. Higher order segmentation methods have to be applied in order to find connected segments. So far, all structures are binary classified and the distinction is made during the rendering process by taking into account a material threshold. However, a more detailed distinction would be helpful.

8. Acknowledgement

Special thanks to Dr. Andres Kriete from the Institute of Anatomy, University Giessen, for providing the chromosome data set.

References

1. J. Amanatides and A. Woo. A Fast Voxel Traversal Algorithm for Ray Tracing. In *Eurographics'87, Conference Proceedings*, 1987.
2. J. Canny. A Computational Approach to Edge Detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.8, No.3, pages 679-698, 1986.
3. C. K. Chui. An Introduction to Wavelets. In *Wavelet Analysis and its Application*, Vol.1, Academic Press, 1992.
4. J. Danskin and P. Hanrahan. Fast Algorithms for Volume Rendering. In *ACM Workshop on Volume Visualization 1992, Conference Proceedings*, pages 91-98, 1992.
5. I. Daubechies. Ten Lectures on Wavelets. In *CBMS-NSF Series in Applied Mathematics*, Vol. 61, SIAM 1992.
6. R. Drebin, L. Carpenter and P. Hanrahan. Volume Rendering. In *Computer Graphics*, Vol. 22, No. 4, pp. 65-74, 1988.
7. H. Edelsbrunner and E. Mucke. Three-Dimensional Alpha Shapes. In *ACM Transactions on Graphics*, Vol. 13, pp. 43-72, 1994.
8. B. Guo. Interval Set: A Volume Rendering Technique Generalizing Isosurface Extraction. In *IEEE Visualization'95, Conference Proceedings*, pp. 3-11,1995
9. B. Guo. A Multiscale Model for Structure-Based Volume Rendering. In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No.4 , pp. 291-301, 1995.
10. M. Gross, L. Lippert, A. Dreger and R. Koch. A new Method to Approximate the Volume Rendering Equation Using Wavelets and Piecewise Polynomials. In *Computers and Graphics*, Vol.19. No.1, 1995.
11. A. Grossmann. Wavelet transform and edge detection.

In *Stochastic Processes in Physics and Engineering*, 1986.

12. S. Jaffarth. Wavelets and nonlinear analysis. In *Wavelets: Mathematics and Applications*, edited by J. Benedetto and M. Frazier, pp. 467-505, 1993.

13. B. Jawerth and W. Sweldens. An Overview of Wavelet Based Multiresolution Analysis. In *SIAM Review*, Vol.36, No.3, 1994.

14. W. Krueger. The Application of Transport Theory To Visualization of 3D Scalar Data Fields. In *IEEE Visualization'90, Conference Proceedings*, pp. 12–15, 1990.

15. D. Laur and P. Hanrahan. Hierarchical Splatting: A Progressive Refinement Algorithm. In *Computer Graphics*, Vol.25, No.4, pp. 12–15, 1990.

16. M. Levoy. Efficient Ray Tracing of Volume Data. In *ACM Transactions on Graphics*, Vol.9, No.3, pp. 245-261, 1989.

17. M. Levoy. Display of Surface from Volume Data. In *IEEE Computer Graphics and Applications*, Vol.8, No.3, pp. 29-37, 1990.

18. W.E. Lorensen and H.E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Computer Graphics*, Vol.21, No.4, pp. 163-169, 1987.

19. D. Maar and E. Hildreth. Theory of edge detection. In *Proceedings Royal Soc. London*, Vol.207, pp. 187-217, 1980.

20. S. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.11, No.6, pp. 674-693, 1989.

21. S. Mallat and W. Hwang. Singularity Detection and Processing with Wavelets. In *IEEE Transactions on Information Theory*, Vol.38, No.2, pp. 617-643, 1992.

22. S. Mallat and S. Zhong. Characterization of Signals from Multiscale Edges. In *IEEE Transactions on Patten Analysis and Machine Intelligence*, Vol.14, No.7, pp. 710-730, 1992.

23. P. Cignoni, C. Montani, E. Puppo and R. Scopigno. Optimal Isosurface Extraction from Irregular Volume Data. In *ACM Symposium on Volume Visualization, Conference Proceedings*, pp. 31-39, 1996.

24. H. Mueller. Ray Tracing Complex Scenes by Grids. Karlsruhe 1989.

25. S. Muraki. Approximation and Rendering of Volume Data Using Wavelet Transforms. In *IEEE Computer Graphics and Applications*, Vol.13, No.4, pp. 50-56, 1993.

26. S. Muraki. Multiscale 3D Edge Representation of Volume Data by a DOG Wavelet. In *ACM Symposium on Volume Visualization 1994, Conference Proceedings*, pp. 35-42, 1994.

27. P. Sabella. A Rendering Algorithm for Visualizing 3D Scalar Fields. In *Computer Graphics*, Vol.22, No.4, pp. 51-58, 1988.

28. G. Sakas and S. Walter. Extracting Surfaces from Fuzzy 3D-Ultrasound Data. In *ACM Siggraph'95, Conference Proceedings*, pp. 465–474, 1995.

29. R. Westermann. A Multiresolution Framework for Volume Rendering. In *ACM Symposium on Volume Visualization 1994, Conference Proceedings*, pp. 51-58, 1994.

30. R. Westermann. Compression Domain Rendering of Time-Resolved Volume Data. In *IEEE Visualization 1995, Conference Proceedings*, pp. 168-176, 1995.

31. J. Wilhelms and A. van Gelder. Octrees for Faster Isosurface Generation In *ACM Transactions on Graphics*, Vol. 11, pp. 201–227, 1992.

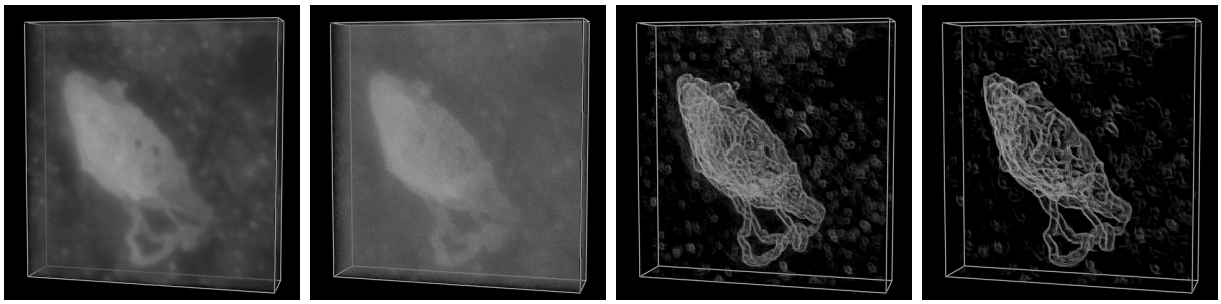


Figure 13: (a) The original data in full resolution. (b) Magnitudes of the gradient maxima at the first scale, (c) at the second scale, and (d) at the third scale.

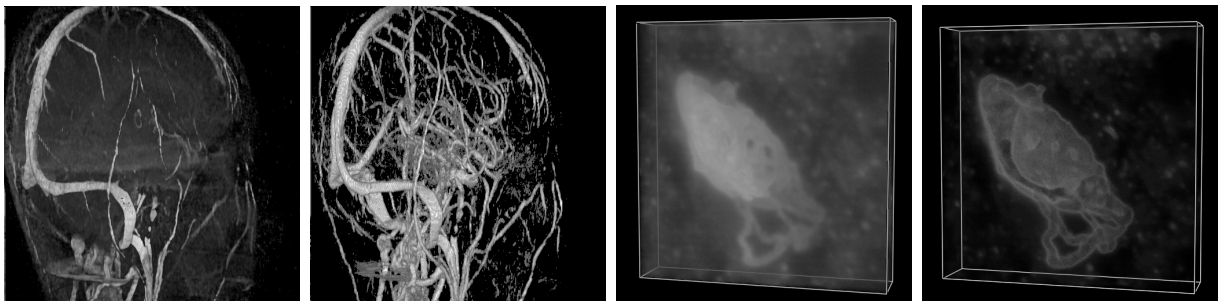


Figure 14: (a) Vessel data set without preprocessing. (b) Ray-casting based on maxima chains with integrated rendering of soft tissue. (c) Chromosome data set without preprocessing. (d) 3D texture projection based on closed maxima chains.

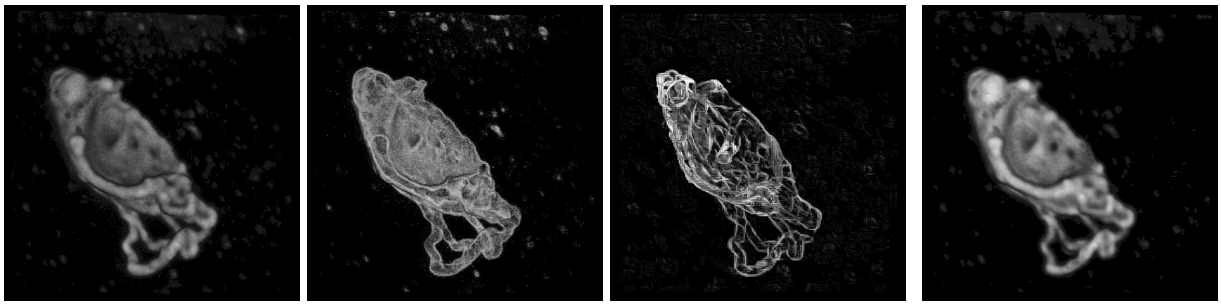


Figure 15: (a) Original data set rendered on full resolution with 100% samples. (b) Ray-casting of finest scale edges with 15% samples and (c) coarse scale edges with 0.5% samples. (d) Ray-casting performed on the reduced octree reconstruction with 30% memory.