

Automatic Adjustment of Transfer Functions for 3D Volume Visualization

Christof Rezk-Salama, Peter Hastreiter, Jörg Scherer, and Günther Greiner

Computer Graphics Group, University of Erlangen
Am Weichselgarten 9, 91058 Tennenlohe, Germany
Email: rezk@informatik.uni-erlangen.de

Abstract

In most volume rendering scenarios implicit classification is performed manually by specification of a transfer function, that maps abstract data values to visual attributes. An appropriate classification requires both specialized knowledge of the interesting structures within the data set as well as the technical knowhow of the computer scientist. Recent automatic data-driven techniques are very-well capable of separating different regions in the data set. However, their applicability in practice is limited, since they do not contain any information about the critical structures which are of interest. In this scenario we propose an efficient and reproducible way to automatically assign transfer function templates, which include individual knowledge as well as personal taste. The presented approach is based on dynamic programming and was successfully applied in medical environment.

1 Introduction

Interactive techniques have become invaluable for the three-dimensional (3D) visualization of abstract data fields. Its application in science, medicine and engineering has lead to an improved spatial understanding and to comprehensive insights into three-dimensional structures. The process of image generation from abstract data fields in general can be described as a three-tier approach (Fig. 1). The raw data field is usually represented by a fi-

nite number of discrete sample points located on a specified grid structure. The preprocessing step comprises filtering, interpolation and resampling of the data. In the subsequent mapping or *classification* step, the abstract data values are transformed into geometric and graphical primitives. According to the material properties or physical measurements that are represented by the abstract data values, at this stage visual attributes like color or transparency are assigned. For most application scenarios, an appropriate data mapping is a non-trivial task, which requires detailed knowledge of the structures inherent in the data. Additionally, the assignment of visual attributes is often ambiguous and usually influenced by the personal taste of the user. Finally, in the rendering step these graphical primitives are used to generate meaningful images.

2 Volume Rendering

There is a variety of different visualization approaches for scalar volumes in multiple application scenarios. Recent approaches are categorized into indirect methods, such as isosurface extraction [8], and direct methods, that immediately display the voxel data. We will focus on interactive direct methods.

For visualization of 3D scalar data on a uniform grid, texture-based volume rendering has proved superior, being able to produce high quality images at interactive frame rates. Due to the large number of trilinear interpolations

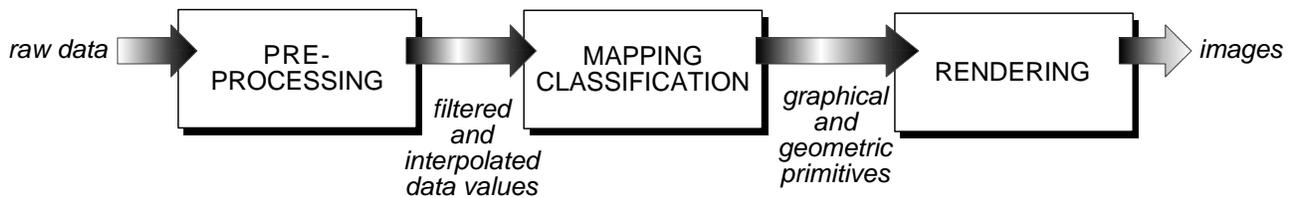


Figure 1: The visualization process can be described as a three-tier approach

that must be processed in order to produce images of high quality, the availability of direct volume rendering has yet been restricted to high-end workstations and special purpose graphics hardware. With computer graphics hardware becoming more flexible and powerful, high quality visualization is becoming available also on low cost hardware platforms [11]. This expands the application of direct visualization approaches to a wider field of scientific research and practice.

2.1 Texture-Based Approaches

The most important texture-based approach was introduced by Cabral [1], who used the 3D texture mapping capabilities of high-end graphics workstations. The volume data set is represented by a stack of adjacent polygon slices. If 3D textures (OpenGL 1.2) are supported by hardware, it is possible to render slices parallel to the viewport with respect to the current viewing direction (see Fig. 2.1 *left*). Changing the camera position requires a re-computation of these *viewport-aligned* slices. Finally, in the compositing step the slice polygons, textured with the 3D image information, are blended back-to-front onto the image plane, which results in a semi-transparent view of the volume. Interactive frame rate are achieved, since trilinear interpolation of texture samples is supported by rasterization hardware.

Contrary to this, if only 2D texture-mapping is available, the slicing planes must be set up parallel to the coordinate axes of the rectilinear data grid (*object-aligned* slices, Fig. 4 *right*). This allows an efficient substitution of trilinear by bilinear interpolation.

However, the orientation of the slicing direction must be adapted, if the viewing direction changes by more than 90 degrees. For interactive visualization this means, that three copies of the data set must be kept in main memory, one set of slices for each slicing direction respectively. Each slice is then rendered as a planar polygon textured with the image information obtained from a 2D texture map and blended onto the image plane. This basic idea of using object-aligned slices to substitute trilinear by bilinear interpolation was first presented by Lacroute and Levoy [7], although the original implementation did not exploit texturing hardware.

The 3D texture-based method has been significantly expanded by Westermann and Ertl [13]. They additionally store gradient information in the 3D textures, which leads to a fast direct multi-pass algorithm for shaded isosurface display. Based on this implementation, Meißner et al. [10] have provided a method to allow for diffuse illumination in semi-transparent volume rendering. In [11] both approaches were successfully adapted to standard PC hardware.

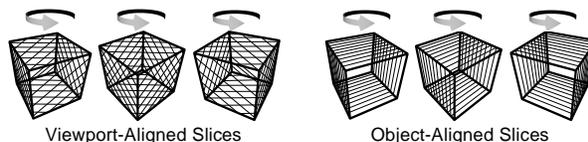


Figure 2: Viewport-aligned slices (*left*) in comparison to object aligned slices (*right*) for a spinning volume object.

3 Transfer Functions

Although the value of direct volume rendering in medicine, science and engineering is indisputable, the applicability suffers from some severe limitations. In many critical application scenarios adequate techniques to compare examination results are required. Physicians and engineers are in demand of a reliable method to reproduce the visualization in order to discuss and re-evaluate their conclusions. According to this, another problem is that manual classification is extremely time-consuming. From experience this often leads to visualization results which are hardly reproducible and thus not valuable for documentation in practice.

As mentioned above, for 3D visualization and analysis, classification of the volume data must be performed by specifying visual attributes for each data value. Especially for tomography data this is a non-trivial task. The assignment is usually performed by means of a transfer function, that maps data values to values for color and opacity. Besides manual specification, there are several image- and data-driven approaches for automatic and semi-automatic generation as will be outlined below.

3.1 Hardware Support

An important aspect, that must be taken into account is the placement of the transfer function within the rendering pipeline. In order to remove visual artifacts, the scalar values at fixed grid positions usually must be interpolated and re-sampled. In this context it is important whether the transfer function is applied prior to or after the interpolation (see Figure 3). Although the pre-interpolative application of a transfer significantly removes artifacts, more precise visual results are obtained by a post-interpolative transfer function, since continuity is usually assumed in the data domain, but not in color space. However post-interpolative transfer functions are only supported by high-end graphics workstations.

This is of special importance for the reproducibility of visualization results on different hardware platforms.

3.2 Templates

In almost every scientific application transfer functions for color and opacity are manually generated using some kind of visual editor. A simple example of such a transfer function assignment is the specification of a linear grey-value window, as it is usually done by a clinician viewing tomographic data. More complex transfer function try to assign different colors to different regions in the data, enabling to obtain an implicit segmentation of significant structures based solely on the data values. This approach however depends on the interactivity provided by the underlying ren-

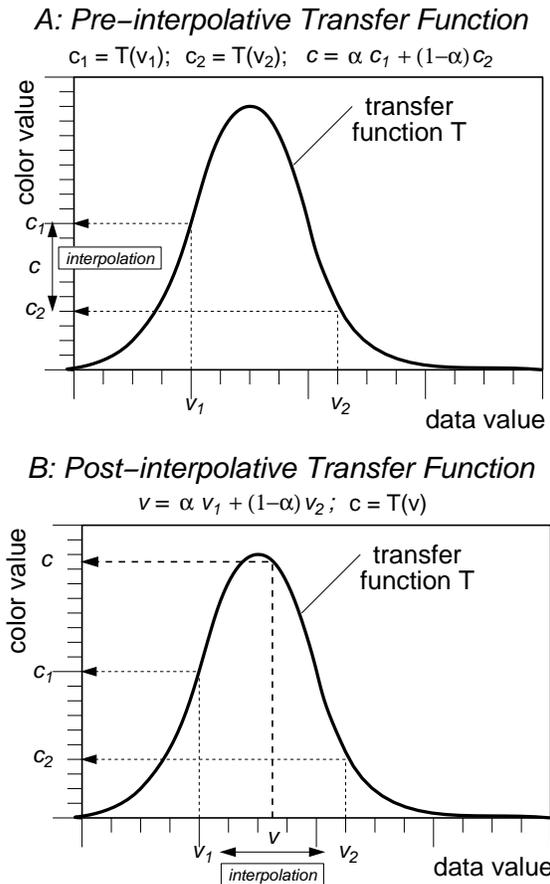


Figure 3: Applying the transfer function in advance (A) or after the interpolation (B) leads to different results.

dering algorithm, as the direct visual feedback within the 3D viewer is indispensable for purposive work.

Since manual transfer functions are usually generated in an iterative process and are based on heuristics, specialized knowledge and personal taste, the whole assignment procedure is rather time-consuming and hardly reproducible. In order to speed up manual assignment, application-specific templates are used, which were adjusted manually to the patient’s individual data. Transfer function are usually stored and applied as lookup tables, one-dimensional arrays of color values. Templates can also be stored as piecewise linear mappings or functions of higher order. Until now manual heuristic methods are the only approaches that include detailed knowledge about the structures inside the volume data and thus are the only approaches that are applicable in practice.

3.3 Related Work

Various approaches for automatic or semi-automatic assignment have been proposed recently. These approaches can be categorized into image-driven and data-driven techniques. Marks et al. [9] have proposed the concept of design galleries for setting visual parameters in general. This example of a semi-automatic image-driven method generates a huge amount of images with different parameter settings, allowing the user to select the image with the optimal visual result. He et al. [5] have developed a technique for semi-automatic transfer function generation using a stochastic search algorithm. This method generates images using an initial population of transfer functions. Subsequently the user selects the best set of images. With this selection a stochastic search is started, that generates a new population of transfer functions. The applicability of these image-based approaches is limited, since they are neither fast nor purposive, and thus not efficient enough for clinical routine. According to data-driven approaches Fang et al. [3] have proposed a

technique based on intensity mappings. The transfer function is modeled as a sequence of 3D image processing operations which allows the user to specify qualitative parameters. A related technique was presented by Sato et al. [12]. Their approach applies 3D image filters in order to accentuate local intensity structures using a multi-dimensional feature space.

Kindlmann and Durkin [6] have recently proposed an interesting data-driven method that takes into account the first and second directional derivatives along the gradient direction of the scalar field. Although the original approach is less applicable for realistic image data, since no information about interesting structures is included, our method to automatically adjust transfer functions is based on this approach, the main aspects of this method are described in detail.

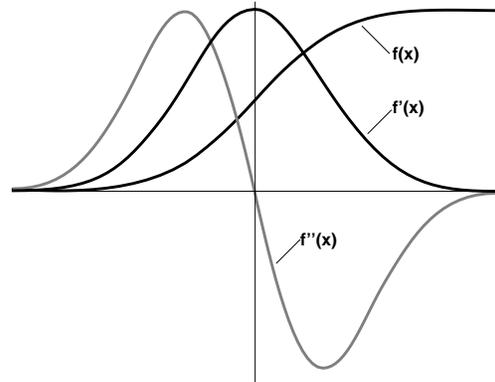


Figure 4: Function $f(x)$ and first and second order derivatives $f'(x)$ and $f''(x)$ in direction of the voxel gradient.

Regarding the data value f in direction of the (non-zero) gradient will result in a monotonically increasing function $f(x)$ as displayed in Figure 4. In image processing, the first and second order derivatives $f'(x)$ and $f''(x)$ are frequently used criteria for boundary detection. Since $f(x)$ is monotonically increasing (and thus invertible), it is possible to express the directional derivatives $f'(x)$ and $f''(x)$ as a function of the data value $v = f(x)$ instead of the position x . Therefore, in practice, for each voxel of the original data set the first and second order derivatives in gradient di-

rection are computed. Subsequently these values are averaged for all voxels with a specific data value v . This results in two functions $g(v)$ for the first and $h(v)$ for the second order derivative. Then, the fraction of these two functions is calculated which results in the *position function*

$$p(v) = \frac{-h(v)}{g(v)} . \quad (1)$$

It describes the average distance of a data point with value v from a boundary in the data set assumed by the underlying boundary model. According to Kindlmann, function $p(v)$ is used to compute a transfer function by applying a boundary emphasis function [6]. However, this calculation is not required for our approach. Although the original algorithm accurately determines boundaries within data sets, its application to realistic data is limited, since the specific visualization problem is not taken into account.

4 Automated Adjustment

In order to bridge the gap between automatic and manual assignment, we present a new method that unifies the advantages of both manual and data-driven strategies. The basic idea of our approach is to use an optimal transfer function as a reference template. This template, which has been manually assigned once for a specific data set, is then automatically adjusted by non-linear distortion to individual data of the same type. In order to search for the optimal distortion of the template function an appropriate similarity metric is required. Based on *non-linear time warping*, a concept known from speech recognition, the optimal non-linear distortion of the data value axis is computed for optimal alignment of the normalized histograms for both data sets.

Dynamic programming (see Sec. 4.1) is used to optimize a one-dimensional non-linear transformation $t(v)$ which is defined on the range of data values v . Let's assume we have manually found an optimal transfer function

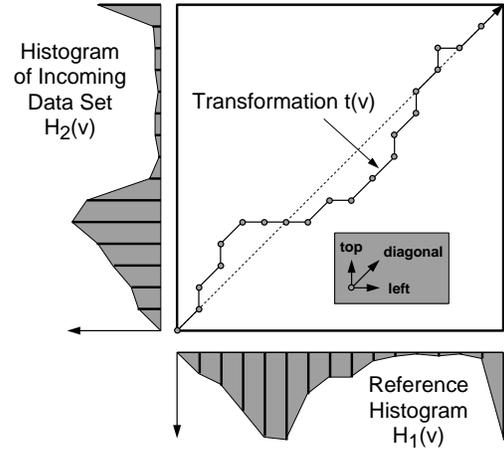


Figure 5: *Non-linear time warping*: a non-linear transformation $t(v)$ of the data value range is optimized for minimal histogram dispersion using dynamic programming

$T_1(v)$ for a specific data set D_1 with a data value histogram $H_1(v)$. In order to adapt this optimal transfer function to a different data set D_2 of the same modality, we use dynamic programming to compute a transformation $t(v)$ that optimizes alignment of both histograms

$$H_2(v) \approx^M H_1(t(v)), \quad (2)$$

according to a specified similarity metric M . For normalized histograms we apply a similarity metric D_t that simply measures the distance

$$D_t(H_1, H_2) = \sum_v |H_1(v) - H_2(t(v))|. \quad (3)$$

Note that this metric is monotonically increasing and thus can be used as cost function for dynamic programming. If the optimal transformation $t(v)$ is found, we apply the same non-linear distortion to adapt the manually assigned transfer function $T_1(v)$ to the new data set D_2 , denoted

$$T_2(v) = T_1(t(v)) . \quad (4)$$

4.1 Dynamic Programming

Dynamic programming [2] is an efficient technique to find an optimal path within a given

parameter space according to a specified cost function (see Fig. 5). The optimization process is performed as a complete search in parameter space, which is based on the assumption that every sub-path of an optimal path is again an optimal path. This assumption is true if the specified cost function is monotonically increasing. Starting at the origin ($v = 0$) of the diagram in Fig. 5, the parameter domain is divided into discrete grid points. The algorithm computes the cost function for the first column, i.e. for every grid point at $v = 0$. Proceeding to the right, for every grid point in the current column, the optimal path is computed. This is achieved by calculating the complete cost of the path to this point by adding the cost of the grid point to the cost of the path to the “cheapest” predecessor. Since the cost function is monotonically increasing, only the path and the cost for every grid point of the actual column need to be stored.

4.2 Enhancements

The described optimization procedure is an efficient method to reuse transfer functions which were heuristically established for a specific data set and to adjust them to different data sets of the same type. However, using the histogram exclusively turned out to be not accurate enough. In order to improve the results the *position function* $p(v)$ of equation 1 proved to be superior to the histogram. The following example from medicine illustrates the accuracy benefit, that comes with the use of Kindlmann’s position function.

Figure 6 displays the histogram $H(v)$ and the position function $p(v)$ for different computed tomography angiography (CTA) data sets of the human head. CTA is a common imaging technique in medical routine, which involves injection of contrast dye in order to analyze intracranial vessels. The significant feature of the histogram is the high peak which is caused by the large amount of soft tissue. However, this characteristic peak has a very limited extent which leads to reduced accuracy. In comparison to this, the

position function $p(v)$ shows a higher number of significant features. The transition between the data values of soft tissue and contrast agent is marked by a high peak in $p(v)$ close to the peak in the histogram. To the left of this peak, there is a significant local maximum with (usually) negative value which describes the boundary between soft tissue and fluid. Additionally, the boundary to bone structures is clearly indicated by another local maximum on the right side relative to the peak in the histogram. Since both, the manually assigned transfer function and the position function $p(v)$ strongly depend on specific boundaries in a data set, $p(v)$ represents a more robust basis for the optimization procedure.

5 Results

The presented functionality was evaluated using computed tomography angiography (CTA) of patients with intracranial aneurysms. In order to treat such vessel malformations, detailed knowledge of the complex vessel topology and the surrounding vulnerable structures such as the cochlea of the inner ear is necessary. Therefore standard DSA (*Digital Subtraction Angiography*) projection images are supplemented by 3D CTA. In this scenario interactive direct volume rendering has lead to a fast and efficient 3D visualization of intracranial vessels [4].

For the automatic classification of CTA data, a transfer function template as displayed in Figure 7 (*right*) has proved its worth in clinical practice. This assignment is used to gen-

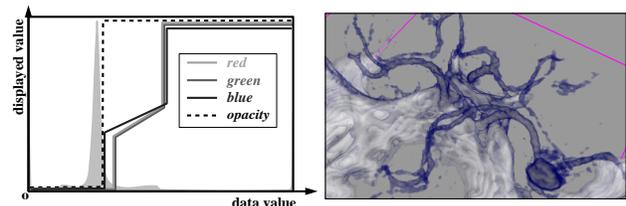


Figure 7: Transfer function template and the resulting image for a CTA data set

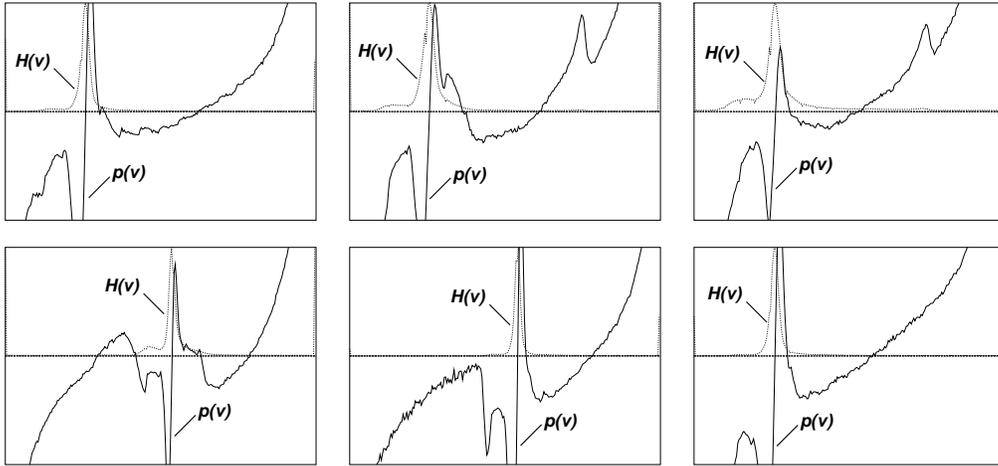


Figure 6: Comparison of histogram $H(v)$ and position function $p(v)$ for six different CTA data sets of the head

erate images, that show the complete vessel information as displayed in Figure 7(left). In this case a completely opaque representation of the vascular structures is desired. Additionally, for anatomical orientation bone structures must be clearly visible. Thus, opacity is set to its maximum for data values that are equal or larger than the value of contrast dye. The color function is then adjusted in order to differentiate between vessel and bony structures. Since local illumination was not available, the blue color component of the transfer functions has been slightly shifted in order to accentuate vessel boundaries and thus enhance the perception of depth.

The automatic adjustment was computed using both the histogram based approach

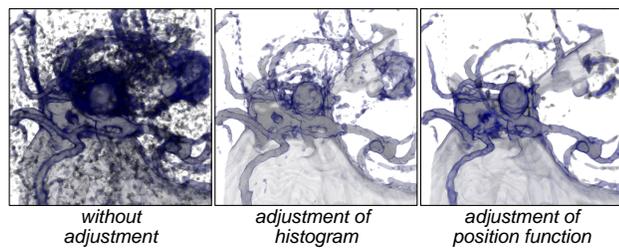


Figure 8: Transfer functions without adjustment, with histogram adjustment and with adjustment based on directional derivatives (position function)

and the method based on directional derivatives. As displayed in Figure 8 the approach based on the derivatives lead to significantly better visual results than the histogram based approach. In very few cases, the results of both approaches are almost equivalent. Volume visualization was performed us-

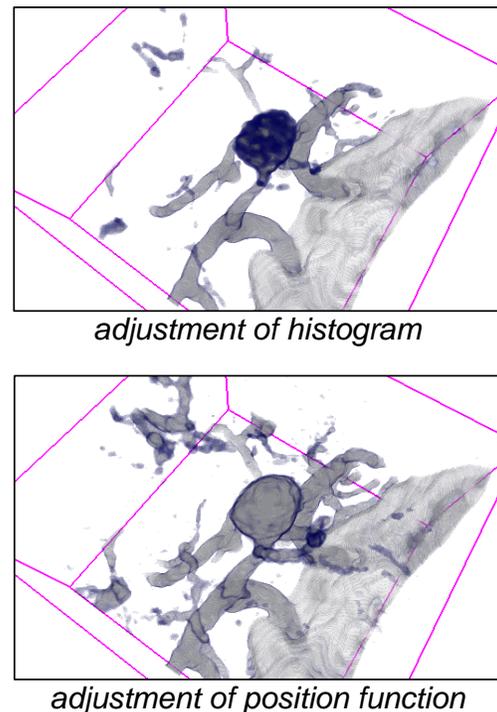


Figure 9: Comparison of adjustment based on histogram and on position function

ing the 3D texture-based approach on an SGI Onyx2 with Base Reality hardware, which supports post-interpolative texture lookup tables. Alternatively an advanced 2D texture-based implementation based on PC graphics boards was used. The automated adjustment of transfer functions also lead to good results with color lookup prior to the interpolation. However the clear delineation of tiny structures is difficult using pre-interpolative color tables. This is due to the nature of pre-interpolative transfer functions and not caused by the adjustment procedure.

6 Conclusion

In order to increase the applicability of direct volume rendering for 3D visualization in scientific environments, we have proposed a reliable method for data-driven classification. Detailed knowledge of the data structure is included into the procedure of transfer function assignment by using predefined templates which are automatically adjusted to individual data.

References

- [1] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. *ACM Symp. on Vol. Vis.*, 1994.
- [2] S.E. Dreyfuss and A.M. Law. *The Art and Theory of Dynamic Programming*. Academic Press, New York, 1962.
- [3] S. Fang, T. Biddlecome, and M. Tuceryan. Image-based transfer function design for data exploration in volume visualization. In *Proc. IEEE Visualization '98*, 1998.
- [4] P. Hastreiter, C. Rezk-Salama, B. Tomandl, K. Eberhardt, and T. Ertl. Fast Analysis of Intracranial Aneurysms Based on Interactive Direct Volume Rendering and CT-Angiography. In *Proc. Med. Img. Comput. and Comp.-Assis. Interv. (MICCAI)*. Springer, 1998.
- [5] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of transfer functions with stochastic search techniques. In *Proc. IEEE Visualization '96*, 1996.
- [6] G. Kindlmann and J.W. Durlin. Semi-automatic generation of transfer functions for direct volume rendering. In *ACM Symp. on Vol. Vis.*, 1998.
- [7] P. Lacroute and M. Levoy. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform. *Comp. Graphics*, 28(4), 1994.
- [8] W.E. Lorensen and H.E. Cline. Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm. *Comp. Graphics*, 21(4), 1996.
- [9] J. Marks, B. Abdalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH*, 1997.
- [10] M. Meißner, U. Hoffmann, and W. Straßer. Enabling Classification and Shading for 3D Texture Based Volume Rendering Using OpenGL and Extensions. In *Visualization '99*, 1999.
- [11] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. *submitted to Graphics Hardware 2000*.
- [12] Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Yoshida, and R. Kikinis. Tissue Classification Based on 3D Local Intensity Structures for Volume Rendering. In *Proc. JAMIT Frontier '97*, 1997.
- [13] R. Westermann and T. Ertl. Efficiently Using Graphics Hardware in Volume Rendering Applications. In *Proc. of SIGGRAPH*, Comp. Graph. Conf. Series, 1998.