

Institut für Computergraphik und
Algorithmen

Technische Universität Wien

Karlsplatz 13/186/2

A-1040 Wien

AUSTRIA

Tel: +43 (1) 58801-18601

Fax: +43 (1) 58801-18698

Institute of Computer Graphics and
Algorithms

Vienna University of Technology

email:

technical-report@cg.tuwien.ac.at

other services:

<http://www.cg.tuwien.ac.at/>

<ftp://ftp.cg.tuwien.ac.at/>

TECHNICAL REPORT

Semantic Layers for Illustrative Volume Rendering

Peter Rautek, Stefan Bruckner, and M. Eduard Gröller

TR-186-2-07-05

April 2007

Keywords: Illustrative Visualization, Focus+Context Techniques, Volume Visualization

Semantic Layers for Illustrative Volume Rendering

Peter Rautek, Stefan Bruckner, and M. Eduard Gröller

April 13, 2007

Abstract

Direct volume rendering techniques map volumetric attributes (e.g. density, gradient magnitude, etc.) to visual styles. Commonly this mapping is specified by a transfer function. The specification of transfer functions is a complex task and requires expert knowledge about the underlying rendering technique. In the case of multiple volumetric attributes and multiple visual styles the specification of the multi-dimensional transfer function becomes more challenging and non-intuitive. We present a novel methodology for the specification of a mapping from several volumetric attributes to multiple illustrative visual styles. We introduce semantic layers that allow a domain expert to specify the mapping in the natural language of the domain. A semantic layer defines the mapping of volumetric attributes to one visual style. Volumetric attributes and visual styles are represented as fuzzy sets. The mapping is specified by rules that are evaluated with fuzzy logic arithmetics. The user specifies the fuzzy sets and the rules without special knowledge about the underlying rendering technique. Semantic layers allow for a linguistic specification of the mapping from attributes to visual styles replacing the traditional transfer function specification.

1 Introduction

Many popular direct volume rendering techniques use a transfer function that maps the measured density values to colors and opacities. These visual properties are usually composited into the final image. More advanced techniques use other volumetric attributes, like gradient magnitude, curvature, or statistical properties and map these values via a multi-dimensional transfer function to visual attributes. All these techniques have in common, that they map attributes of the underlying data on visual appearance via a transfer function. Transfer functions are a powerful tool to achieve various visualizations. However, the specification of transfer functions is a complex task. The user has to have expert knowledge

about the underlying rendering technique to achieve the desired results.

Especially the specification of higher-dimensional transfer functions is challenging. Common user interfaces provide methods to brush in two dimensions. While brushing in 2D is an intuitive method to select regions of interest or specify features, user interfaces for higher-dimensions are more challenging and often non-intuitive.

We propose an alternative method to achieve meaningful mappings from volumetric attributes to visual appearance. Our method enables a multi-dimensional mapping from several volumetric attributes to multiple visual styles. We replace the complex task of multi-dimensional color transfer function design by introducing semantic layers. A semantic layer linguistically describes a mapping from a combination of volume attributes to one visual style. An opacity transfer function is set by the user to specify *what* is shown. The semantic layers describe *how* it is shown. A semantic layer establishes a mapping from semantic values of volume attributes to semantic values of visual styles. The volume attribute *density* is for example described by semantic values ranging from *zero*, over *very low*, *low* and *middle* to *high*. The style *shading* can for example range from *soft*, over *hard* to *cartoonish*. A semantic layer can for example map the semantic value *high density* to the semantic value *cartoonish shading*.

The semantic values are defined using fuzzy sets. The fuzzy sets are described by membership functions that are specified by the user. The mapping of various volume attributes to a given visual style is achieved with fuzzy logic arithmetics. Fuzzy logic rules specify the mapping with the natural language of the application domain.

The objective of the semantic layers concept is a meaningful mapping from given volume attributes to given visual styles for the purpose of illustration. We identified two major challenges to make our novel concept applicable. On the one hand the mapping has to be bijective in order to be an adequate replacement of the well established color transfer functions. On the other hand

it is desirable to achieve a semantic mapping by the use of semantically meaningful values.

Bijjective Mapping: A bijective mapping ensures that the generated image can be interpreted by the viewer. For the purpose of visualization it is necessary that the meaning of the image can be resolved. The viewer of the image has to be provided with details about the mapping procedure in order to resolve the meaning of the image. A common example for a bijective mapping is a visualization with color encoding. The legend accompanying the visualization ensures that the image can be resolved and has a meaningful interpretation in the underlying parameter domain.

Volume rendering with semi-transparent regions in the volume cannot guarantee to provide bijective visualizations. The compositing of the semi-transparent colors introduces ambiguities. However, volume rendering does potentially convey more information about the data than pure iso-surface renderings. Because of the semi-transparency inner structures are unveiled and can be explored. The ambiguities introduced by the compositing function can be resolved if the used colors are chosen carefully. Animation or interactive exploration provides further help to resolve the ambiguities.

Semantic Mapping: The specification of the semantic mapping from volume attributes to visual styles in our approach is done with semantic values. Measured and simulated data have usually several meaningful intervals that are relevant to the user. For example a PET scan of a brain measures brain activity. It shows homogeneous regions of activity in the brain that are labeled by experts with semantic values such as *low activity* or *high activity*. Diffusion MRI data provides information about the healthiness of tissue regions and is classified by experts with semantic values like *healthy*, *diseased*, or *necrotic*. Medical CT data encode the measured density values in Hounsfield units. Specific intervals of the Hounsfield scale refer to different tissue types like air, soft tissue, bone, contrast enhanced vessels, etc.

There are many more examples of domains where semantic parameters exist. Common visualization techniques (especially direct volume rendering with transfer functions) do typically not make use of these parameters.

Common direct volume rendering techniques also do not make use of semantics for the description of visual styles. Although several semantic parameters exist and are used naturally by illustrators. For example the descriptions of *shading*, *tone*, *rendering style*, *saturation*, *texture*, etc. are done with semantic values. We propose to use these parameters in the specification of the visualization mapping as well.

The remainder of this paper is structured as follows: In Section 2 we review related work. In Section 3 the novel concept of semantic layers for illustrative volume rendering is described in detail. We give an overview of the concept and illustrate it with a simple example. In Section 4 we give details about the implementation of our prototype system. In Section 5 we show exemplary results of our system for different potential application areas.

2 Related Work

We divide the related work into three different categories. We review the related work in illustrative visualization in Section 2.1. In Section 2.2 we compare our work to other multi-dimensional visualization techniques that map multiple volume attributes to multiple visual styles. In Section 2.3 we describe related work in the area of medical visualization that deals with the semantic classification of volume data.

2.1 Illustrative Visualization

Many previous approaches for the selective application of visual styles have been presented. Seligmann and Feiner [13] present a system for intent-based illustrations. They use design rules to achieve the intended visualization of geometric objects. Svakhine et al. [16] present the idea of illustration motifs. They generate illustrative visualizations guided by a specific motif like the purpose or the targeted audience of the illustration. The level of expertise of the viewer serves as input to adjust the automatically generated illustration. Yuan and Chen [20] present a method to enhance volume rendering with different styles of non-photorealistic surface rendering techniques. Hauser et al. [4] introduce two-level volume rendering that allows the selective combination of multiple rendering methods for different regions. The selective application of specific styles based on volume attributes was also used in other previous work [1, 9]. Our approach also provides the possibility to selectively show features of interest with different visual styles. However the focus of our work lies on the semantic mapping of volumetric attributes to visual styles and a uniform approach for the specification of different styles.

Sloan et al. [14] present a technique to render predefined artistic styles. Their method allows the specification and use of very different artistic styles in a uniform framework. We adapted their technique to serve as a basis for the parameterized representation of different styles.

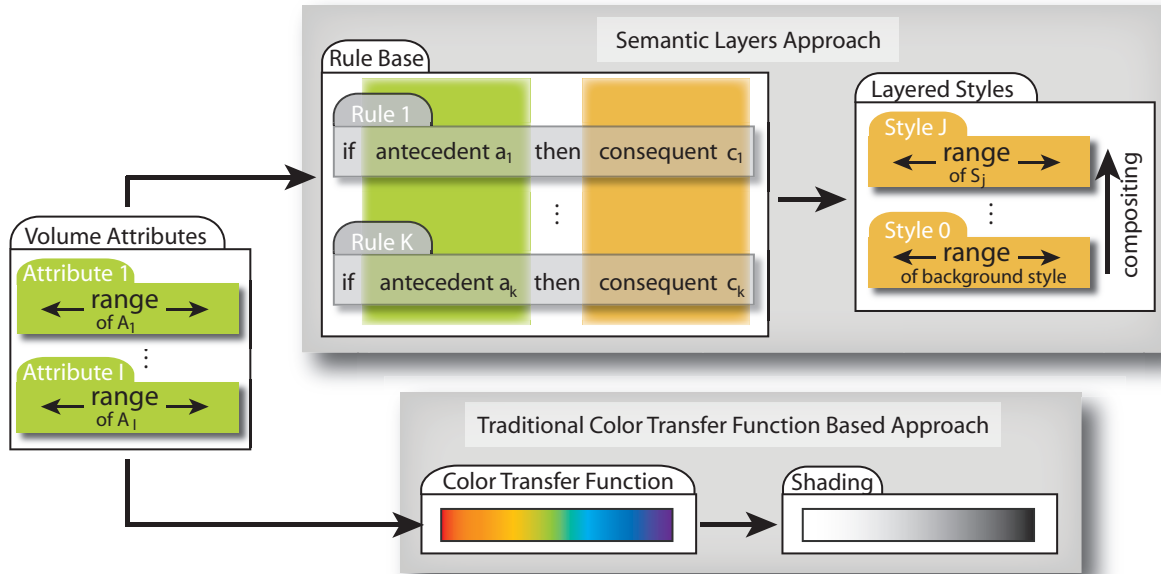


Figure 1: Comparison of the traditional color transfer function based shading and the novel semantic layer approach: In the traditional pipeline multiple volume attributes are used to look up color in the color transfer function. The color is commonly shaded afterwards. In the semantic layers pipeline the volume attributes are described by semantic values. Rules are specified using the semantic values of the volume attributes in the antecedent part of the rule. The rules map the antecedents to semantic values for visual styles specified in the consequent part. All rules affecting one visual style are evaluated to determine the numeric values for the style. The visual styles are layered on top of each other and composited from the background to the top most style.

2.2 Multi-Dimensional Volume Visualization

Kniss et al. [7] present an approach for multi-dimensional transfer functions. They also employ this technique to quantify statistical measures of multiple fuzzy segmentation volumes [8]. Hladuvka et al. [5] as well as Kindlmann et al. [6] used multi-dimensional transfer functions based on the density and a curvature measure. In the work of McCormick et al. [10] as well as Stockinger et al. [15] systems are presented that allow a formulation of the visualization mapping as mathematical expressions. Our approach hides the complexity of mathematical formulations by the use of fuzzy logic. In the work of Doleisch et al. [3] a system for the interactive exploration of complex data is presented. Woodring and Shen [18] use set and numerical operators to visualize and compare multi-variate and time-varying data. Sato et al. [12] use rules to identify tissue structures in multi-modal data.

Our approach is similar to these approaches as it uses a rule based specification of features in the data. However the specification of the mapping to visual attributes in our system is done using fuzzy logic. Fuzzy logic allows

the specification of rules mapping semantic values of volume attributes to semantic values of visual styles.

2.3 Medical Visualization

We see illustrative visualization for operation planning, illustration for patient briefing, and multi-modal visualization as potential applications of our approach. Others have presented approaches related to our work.

Tappenbeck et al. [17] as well as Zhou et al. [21] modify the appearance of volumetric structures based on the distance to a predefined region. Our system provides a similar functionality. The user can define a style that is used in dependence of the distance to a given region. The chosen style can be altered and modified interactively. Rezk-Salama et al. [11] present a high-level user interface for the specification of transfer functions with semantics. We follow up on the idea of specifying a mapping from volume attributes to a visual style by semantically meaningful parameters. We do however not employ a traditional color transfer function to achieve this mapping. The derivation of semantics for parameters in our system can be done once by experts whereas

in the work of Rezk-Salama et al. [11] it is done automatically using principal component analysis on a set of predefined transfer functions.

3 Overview of the Semantic Layers Concept

A comparison between the traditional color transfer function based approach and the semantic layers approach is shown in Figure 1. The traditional approach takes multiple volume attributes as input and derives a basic color via the color transfer function for each sample position. The color is usually shaded and used for compositing to get the final color of the pixel. The semantic layers approach also takes multiple volume attributes as input but in contrast evaluates a set of rules to determine the different visual styles applied to the current sample. The styles are composited from the background style to the top most layer to determine the color of the current sample. The color of the sample is used for compositing to determine the color of the pixel.

To achieve the linguistic mapping from values of volume attributes to values of visual styles we use semantic values. The semantic values are described in Section 3.1. The mapping from semantic values of volume attributes to semantic values of visual styles is specified with rules. The rule base described in Section 3.2 computes the mapping onto visual styles. The evaluation of the rules results in numeric values for each visual style. Each style is parameterized and contributes according to the numeric value to the final visual appearance of one sample. The styles are described in Section 3.3. The final visual appearance of a sample is determined by the composition of the layered visual styles. The concept of layered styles is described in Section 3.4.

3.1 Semantic Values

A semantic value is a linguistic description of a value of a volume attribute or a visual style. It is defined as a fuzzy set given by its membership function.

Figure 2 shows an example of the specification of semantic values for the volume attributes *density* and *curvature*. The attribute *density* is described with the semantic values *low* and *high*. The attribute *curvature* is described with the semantic value *close-to-zero*. The semantic values are specified with a membership function. In Figure 2 only simple examples are shown to illustrate the concept of membership functions and semantic value definition. The membership functions shown

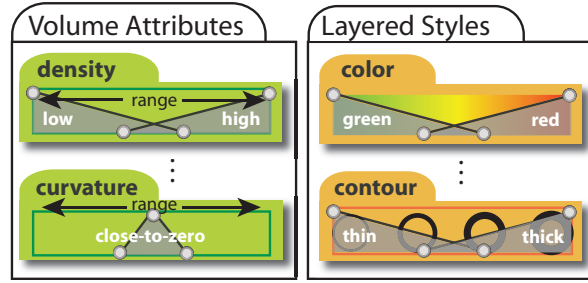


Figure 2: Exemplary specification of semantic values. Simple semantic values for the volume attributes *density*, and *curvature* are shown as well as for the visual styles *color* and *contour*.

in Figure 2 are just straight lines. In practice membership functions can be more complex and attributes are described with more semantic values. For example the volume attribute *density* could have the values *very low*, *low*, *middle* and *high* referring to meaningful types of tissue like *air*, *soft tissue*, *bone*, *contrast enhanced vessels*, etc.

The linguistic values for visual styles are also defined using membership functions. In Figure 2 simple examples for the visual styles *color* and *contour* are shown. For the visual style *color* the definition of the semantic values *green* and *red* are shown. The visual style *contour* is described by the semantic values *thin* and *thick*. *Shading* is another examples of a visual styles that could be described with the semantic values *none*, *soft*, *phongish* and *cartoonish*. The style *desaturation* could range from the value *no* to the semantic value *full*.

3.2 Rule Base

The central component of our system is the rule base that gives a linguistic description of the desired mapping. A rule states the premise in the antecedent part and the conclusion in the consequent part. The premise is a logical combination of semantic values of volume attributes. The conclusion is a list of styles that are affected by the rule. A rule could for example state *if density is middle and brain activity is very high then color-coding is red*. The result of the evaluation of one rule is another fuzzy set quantifying the membership to the antecedent. Antecedents of all rules that have implications on the appearance of one style are aggregated and defuzzified. The result of the defuzzification is a style volume describing the value of the visual style at each voxel position. The implication, aggregation and defuzzification can for the moment be seen as black boxes

that map the antecedents of the rules to values for the visual styles. In Section 4 we give details about the used fuzzy logic methods.

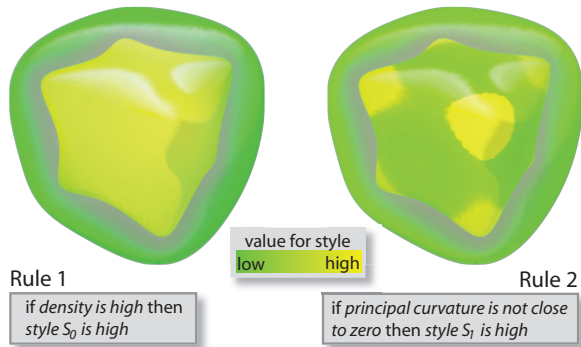


Figure 3: Style volumes: Two iso-surfaces of the cube dataset for each style volume are rendered. A color-coding was used to encode the numeric values for each style. The left style volume shows the result of a density based rule. The right image shows the result of a curvature based rule. Green color means low values for the given style, yellow color means high values for the given style.

In Figure 3 two style volumes are shown. A color coding was applied on two nested iso-surfaces of the cube dataset to encode the value for the style. Green means low value for the style and yellow means high value for the style. The left style volume is the result of the rule *if density is high then style S_0 is high*. The right style volume is the result of the rule *if curvature is not close-to-zero then style S_1 is high*.

3.3 Styles

Each style volume specifies the value for a style at each position. To apply a style to a sample the style needs to be parameterized. The parameterization of a style ensures a continuous application of one style.

Parameterization: A common example of a parameterized style is a color scale. In traditional illustration however a greater variety of gradually varying styles exists. It is desirable that the application of one style volume results in a gradual application of the style. Parameterized styles are needed to achieve this gradual transition. An example of parameterized styles can be seen in Figure 4. The spheres in Figure 4 are drawn manually. The glossyness varies over the vertical axis from *none*- over *weak*- to *strong*- glossyness. The style contours varies along the horizontal axis from *none* over *thin* to *thick*.

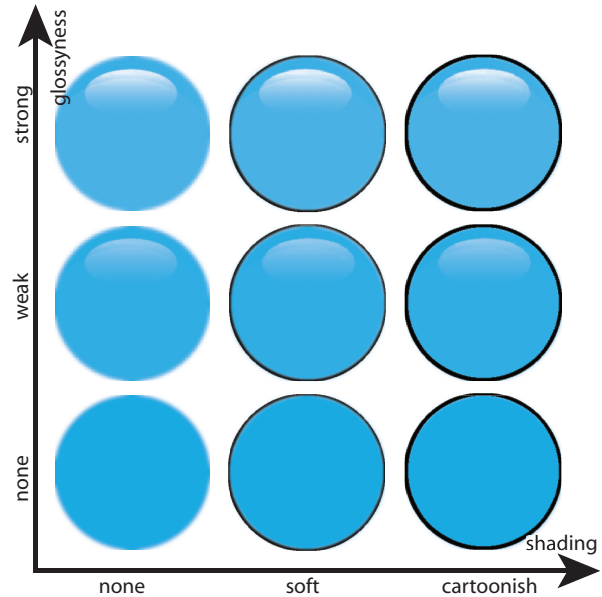


Figure 4: Combination of three parameterized orthogonal styles: The style *color* is constantly set to blue. The styles *contours*, and *glossyness* gradually vary over their specified domain.

In practice all three axis (the color, the shading, and the contours) vary continuously allowing a gradual change in each dimension. In Figure 4 the style *color* is constantly set to *blue*.

Orthogonality: We propose to use orthogonal styles to achieve meaningful mappings. Potentially the combination of different styles leads to ambiguities. It is desirable to achieve mappings of multiple styles that can be combined without leading to ambiguities. We define orthogonal styles as a set of styles that do not infer with each other. The concept of orthogonal styles can be seen in Figure 4. The example illustrates the combination of the three orthogonal styles *color*, *contours* and *glossyness*. Nine exemplary combinations of the gradually varying styles *contours* and *glossyness* are shown. The *color*-dimension is not shown in Figure 4, however it is orthogonal to the shown dimensions *contours* and *glossyness*.

The orthogonality of visual styles does in general not solve the problem of ambiguities in volume renderings using semi-transparency. The use of orthogonal styles is (not sufficient, but) necessary to achieve meaningful and bijective volume visualizations.

3.4 Layered Styles

From an illustration point of view it is desirable to apply the styles incrementally and selectively.

Incremental Application of Styles: Our approach for the incremental application of multiple styles resembles the work-flow of a traditional illustrator. Traditional illustrations are drawn in layers. Each layer is applied on top of another. First a basic style is applied to a specific region. On top of the basic style shading is applied, contours are added and specular highlights are drawn.

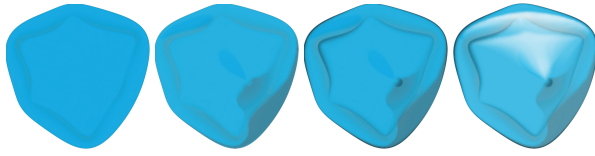


Figure 5: Incremental application of styles: Two nested iso-surfaces of the cube dataset are shown. From left to right the following layers are incrementally applied: background style, subtle cartoonish shading, contours, and glossy highlights.

Figure 5 shows the incremental application of layered styles. Two nested iso-surfaces of the cube dataset are rendered using the same styles as shown in Figure 4. During rendering we automatically apply the style of the manually drawn spheres to achieve the same appearance of the rendered object. Details about the application of pre-defined styles during rendering are given in Section 4. The different layers respectively the different styles are composited on top of each other, starting with a background style applied to all regions. The leftmost image in Figure 5 was rendered using the unshaded background style. In this case the background style is specified by the sphere in the lower left corner of Figure 4. The second image in Figure 5 is drawn applying a subtle cartoon shading style. The cartoon shading style is not shown in Figure 4. The third image in Figure 5 is rendered adding a contour. The contour style is given by the sphere in the lower right corner in Figure 4. Finally the layer that resembles a glossy highlight is applied. The result is shown in the rightmost image of Figure 5. The upper left image in Figure 4 was used to describe the glossy style. Each style can be applied gradually, however in Figure 5 all styles are fully applied to show the effect of incremental application of several layered styles.

Selective Application of Styles: Illustrators are taught to avoid mixtures of too many styles. A selective application of styles to specific regions can aid to differentiate the individual regions. Each semantic layer defines

one style according to a set of rules. The rules allow for a differentiation of regions and for a selective application of the styles. Each layer in the hierarchy is applied in regions according to its style volume resulting in a selective application of the style. Note that a completely opaque style overdraws all styles lower in the hierarchy. To achieve meaningful illustrations it is important to choose the styles carefully.

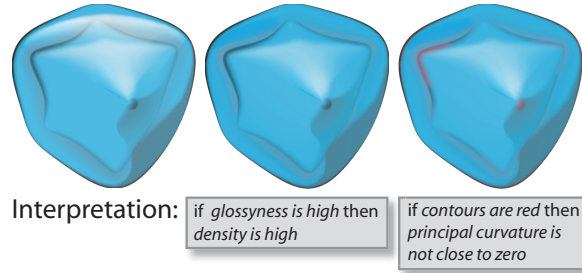


Figure 6: Selective application of styles: The left image shows two iso-surfaces of the cube dataset. All styles are fully applied. In the middle image the style *glossyness* is determined by a density based rule. Glossy highlights are only drawn in regions of high density. In the right image the style *contours* is determined by a curvature based rule. The contours are drawn in red in regions of high absolute principle curvature. The rules specifying the mapping can be read backward as an interpretation of the images.

Figure 6 shows the selective application of visual styles. The leftmost image shows two iso-surfaces of the cube dataset. All styles are fully applied. In the middle image of Figure 6 the rule *if density is high then glossyness is high* is applied. The style *glossyness* is affected by this rule and applied only to regions of high density. The right image of Figure 6 demonstrates the selective application of a style following a curvature based rule. The contours are drawn in red in regions with the first principal curvature not close to zero. The rules that lead to the selective application of the visual styles can be read in the opposite direction to resolve the image and to map back the styles to the original semantic parameters. The image in the middle of Figure 6 can be interpreted with the sentence *if glossyness is high then density is high* and the image on the right can be interpreted with the sentence *if contours are red then curvature is not close-to-zero*. The augmentation of the final result with the inverted rules ensures the bijectivity of the achieved visualization.

In this Section we introduced the concept of semantic layers for illustrative volume rendering. Semantic

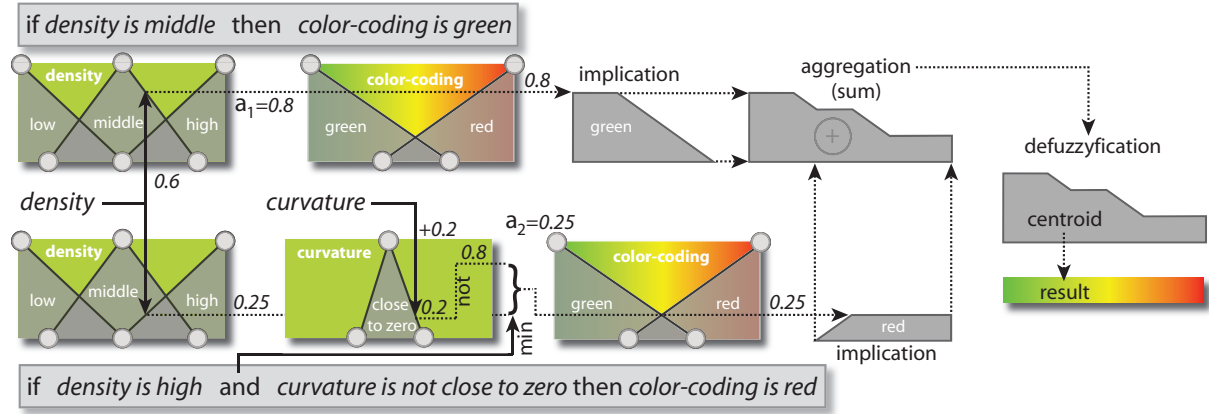


Figure 7: Overview of the fuzzy logic system: The evaluation of two simple rules influencing the same style is illustrated. The density and curvature based rules are evaluated. The implication truncates the membership function of the semantic values specified in the consequent part of the rules. The membership functions after the implication are aggregated and defuzzified. The result is a numeric value for the style *color-coding*

layers provide a methodology for a linguistically specified mapping from volumetric attributes to visual styles. Note that each of the described components can be exchanged independently. For example the rule base for a medical application can be reused for illustration purposes. Also a set of predefined styles can be reused. For example a set of styles for illustrative visualization can be reused in different illustration scenarios.

4 Implementation

In this Section we describe the implementation of the concepts introduced in Section 3. We chose fuzzy logic to achieve the definition of semantic values for volume attributes as well as for styles. Each semantic value is specified by a membership function of a fuzzy set. The actual mapping of the values of volume attributes to values of visual styles is specified using fuzzy rules. The evaluation of the rule base is done using fuzzy logic operations. The whole fuzzy logic inference process used in our system is described in Section 4.1. An more general introduction to fuzzy logic can be found in [19]. The result of the fuzzy logic inference are style volumes that describe the value of each voxel for a given style. Each style provides a parameterization. The parameterized definition of styles is achieved in our system by style descriptors. For the final rendering the styles are applied incrementally. In our implementation the incremental application of the layered styles is done during rendering. We introduce style descriptors in Section 4.2

and give details about the implemented direct volume rendering algorithm.

4.1 Fuzzy Logic Inference

In Figure 7 an overview of the fuzzy logic inference process for one visual style is shown. The inference process evaluates the value for a given style at each voxel position. In Figure 7 two rules affect the visual style *color coding*. The fuzzy logic inference is illustrated for a voxel with density value 0.6 and a curvature value 0.2. We use the example of Figure 7 for the description of all fuzzy logic inference steps: membership function evaluation, fuzzy set operations, implication, aggregation and defuzzification.

Membership Functions and Fuzzy Set Operations:

The base for the semantic description of parameters are fuzzy sets. Fuzzy sets are specified via membership functions. In Figure 7 the definition of the piecewise linear membership functions *low density*, *middle density*, *high density*, *close to zero curvature*, *green color-coding*, and *red color-coding* can be seen. The membership for a given value of a volume attribute to a fuzzy set is described by the membership function. In Figure 7 the evaluation of the membership function *middle density* results in 0.8 for a density value 0.6, whereas the membership function *high density* results in 0.25. The membership function *close-to-zero curvature* results in 0.2 for the curvature value 0.2 in the example of Figure 7.

Logical operations can be applied on fuzzy sets. The unary negation operator (i.e. *not*) results in the calculation of one minus the membership function. In the example of Figure 7 the negation is applied to the fuzzy set *curvature is close to zero*. The curvature value of 0.2 therefore results in a membership value of 0.8 for the fuzzy set *curvature is not close to zero*.

The combination of multiple fuzzy sets is done using one of the binary operators *and*, and *or*. The *and* operator results in a minimum operation for the membership functions of the operands. The *or* operator results in a maximum operation for the membership functions of the operands. In the example of Figure 7 the *and* operator is applied to the fuzzy sets *density is high* and *curvature is not close-to-zero*. The result of this operation is 0.25 as this is the minimum of the chosen values in the example.

Implication, Aggregation and Defuzzification: The fuzzy implication results in a fuzzy set for a visual style. The resulting membership function is the minimum of the membership function describing the semantic value of the style and the value of the antecedent. In Figure 7 the result of the implication of the upper rule is the truncated membership function for the semantic value *green color-coding*. The truncated membership function describes a fuzzy set. The lower rule in Figure 7 results in the truncation of the membership variable for the semantic value *red color-coding*. The implicated fuzzy sets *green color-coding* and *red color-coding* are combined using the aggregation function.

During aggregation multiple fuzzy sets influencing one visual style are combined. In Figure 7 the aggregation of the two fuzzy sets describing the *color-coding* are aggregated resulting in the sum of the two membership functions.

The input for the defuzzification is the fuzzy set of the aggregation. The output of the defuzzification is a numeric value for a given visual style. Many common defuzzification methods exist. We chose in our implementation the centroid method. In Figure 7 the defuzzification results in a numeric value for the style *color-coding*. The result of the example in Figure 7 for the style *color-coding* is a yellowish green. We chose the centroid method because its output (unlike for other popular defuzzification methods like the smallest-, the middle-, or the largest of maximum) varies continuously if the input varies continuously.

In our implementation we first compute the antecedents of all specified rules and store them in antecedent volumes. Whenever a style volume has to be evaluated all previously computed antecedent volumes that affect the

style are used for implication, aggregation and defuzzification. The implication, aggregation and defuzzification are done in one calculation per voxel position. For defuzzification we want to find the centroid of a function $f(x)$ resulting from the aggregation. The centroid c_f of a function $f(x)$ is given by the equation:

$$c_f = \frac{\int x f(x) dx}{\int f(x) dx} \quad (1)$$

Let the membership function for a semantic value s_i be $m_i(x)$. The membership function for the semantic value s_i after the implication of a given rule r_j is then given by the equation

$$m_i'(x, a_j) = \min(a_j, m_i(x)) \quad (2)$$

where a_j is the antecedent value of the rule r_j . The aggregated membership function $f(x)$ is then given by

$$f(x) = \sum m_i'(x, a_j) \quad (3)$$

The centroid of the aggregated function can then be calculated by substituting Equation 3 in Equation 1:

$$c_f = \frac{\int x \sum m_i'(x, a_j) dx}{\int \sum m_i'(x, a_j) dx} \quad (4)$$

We can rewrite Equation 4 as follows:

$$c_f = \frac{\sum \int x m_i'(x, a_j) dx}{\sum \int m_i'(x, a_j) dx} \quad (5)$$

In Equation 5 it can be seen, that the summands in the nominator as well as in the denominator do solely depend on the a_j . We precompute the summands of the nominator as well as of the denominator and store them in a lookup table. During evaluation the a_j are used as index for the lookup tables. The sum and the maximum are the most common aggregation functions. Using the maximum as aggregation function does however not allow the precomputation of the summands. The defuzzification is computationally much more expensive using the maximum for the aggregation. We therefore recommend to use the sum as aggregation function.

In our implementation the antecedents are computed and cached. If semantic values of styles or the consequent part of a rule changes the cached antecedents are reused. The time for the evaluation of a rule depends very much on the complexity of the rule and the dataset size. The time for the evaluation of the rules shown in the examples and results take a few seconds with our current unoptimized implementation.

4.2 Rendering

The outcome of the fuzzy logic inference is a style volume for each specified style. During rendering the style volumes are used to determine the parameter for the application of each style. To achieve a parameterization of different styles with one uniform framework we adapt the style transfer functions approach, introduced by Bruckner and Gröller [2]. Style transfer functions were introduced to apply artistic styles to volume rendering. A style transfer function, like a common one dimensional transfer function, defines an opacity for each density value. Instead of defining a color for each density value it defines a style. The style is given by an image of a sphere. Each sphere might be manually shaded by an artist resembling the desired artistic style. The image of the sphere implicitly defines a color value for each normal direction in eye-space. The pre-shaded sphere can therefore be used as a look-up table during rendering. The gradient in eye-space is used to index the look-up table. The color value of a sample during volume rendering is simply defined by the color value of the sphere in the direction of the gradient. We adapted this approach to parameterize different styles that can be applied incrementally and selectively. In Figure 8 a screenshot of the style descriptor widget can be seen. The small gray triangles indicate the positions of the user specified pre-shaded spheres. Each row is a parameterization of a style. The spheres displayed by the widget in one row are examples for different parameters interpolating the user specified spheres. The top row shows an artistic surface style ranging from matt gray over shiny black to glossy metallic. The second row shows a tissue type style often found in medical illustration. The tissue style in Figure 8 shows examples for soft tissue, skin, and bone. The third row describes different contour styles ranging from transparent, over black to red contour. The last two rows in Figure 8 show examples for shading styles and glossy highlight styles. Style descriptors provide a uniform framework for the parametrization of styles. The user defines a few spheres that are used to parameterize the styles. Layered style descriptors can be combined during rendering.

Our rendering approach is a GPU based ray-casting algorithm. We use one color channel per style. During the evaluation of each style we do one texture lookup per sample. The styles are composited from the background style to the top most layer. Our approach runs with interactive frame rates on a GeForce 8800GTX graphics card. We measured the average frame rates for the datasets shown in the result images with a sample distance of 0.5 and a viewport size of 800×600 . The average frame rate for the dataset (of size $256 \times 256 \times 166$)

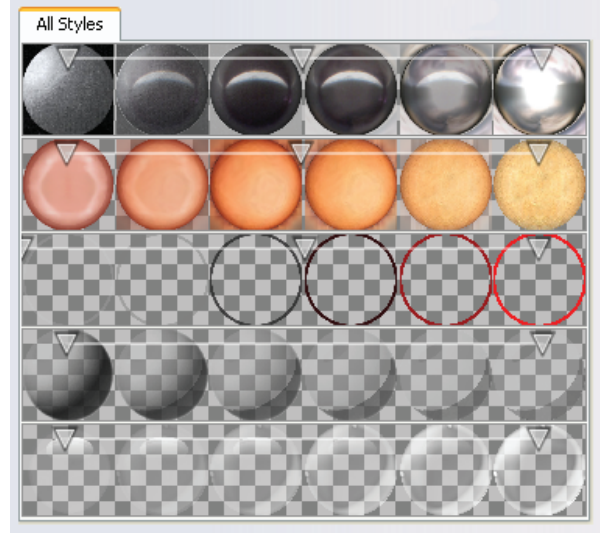


Figure 8: Screenshot of the style descriptor widget. Each row shows a parameterized style. From the top to the bottom row the following styles are shown: *artistic surface style*, *tissue style*, *contour style*, *shading style*, and *glossy highlight style*.

shown in Figure 10 using the same styles and opacity transfer function was 9fps. For the engine block dataset (of size $256 \times 256 \times 256$) shown in Figure 9 we achieved an average of 7fps. However using a less transparent opacity transfer function for the same settings resulted in an average frame rate of 20fps. For the monkey atlas dataset (of size $256 \times 256 \times 62$) shown in Figure 11 average frame rates from 9 to 16fps were achieved depending on the used opacity transfer function.

5 Results

We show result images demonstrating the flexibility of our approach. The results show the capability of the system to deal with very different volume attributes as input. Our approach uses the method of abstraction of the data. The introduction of semantic layers allows for a great flexibility for the used input data and for the desired visualization method.

In Figure 9 three renderings of the engine block dataset are shown. The chosen opacity transfer function and the chosen viewpoint allow the view on many details of the engine block dataset. On the left image in Figure 9 cartoonish shading is used to make surfaces of the engine block with different normals clearly distinguishable. In the middle image of Figure 9 the cartoonish

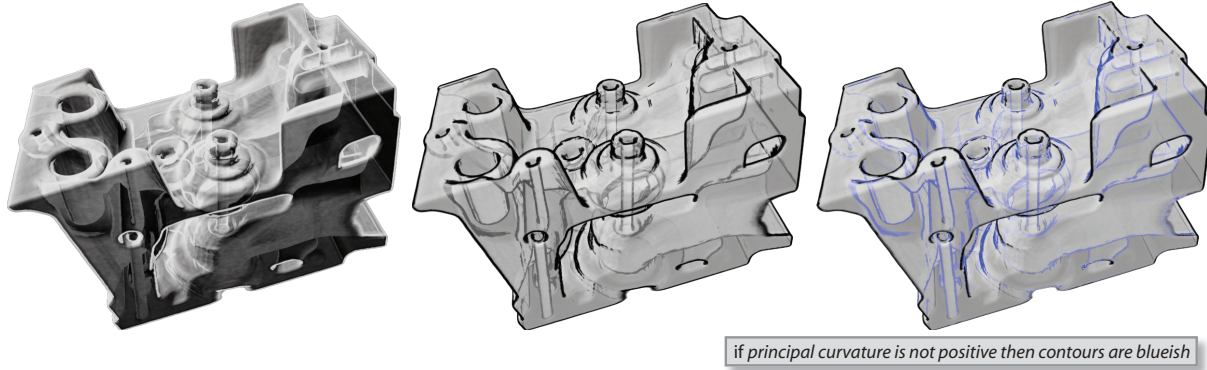


Figure 9: Incremental and selective application of styles on the engine block dataset. The left image shows a cartoonish shading effect. The middle image shows a reduced cartoonish shading effect and enhances the edges of the engine block dataset with contours. The right image selectively colors the contours. Convex regions are shown in black, whereas concave regions are shown with blueish contours.

shading style is reduced and contours are added. The contours accentuate the individual parts of the engine block. In the right image the rule *if curvature is not positive then contours are blueish* is used. The curvature based selective application of the blue contours makes it possible to distinguish between convex and concave contours in the image.

In Figure 10 an illustrative rendering of a CTA scan of a human head is shown. The left image of Figure 10 uses two different tissue styles for bone and skin. In the right image of Figure 10 black contours are selectively applied on the bone using the simple rule *if density is high then contours are thick*. The major vessels of this dataset are segmented. We used a distance transform of the segmented data to apply a distance based rule. The rules *if distance to vessels is very low then color is red* and *if distance to vessels is low then color is yellow* were used. We applied the style *color* on top of the other styles. Regions of low distance to the vessels can be seen in yellow and red in Figure 10.

In Figure 11 three images of the monkey atlas dataset are shown. The monkey atlas dataset contains a registered CT and PET scan of a monkey head. The CT data was used for the rendering. The PET data was used for the application of the rule *if brain activity is high contours are red*. In all images the red contours are shown. In the top row of Figure 11 two images are shown using a semi-transparent opacity transfer function. The lower image in Figure 11 uses a more opaque transfer function and a more illustrative style.

6 Conclusion and Future Work

In this paper we presented a novel approach for the specification of the mapping from volume attributes to visual styles. We enable a linguistic description for the specification of the desired visualization by using semantic values for volume attributes and for visual styles. The novel methodology describes *how* different features in the data are rendered. We believe that this paper opens up a new research direction dealing with the semantic specification of visualizations.

The next step will be an adaption of our system to incorporate view dependent attributes like *distance to the mouse cursor*, *depth* (i.e. distance to the image plane), *viewing direction*, etc. This will allow the selective application of styles driven by view dependent attributes. An adaption of the algorithm is necessary to make view dependent rule evaluation possible on the graphics card. However the used separable aggregation method described in Section 4 makes it possible to evaluate the rules partially on the CPU and partially on the GPU. We expect the view dependent attributes to be a powerful tool for interactive illustration.

Further we want to investigate the automatic (or semi-automatic) derivation of membership functions. We experienced, that the membership function specification for different applications usually follows a similar strategy. The membership functions are usually specified over the range of values that occur within the data. We plan to use simple heuristics that automatically provide an initial specification of membership functions. The automatic generation of membership functions will then be triggered by the keyword *relatively* in a rule. For example a rule stating *if curvature is relatively low ...* will

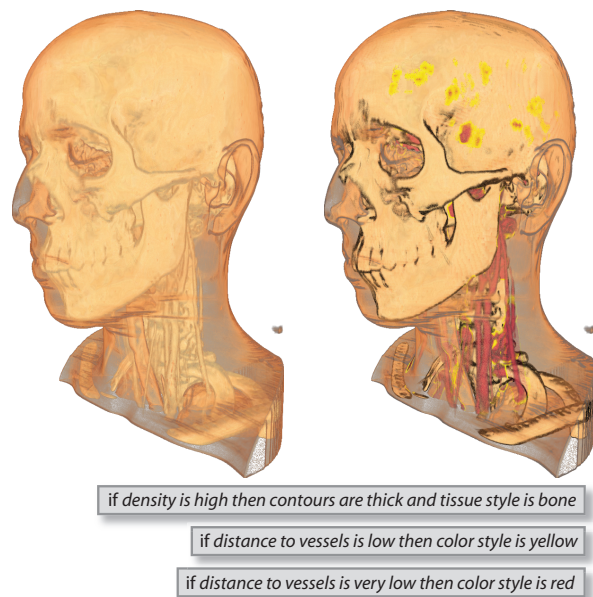


Figure 10: Illustrative rendering of a human head. Left image shows the use of two different tissue styles. The right image shows the selective application of contours to regions of high density. Further a distance based color style was overlayed. Yellow encodes low distance to vessels and red encodes very low distance to vessels.

be translated into a membership function having a peak at the minimum value of the curvature and will range to include a pre-specified percentile of the data.

Image manipulation programs like Adobe Photoshop or Gimp use the concept of layered images. We analogously use the concept of volumetric layers. Currently the composition of layers is done using the default composition method. However, following the analogy of image manipulation programs we could allow an interactive specification of other composition operations like difference, inversion, multiplication, etc.

Acknowledgements

The work presented in this publication is carried out as part of the **exvisation** project (www.cg.tuwien.ac.at/research/vis/exvisation) supported by the Austrian Science Fund (FWF) grant no. P18322. We also want to acknowledge the Laboratory of Neuro Imaging, UCLA, USA for the monkey atlas dataset and for kindly providing details about the PET data. The engine block dataset is courtesy of General Electric, USA.

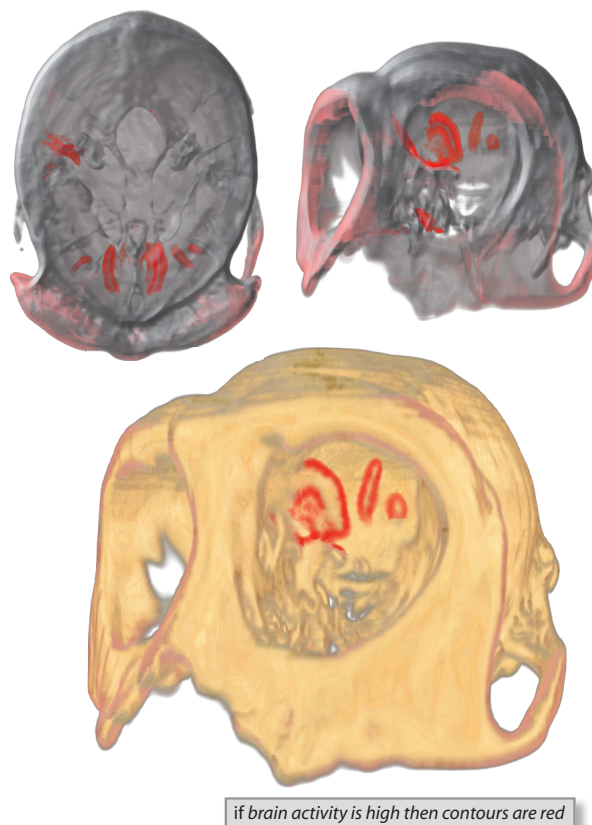


Figure 11: Illustration of the multi-modal monkey atlas dataset. The top images show two different views of the dataset using a transparent opacity transfer function. The lower image shows an illustrative rendering from the same view as the top right image. Red contours are drawn in all three images in region of high brain activity.

References

- [1] S. Bruckner and M. E. Gröller. VolumeShop: An interactive system for direct volume illustration. In *Proceedings of IEEE Visualization 2005*, pages 671–678, 2005.
- [2] S. Bruckner and M. E. Gröller. Style transfer functions for illustrative volume rendering, submitted for publication. Technical Report TR-186-2-07-02, 2007.
- [3] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the Symposium on Data Visualization 2003*, pages 239–248, 2003.

- [4] H. Hauser, L. Mroz, G.-I. Bischi, and M. E. Gröller. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):242–252, 2001.
- [5] J. Hladůvka, A. König, and M. E. Gröller. Curvature-based transfer functions for direct volume rendering. In *Proceedings of the Spring Conference on Computer Graphics 2000*, pages 58–65, 2000.
- [6] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, 2003.
- [7] J. Kniss, G. Kindlmann, and C. Hansen. Multi-dimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.
- [8] J. Kniss, R. V. Uitert, A. Stephens, G.-S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. In *Proceedings IEEE Visualization 2005*, pages 287–294, 2005.
- [9] E. B. Lum and K.-L. Ma. Lighting transfer functions using gradient aligned sampling. In *Proceedings of IEEE Visualization 2004*, pages 289–296, 2004.
- [10] P. McCormick, J. Inman, J. Ahrens, C. Hansen, and G. Roth. Scout: a hardware-accelerated system for quantitatively driven visualization and analysis. In *Proceedings of IEEE Visualization 2004*, pages 171–178, 2004.
- [11] C. Rezk-Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.
- [12] Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3d local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.
- [13] D. D. Seligmann and S. K. Feiner. Automated generation of intent-based 3D illustrations. In *Proceedings of ACM Siggraph 1991*, pages 123–132, 1991.
- [14] P.-P. Sloan, W. Martin, A. Gooch, and B. Gooch. The lit sphere: A model for capturing NPR shading from art. In *Proceedings of Graphics Interface 2001*, pages 143–150, 2001.
- [15] K. Stockinger, J. Shalf, W. Bethel, and K. Wu. Query-driven visualization of large data sets. In *Proceedings of IEEE Visualization 2005*, pages 167–174, 2005.
- [16] N. Svakhine, D. S. Ebert, and D. Stredney. Illustration motifs for effective medical volume illustration. *IEEE Computer Graphics and Applications*, 25(3):31–39, 2005.
- [17] A. Tappenbeck, B. Preim, and V. Dicken. Distance-based transfer function design: Specification methods and applications. In *SimVis*, pages 259–274, 2006.
- [18] J. Woodring and H.-W. Shen. Multi-variate, time varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):909–916, 2006.
- [19] R. R. Yager and L. A. Zadeh, editors. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, volume 165 of *International Series in Engineering and Computer Science*. Springer, 1992.
- [20] X. Yuan and B. Chen. Illustrating surfaces in volume. In *Proceedings of Joint IEEE/EG Symposium on Visualization 2004*, pages 9–16, 2004.
- [21] J. Zhou, A. Döring, and K. D. Tönnies. Distance based enhancement for focal region based volume rendering. In *Proceedings of Bildverarbeitung für die Medizin 2004*, pages 199–203, 2004.