2D Histogram based volume visualization: Combining intensity and size of anatomical structures

Article *in* International Journal of Computer Assisted Radiology and Surgery · November 2010

DOI: 10.1007/s11548-010-0480-1 · Source: PubMed

3 authors, including:

Stefan Wesarg
Fraunhofer Institute for Computer Graphics Research IGD
144 PUBLICATIONS **765** CITATIONS
SEE PROFILE

M. Fawad Khan
Goethe-Universität Frankfurt am Main
66 PUBLICATIONS **836** CITATIONS
SEE PROFILE

Int J CARS (2010) 5:655–666
DOI 10.1007/s11548-010-0480-1

ORIGINAL ARTICLE

2D Histogram based volume visualization: combining intensity and size of anatomical structures

S. Wesarg · M. Kirschner · M. F. Khan

Received: 17 December 2009 / Accepted: 27 April 2010 / Published online: 30 May 2010

Abstract
Purpose Surgical planning requires 3D volume visualizations based on transfer functions (TF) that assign optical properties to volumetric image data. Two-dimensional TFs and 2D histograms may be employed to improve overall performance.
Methods Anatomical structures were used for 2D TF definition in an algorithm that computes a new structure-size image from the original data set. The original image and structure-size data sets were used to generate a structure-size enhanced (SSE) histogram. Alternatively, the gradient magnitude could be used as second property for 2D TF definition. Both types of 2D TFs were generated and compared using subjective evaluation of anatomic feature conspicuity.
Results Experiments with several medical image data sets provided SSE histograms that were judged subjectively to be more intuitive and better discriminated different anatomical structures than gradient magnitude-based 2D histograms.
Conclusions In clinical applications, where the size of anatomical structures is more meaningful than gradient magnitude, the 2D TF can be effective for highlighting anatomical structures in 3D visualizations.

Keywords 3D visualization · Volume rendering · Histogram · Transfer function · CT · MRI

S. Wesarg (✉) · M. Kirschner
Interactive Graphics Systems Group (GRIS),
TU Darmstadt, Fraunhoferstr. 5, 64283 Darmstadt, Germany
e-mail: stefan.wesarg@gris.informatik.tu-darmstadt.de

M. F. Khan
Department of Diagnostic and Interventional Radiology,
Goethe University, Theodor-Stern-Kai 7,
60590 Frankfurt am Main, Germany

Introduction

State-of-the-art imaging modalities used in medicine, such as computed tomography (CT) or magnetic resonance imaging (MRI), provide 3D image data. However, radiologists, who diagnose based on the image data, prefer simple 2D visualizations. They create the corresponding 3D structures *mentally* by connecting the visual information gathered from several 2D slices. This 3D representation, which exists only in the radiologist's mind, cannot be used by a surgeon who has to perform an intervention. At the interface between radiology and surgery, 3D visualizations of medical image data play an increasing role. Important diagnostic information for the surgeon has to be presented in 3D due to the fact that surgeons are accustomed to dealing with 3D objects, namely the anatomical structures they are working on.

Thus, there is a need to provide methods for creating meaningful 3D visualizations of clinical image data. One major problem when generating 3D renderings is occlusion. Important anatomical structures may be invisible because they are hidden behind others. This difficulty can be compensated for—at least to some extent—if transfer functions (TFs) [17] are defined for assigning low opacity values to voxels of a specific intensity. That way, disturbing structures can be masked in order to *look behind them*.

Similarly, TFs for assigning color to the 3D rendering can be defined. These one-dimensional TFs solely take the intensity values of the image data into account. This means that all voxels sharing the same intensity values are rendered in the same way. In order to provide a means to distinguish between those voxels, additional image properties can be taken into account for generating multi-dimensional TFs [10]. One such property is gradient magnitude which can be easily derived from the original image data [9]. In this case, edges which are represented by large gradient

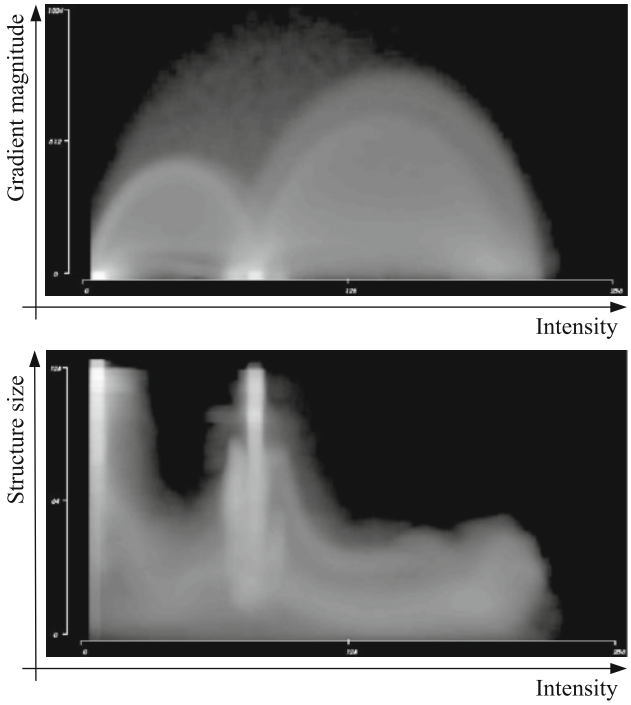

Fig. 1 A conventional IGM histogram (*top*) compared to the new SSE histogram (*bottom*), both computed for a head CT image data set (see Figs. 3, 6). The brightness of the 2D bins corresponds to the bin frequency. The IGM histogram shows the typical arch structures representing boundaries in the image data (see [10])

values can be rendered with different optical properties (opacity, color) than homogeneous image regions that share the same intensity values, but which are characterized by very low gradient values. Hence, organ boundaries can be emphasized by rendering them more opaque than the interior part.

The question that arises is how to select voxels with a specific combination of intensity and gradient magnitude values. For this purpose, 2D histograms can be employed. Typically, the x axis represents intensity, the y axis gradient magnitude, and the brightness of the 2D bin—a coordinate in the 2D histogram—the number of voxels which have that specific combination of intensity and gradient magnitude (see Fig. 1). In the remainder of the paper, we call them *IGM histograms* (intensity + gradient magnitude). Two problems concerning the interaction with such IGM histograms can be identified. Firstly, histograms only represent a statistical distribution of image properties (e.g. intensity) and do not contain any spatial information. That means that the voxels in two adjacent bins in the histogram may be located anywhere in the image data. Secondly, the selection of specific 2D bins in the IGM histogram is not intuitive. This is especially true if the users of such visualization approaches are untrained in the field of image processing and computer graphics. And this is the case for most clinicians.

In this paper, we address both problems. We introduce the *structure size enhanced* (SSE) histogram. Here, gradient magnitude as second property used in the 2D histogram is replaced by the *size of anatomical structures* (see Fig. 1). Thus, an intuitive selection of voxels representing for instance *large structures with medium intensity* is possible. In addition, we add spatial information to the 2D histogram by adopting the idea of *spatialized TFs* [19]. For this, we perform a clustering of the bins in the SSE histogram either by the 3D location of the corresponding voxels or by their distance from one or more user given reference positions.

The structure of the paper is as follows. In "Related work" we review existing approaches related to the generation of TFs for volume rendering. Our methods for the computation of the structure size image, the SSE histogram generation, and the creation of the 3D renderings are presented in detail in "Material and methods". Section "Results" gives examples of visualizations for a wide range of clinical image data and a comparison of our SSE histogram with the IGM histogram. Finally, we discuss our contribution in "Discussion".

Related work

Bajaj et al. [1] introduced the *contour spectrum* which represents several measures for describing isosurfaces and a user interface for interactively setting parameters for the isovalue definition. A similar approach for isosurface visualization was proposed by Pekar et al. [13] with the Laplacian-weighted intensity histogram.

Several early approaches for assigning TFs were discussed in [15]. This comparison showed that a semi-automatic approach combining intensity with first and second order derivatives into the *histogram volume* [9] is the most promising one. By projecting the histogram volume along one of the three axes one obtains a 2D histogram (e.g. intensity + gradient magnitude) which can be used for the definition of an opacity TF emphasizing the boundaries in the data set. Kniss et al. [10] extended this approach to interactive volume rendering based on multi-dimensional TFs focusing on interaction widgets for effectively setting the parameters for opacity and color TFs in the 2D histogram domain.

The work of Botha and Post [3] also deals with interactive TF specification. In contrast to Kniss et al. [10], they use only the intensity information but provide a real-time visual feedback during the TF definition process. Also only based on voxel intensity is a method proposed by Weber and Scheuermann [24] for finding isovalues that represent contours related to the topological structure of the data set.

Later work is mostly focused on two-dimensional TFs that employ intensity and gradient magnitude information. Sereda et al. [21] derive the so-called *LH histogram* that represents lower and higher intensities forming the boundaries.

TF specification is done in the LH domain, where homogeneous regions are located at the diagonal of the rectangular histogram, and boundaries form clusters above that line. An improved method for a fast generation of those histograms and a semi-automatic generation of LH TFs has been recently introduced by Praßni et al. [16].

A method for picking arch structures (see [10] and Fig. 1) directly in the IGM histogram are *spatialized TFs* introduced by Roettger et al. [19]. They incorporate the spatial location of each voxel by computing the mean position of all voxels of one single bin and its variance. 2D bins are then classified using these measures. Our approach presented here adopts the idea of spatialized TFs.

A *spatial-based TF* has been proposed by Reitinger et al. [18]. For all voxels of the volume data, the minimum Euclidean distance to one or more user defined reference positions is computed and stored in a distance map. In this case, the opacity TF does not only depend on the intensity of each voxel but also on the corresponding value of the distance map. Tappenbeck et al. [22] also describe an approach for opacity TF definition based on a distance measure. Their approach requires the segmentation of a structure of interest. Its boundary serves as a reference contour for computing a distance value, which is used as second property for a 2D histogram.

Recently, *feature size* was introduced by several groups [5,8,25] as an alternative property that can be derived from image data. Correa and Ma [5] employ a scale space approach for assigning size information to each voxel. Here, intensity and feature size are represented by two 1D histograms, and consequently independent 1D transfer functions have to be defined by the user. Hadwiger et al. [8] present in their work a region growing based approach for deriving feature size information. During an unattended preprocessing stage, seeds are placed automatically and a region growing with various parameters is performed. This information is used for generating a 3D histogram with a *time* dimension standing for the different parameter settings for the region growing. Each 2D sub-space for a given time value represents a 2D histogram combining intensity and feature size. Our *structure size enhanced* (SSE) histogram approach [25] computes feature size based on a user specified tolerance value. It is presented in detail in "Material and methods".

Another very recent approach for TF-based volume visualization is the method introduced by Chan et al. [4]. They investigate the visualization of layered semi-transparent structures in volume data sets by incorporating psychological principles. Correa and Ma [6] introduced the *occlusion spectrum*—a collection of typical occlusion patterns related to several types of image data (CT, MRI, etc.)—and show how to use this information to improve the visualization of medical image data sets. Maciejewski et al. [11] describe a non-parametric clustering method for the bins of a 2D histogram, which is an alternative to the aforementioned spatial classification. A method for automating the TF generation has been proposed by Zhou and Takatsuka [26]. Their approach utilizes topological attributes that have been derived from the contour tree of the volume. Selver and Güzeliş [20] have recently introduced the *Volume Histogram Stack* (VHS) where a stack of conventional 1D histograms is generated, corresponding to the intensity distribution in the stacked image slices. Gaussian bases associated with the lobes of different tissues are determined automatically in order to improve their distinction in case that they have overlapping intensities. TFs initialized semi-automatically using their method are employed for the visualization of abdominal image data.

Material and methods

Our method for generating 3D visualizations of medical volume data combines several techniques. In the following, we first describe our approach for deriving image data that contains information about the size of the anatomical structures. Afterward, the generation of the SSE histogram including the bin classification is described in detail. This step is based on the work of Roettger et al. [19] and extended to a distance-based classification. Finally, interaction with the SSE histogram as well as the generation of the TFs for the volume rendering are explained.

Computation of the structure size image data

The main contribution of our work is the use of a *structure size image* as a second property for a 2D histogram, which is used for defining color and opacity TFs. This information is derived from the original image data using a threshold-based voxel counting method.

Image smoothing The computation of the structure size image is based on a homogeneity assumption. This means that neighbored voxels are considered as belonging to the same anatomical structure if they have similar intensities. Due to noisy image data or imaging artifacts, this ideal case is not fulfilled for most of the data sets. In order to compensate for that, we perform an edge preserving smoothing step using a 3D anisotropic diffusion filter [14]. Thus, also for noisy data (e.g. MRI) this assumption can be made.

The 26 neighborhood Each voxel in 3D has a number of 26 direct neighbors connected to the center voxel by faces (6 voxels), edges (12 voxels), and corners (8 voxels). Virtual lines drawn from the center voxel to each of its neighbors define 26 directions (see Fig. 2). These are the search directions used for the threshold based voxel counting. Thus, for each voxel of the original data set, the algorithm steps along

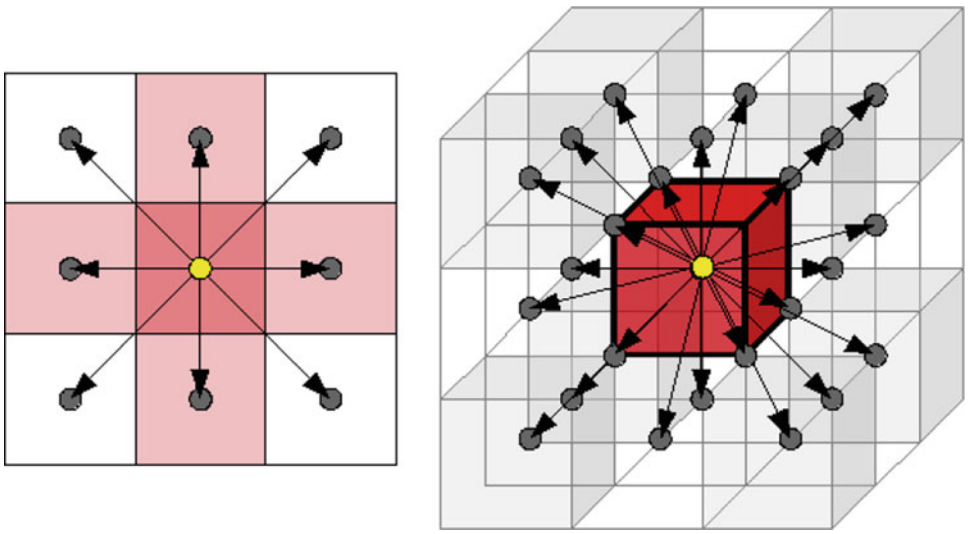

Fig. 2 The neighborhood and the corresponding search directions for a voxel shown in 2D (*left*) and 3D (*right*). Along the lines which connect the center voxel (*yellow*) with its direct neighbors (*gray*) a depth-first search is performed as long as a threshold condition is fulfilled

Listing 1 The algorithm for estimating the structure size. Code for checking boundary conditions etc. is omitted for brevity.

```
min_S := min(dimX, dimY, dimZ) / 2;
N := 8;
I_max = maximum intensity in the input image;
for all voxels of input & output {
    size_struct := 0;
    I_ref := intensity of the input voxel;
    t_l := I_ref - tau * I_max, t_u := I_ref + tau * I_max;
    for all directions of the N26 neighborhood {
        step_count := 0;
        I_curr := I_ref;
        while (I_curr >= t_l && I_curr <= t_u) {
            Do one step into the corresponding direction;
            I_curr := intensity at the current input voxel position;
            step_count++;
        }
        for (scale := N, divisor := 1; scale >= 1; scale--, divisor*=2) {
            S := min_S / divisor;
            if (step_count >= S) {
                size_struct += N;
                break;
    }   }   }
    output_voxel := size_struct;
}
```

the rays defined by these directions as long as a threshold condition is fulfilled.

Multi-scale voxel counting For each reference voxel v_{ref}, the number of surrounding voxels with a similar intensity is counted. An upper threshold t_u and a lower threshold t_l is computed depending on the intensity I_{ref} of v_{ref}. In this work, we used a fixed threshold range which is given by $\mathcal{I}_{range} = [I_{ref} - \tau \cdot I_{max}, I_{ref} + \tau \cdot I_{max}]$, where I_{max} stands for the maximum intensity contained in the data set and τ is a tolerance value in the interval (0, 0.5). A typical value used for τ in this work is $\tau = 0.05$.

For each voxel of the data set, a depth-first search is performed where each voxel becomes the reference voxel v_{ref} once during the search. The algorithm (see Listing 1) tries to do the maximum number of steps in each of the 26 directions starting at v_{ref} as long as the intensity I_{curr} at the current voxel position is within the interval $\mathcal{I}_{range}$. These steps are counted for each direction. However, the structure size value assigned to the voxel v_{ref} is not simply the accumulated step value for all directions.

Instead, we use a multi-scale approach that provides structure size images with less inhomogeneities and artifacts. A fixed number of 6 scales is defined. Each scale value N_{scale} corresponds to a specific number of steps. For the largest scale $N_{scale} = 6$, the number of steps S_{min} is given by the minimum dimension of the data set considering x, y, and z

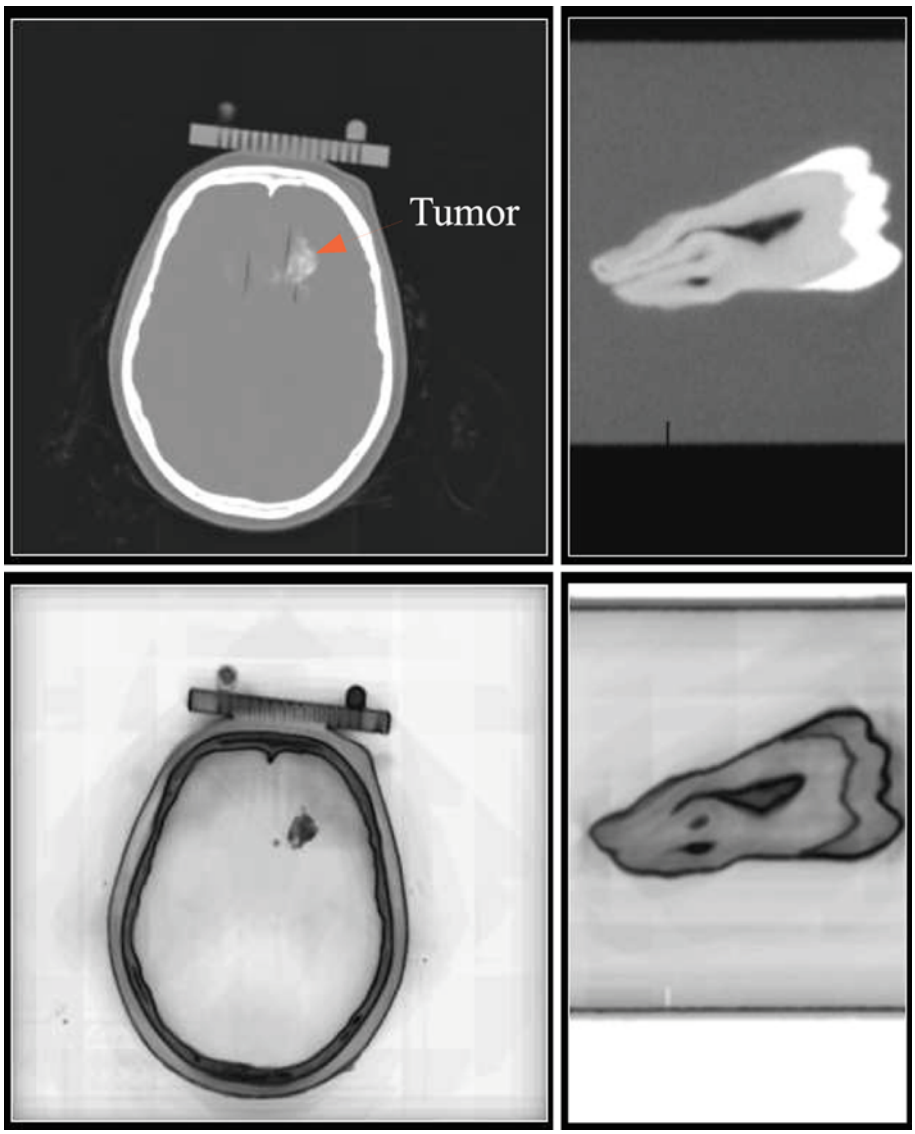

Fig. 3 Examples for structure size images (*bottom*) in comparison with the original images (*top*). From left: CT of a head, CT of a tooth. Representative 2D slices are shown

direction divided by 2. Thus, half of the minimum dimension (in voxels) of the data set corresponds to a number of steps S_{min} which get the scale value $N_{scale} = 6$. If S steps with $S_{min} > S \geq S_{min}/2$ can be done, this corresponds to $N_{scale} = 5$. For s steps with $S_{min}/2 > S \geq S_{min}/4$ it corresponds to $N_{scale} = 4$, and so on. The scale value is computed for each of the 26 directions from the number of steps done and then accumulated. This final value is assigned to the structure size image at position v_{ref}. Afterward, the obtained structure size image data is smoothed employing a uniform 3D Gaussian kernel with $\sigma = 1.0$.

An interesting property of our structure size computation approach is that boundaries which represent the transition between regions of different intensities are rather small regions with a quite unique gray value. Thus, these are structures of relatively small size and they mostly have a dark appearance in the structure size image (see Fig. 3). This means that boundaries can be emphasized by assigning a high opacity value to small structures and a smaller opacity value to larger structures. This is a desired similarity to 2D TFs, which are defined on conventional IGM histograms.

Generation of the SSE histogram

The SSE histogram can be generated by combining the intensity image with the structure size image. At each 2D bin position, the number of voxels with the corresponding combination of intensity and structure size is stored. In order to reduce noise in the SSE histogram, we use a k-neighborhood with $k = 1$ as it has been proposed by Roettger et al. [19]. To some extent, this avoids under-populated bins. The supersampling however, which is also proposed in that work, is not used because it requires a large amount of memory.

A logarithmic scale is used for the visualization of each bin's frequency in the 2D histogram.

Spatial classification A severe limitation of histograms is the fact that they represent mainly statistical information—a probability distribution—and that any spatial information is lost. However, this information can be kept if the bins are weighted by the mean and the covariance of the spatial locations of all voxels that contribute to that bin. This concept, called *spatiogram*, has been introduced by Birchfield and Rangarajan [2].

Roettger et al. [19] propose a quite similar approach for enhancing a conventional IGM histogram with spatial information by computing the barycenter $\mathbf{b}(T)$ and the variance $v(T)$ of all voxels that fall into the same 2D bin T. The spatial relationship between an arbitrary bin T and a reference bin T_0 can be evaluated by considering the distance norm

$$N(T, T_0) = \|\mathbf{b}(T) - \mathbf{b}(T_0)\| + |v(T) - v(T_0)| \,. \tag{1}$$

If the value of $N(T, T_0)$ falls below a user-defined value—the feature radius r_F—the two bins T and T_0 are considered to belong to the same feature. Thus, the bins in the 2D histogram are classified, and a unique randomly defined color is assigned to them. See reference [19] for details.

Distance-based classification Another possibility for spatially classifying the 2D bins is the use of a scalar distance measure d instead of the three coordinates x, y, z. This requires the definition of one or more reference positions $\mathbf{p}_1 \ldots \mathbf{p}_N$ in order to compute a distance map, where for each voxel at position $\mathbf{p}_V$ its minimum distance

$$d_{min}(T) = \min(\|\mathbf{p}_V - \mathbf{p}_1\| \,, \ldots, \|\mathbf{p}_V - \mathbf{p}_N\|) \tag{2}$$

to one of the reference points is stored [18]. A classification of the 2D bins T can be done using such a distance map by computing the mean distance $\mu_d(T)$ and its variance $v_d(T)$ for all voxels that contribute to the same bin, and cluster those bins where the modified distance norm

$$N_d(T, T_0) = |\mu_d(T) - \mu_d(T_0)| + |v_d(T) - v_d(T_0)| \tag{3}$$

falls below the corresponding feature radius $r_{F,d}$. Coloring can be done as described earlier.

Interaction with the SSE histogram

Areas in the SSE histogram can be selected and deselected by pointing and clicking on the corresponding cluster. In extension to the work of Roettger et al. [19], our implementation allows to control the selection of clusters using two different modes: in global mode all 2D bins with the same label are selected, in local mode only those 2D bins which are connected (over bins of the same label) to the 2D bin picked by the user are selected. Its purpose is a refined picking in case that 2D bins of the same label form disjointed clusters. A tool for selecting and deselecting arbitrary regions in the SSE histogram has been implemented by means of closed splines with control points adjustable in their number and position.

Visualization of the volume data

The SSE histogram is a means for selecting voxels with specific properties for the visualization and for configuring the corresponding opacity and color TFs. In our approach, we employ a semi-automatic opacity assignment. As its result, each position (u, v) of a 2D bin has a well-defined opacity value $f_{op}(u, v)$. Opacity increases with increasing intensity and decreases with increasing structure size. Consequently, large structures appear more transparent, whereas small structures are rendered more opaque. The user has to define the desired opacity values for all four corners of the SSE histogram, and the values in between are linearly interpolated. In addition, a global opacity value $\omega \in [0, 1]$ can be set by the user. Hence, all voxels which are represented by the 2D bin at position (u, v) will be rendered with an opacity $\omega \cdot f_{op}(u, v)$.

Different coloring modes are employed. The first one is a standard gray scale—a linear ramp from black to white over different shades of gray. The smallest values in the data set appear dark, whereas the largest ones appear bright. The other methods use the color assigned to the 2D bin in the SSE histogram for setting the color of the corresponding voxels in the 3D rendering. The color assignment to the 2D bins can be chosen by the user by defining the hue range in HSV space which should be used and furthermore by defining:

1. The relative number of voxels represented by the 2D bin: small hue values for lowly populated bins, large hue values for highly populated bins,
2. The value of the structure size image: hue assignment for small and large values as earlier,
3. A randomly assigned color during the spatial and distance based classification, respectively.

Saturation and value components in HSV space depend on the structure size, i.e., the position of the 2D bin with respect to the y axis. With increasing structure size, saturation linearly decreases, whereas value linearly increases. This provides a darkening of the boundaries as well as a color attenuation for larger structures. The value ranges for both can be set by the user.

Results

The methods described earlier have been integrated into a visualization application. In this section, we show visualization examples for clinical data sets as well as publicly

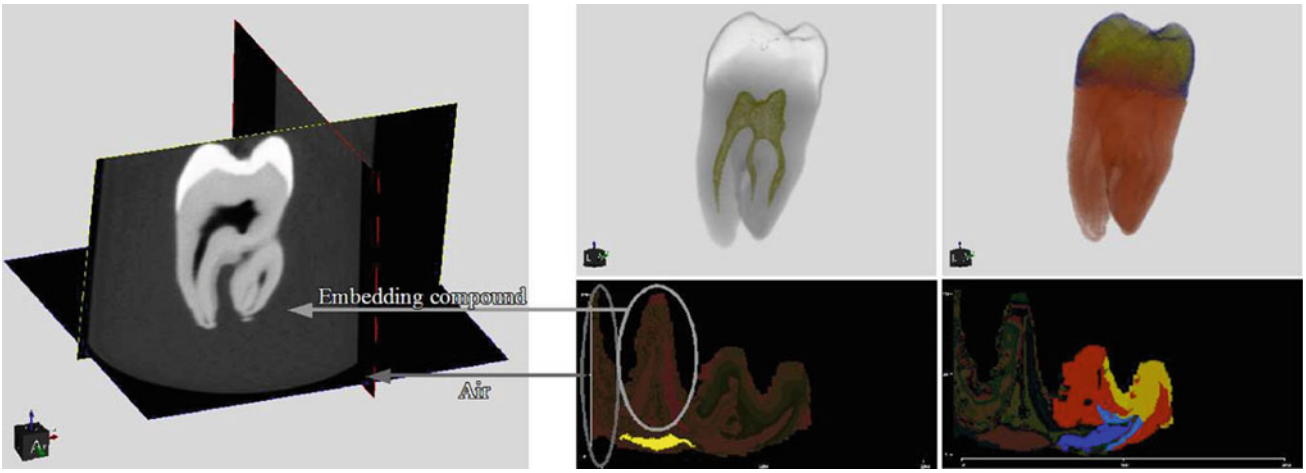

Fig. 4 SSE histograms and 3D visualization for CT data of a tooth. On the *left*, the original image data is shown using a plane-based visualization. It can be seen that the tooth is embedded in some compound which is surrounded by air. The latter structures mainly occupy the left part of the SSE histogram. A clustering of the 2D bins by spatial classification and a highlighting of the nerve cavity is shown in the *middle* column. Distance-based classification using one reference position located at the center of the tooth and a selective visualization of dentine/cementum (*reddish*) as well as enamel (*yellowish*) and its outer boundary (*blueish*) is shown in the *right* column. (See [19] for visualization examples employing an IGM histogram)

available image data taken from [12,23]. Thereby, we investigate

- the ability to highlight specific anatomical structures,
- the hypothesis that the SSE histogram is more intuitive than the conventional IGM histogram.

Before discussing the example data sets, a general remark has to be made. The relative volume occupied by voxels that belong to the same feature is not directly reflected in the SSE histogram. If as an example we consider the air which surrounds the scanned object in a CT data set, we notice that in the structure size image a quite similar size value is assigned to all the air voxels (see bottom left of Fig. 3). Consequently, their huge number is represented by a rather small area in the SSE histogram since they all fall into adjacent 2D bins. How far a structure is distributed in the SSE histogram is determined by the number of size values computed for the corresponding voxels. For instance, more different size values will be assigned to a rather concave-shaped (homogeneous) structure than to a convex-shaped structure. Thus, the latter will occupy a smaller area in the SSE histogram.

Tooth

This data set is a CT scan of a single tooth taken from [23]. The tooth itself mainly consists of three parts: an enamel cap, the dentine/cementum, and the nerve cavity (see also Fig. 3). These regions are quite homogeneous, thus the structure size computation as well as the classification of the 2D bins in the SSE histogram perform well. We tried to highlight the nerve cavity by taking into consideration its dark appearance in the original image and its rather small size compared to the other dark parts of the image—air and a compound where the tooth is embedded. The point-and-click selection of the corresponding cluster in the spatially classified SSE histogram perfectly highlights the nerve which is rendered in color, whereas the other voxels remain unchanged (see Fig. 4, middle).

In addition, we performed the alternative distance-based classification introduced in this work. In this case, one reference position located at the center of the tooth was set. Three large regions emerged in the right part of the SSE histogram representing the bright voxels of the original image. By selecting them, three distinct regions of the tooth are highlighted: the dentine/cementum, the enamel cap, and the outer boundary of the latter structure (see Fig. 4, right). The areas which were not selected for inclusion in the visualization are mainly the nerve, the embedding compound, and the surrounding air.

Thus, by using little prior knowledge about the data set interesting structures can easily be selected and emphasized in the 3D visualization. IGM-based 3D visualizations of this data set can be found in Roettger et al. [19].

Abdomen

The second example is taken from the Osirix data base [12]—BREBIX data set. This is a CT scan of the abdominal region of a patient with a renal cell carcinoma acquired at the venous phase. We selected this example due to the similar gray value intensities for liver and spleen which makes the separation of

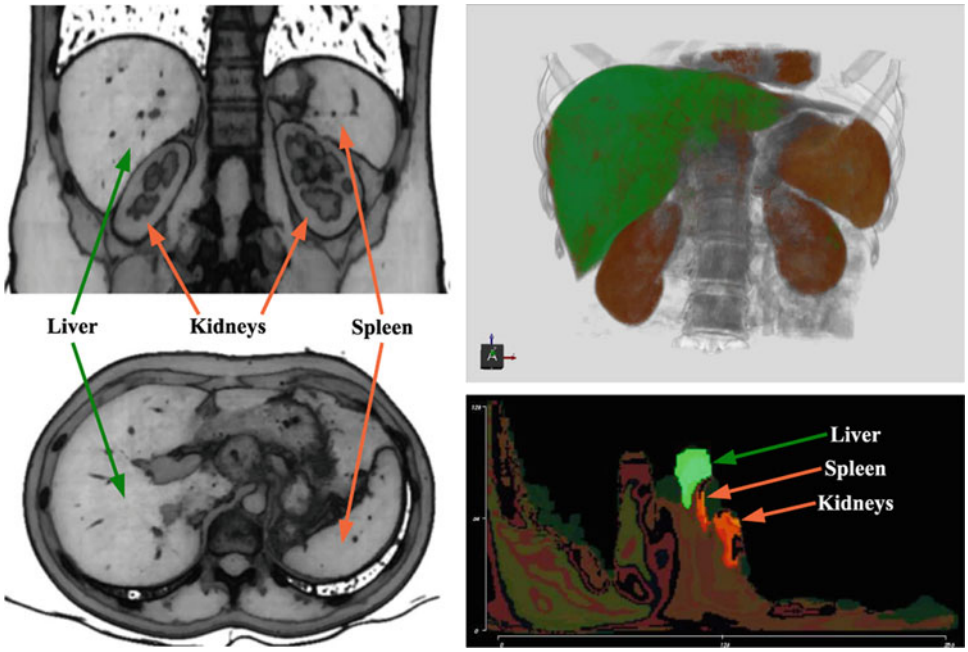

Fig. 5 Structure size images for the BREBIX data set taken from [12] show slightly different size values assigned to liver and spleen (*left*). In the SSE histogram, both structures can be separated from each other even though their intensity values overlap (*bottom right*). The spatial classification performed on the SSE histogram clusters the 2D bins belonging to the liver (*green*) and those representing spleen and kidney (*orange/brown*). Thus, in the corresponding 3D visualization the organ structures can easily be distinguished (*top right*)

these two tissues challenging [20]. The left of Fig. 5 shows two slices of the derived structure size image data scaled to the gray value range $[0, \ldots, 127]$. Here, the structure size values for the liver are mainly between 80 and 95, whereas those for the spleen range from 70 to 80.

Due to their different values in the structure size image, liver and spleen are represented by different 2D bins in the SSE histogram (Fig. 5). As expected, this separation of both structures is possible in case that structure size is employed as a second property. During spatial classification, different colors are assigned to both structures, and they are differently highlighted in the final 3D rendering (Fig. 5). Only some voxels in the boundary area of the liver are incorrectly highlighted showing the same orange/brown coloring as kidney and spleen. Presumably, for those voxels also the size values overlap. Different reasons for that can be identified.

Firstly, their exist inhomogeneity concerning the appearance of abdominal organs in the image data, since e.g. in CT their absorption behavior is slightly different over their interior. The anisotropic diffusion which is performed as a preprocessing step ("Material and methods") cannot compensate for this.

Secondly, the structure size computation employs a homogeneity criterion. Depending on the selection of the tolerance value τ ("Material and methods"), voxels with a slightly different gray value may be considered as being part of the structure or not. Trying different tolerance values may lead to an improvement of the structure size computation, but we did not perform those tests in this work.

Thirdly, a Gaussian smoothing is performed on the structure size image. This mainly affects voxels in the boundary area of the organs, since edges are not preserved.

Finally, the spatial classification [19] may cluster 2D bins that represent spatially neighbored voxels, which belong to different organs. In addition, the influence the user has on the assignment of colors is limited during that step.

Head

CT This data set was acquired during a brachytherapy of a brain tumor, which is well visible in the CT data set (see Fig. 3). Here, we again compared IGM versus SSE histogram based highlighting of anatomical structures (see Fig. 6). It turned out that it was impossible to select only the brain in case of the spatially classified IGM histogram, since the selection of a very small cluster at the bottom of the corresponding 2D histogram highlighted the brain as well as adjacent structures. In addition, finding this cluster was again a tedious trial-and-error task.

It was much more intuitive to select clusters in the spatially classified SSE histogram that correspond to large structures—knowing that the brain is relatively large compared to other structures in this data set. This is also an example for what we have explained at the beginning of this section. Although the brain voxels occupy a large region in the original image data, in the structure size image a similar value is assigned to those positions. Consequently, they are represented by a rather small area in the SSE histogram. An interesting effect when brain-related clusters were picked was that not only the brain was highlighted. It also revealed the tumor position that appeared as a hole within the color highlighted brain.

We concluded from this data set that in the IGM histogram it is more likely that different structures are represented by

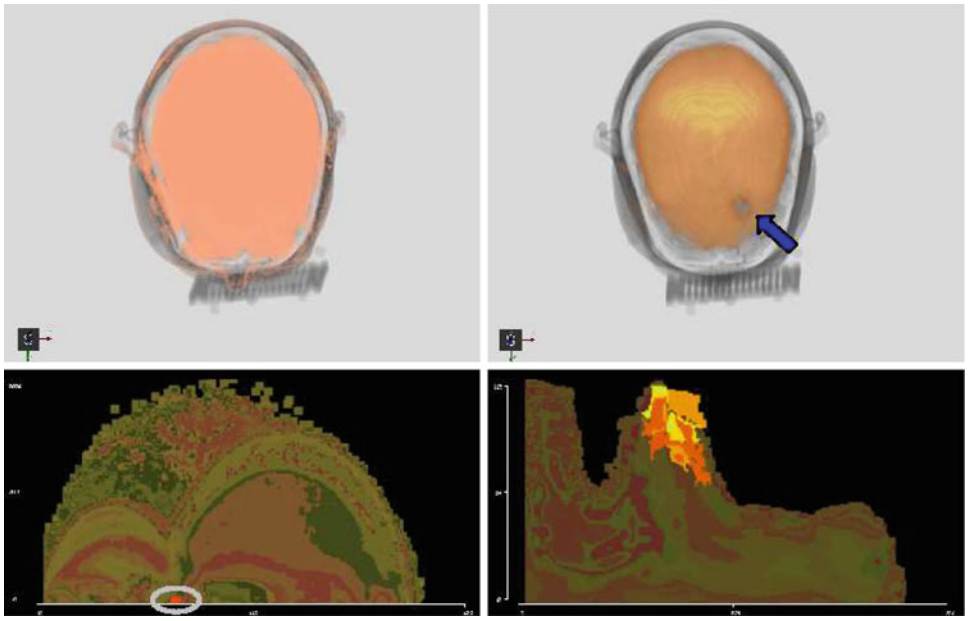

Fig. 6 A CT data set of a head acquired during brachytherapy. In case of the IGM histogram (*left*), the selection of one very small cluster highlights the brain and adjacent structures. Using the SSE histogram (*right*), a more detailed and more intuitive selection of the brain is possible. In addition, in this case the tumor position (*arrow*) can be easily perceived in the 3D visualization (see also Fig. 3)

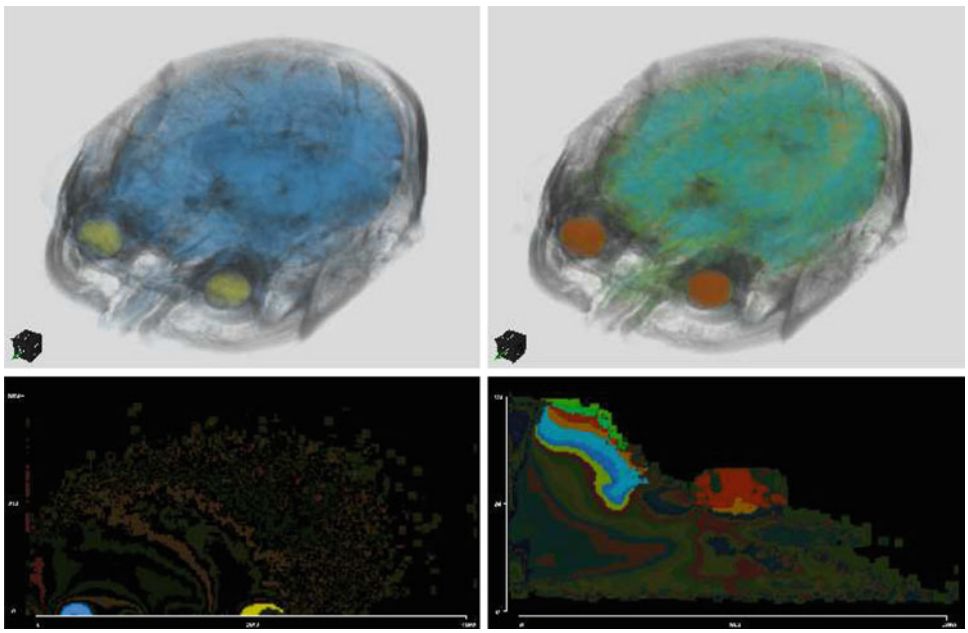

Fig. 7 MRI data set of a head with two structures highlighted—the eyes and the brain. In the IGM histogram (*left*) they are represented by small clusters located at the bottom of the 2D histogram. The SSE histogram (*right*) has a more intuitive meaning, and brain and eyes are easy to select, being the largest structures in their corresponding gray value ranges

only a small number of 2D bins. Thus, by picking them many voxels in the 3D data set are affected. In this case, this makes it more difficult to perform a detailed adjustment of visualization parameters compared to the SSE histogram, because in the latter case different structures are presumably of different sizes. Consequently, the corresponding voxels are likely to be located at different 2D bin positions in the SSE histogram.

MRI I This is another data set from [23]. Again, we compared the possibility to highlight different structures by selecting clusters in the spatially classified IGM and SSE histograms (see Fig. 7). The separate selection of the eyes was possible in both cases, however, again much more intuitive in the SSE histogram: the eyes are the largest structures for the corresponding gray value range. Selecting one additional cluster in the IGM histogram for highlighting the voxels representing the brain also selected some adjacent structures. In the SSE histogram, more than one cluster had to be selected in order to completely color the brain voxels. In this case, different parts of the brain (white and gray matter) were colored slightly differently. We conclude that the SSE histogram discriminates better between different anatomical structures in this data set.

MRI II Beside the cortex, which was highlighted in the previous example, also its distinction from subcortical structures is of interest. Both overlap in traditional 1D histograms. Our approach is able to separate them and highlight

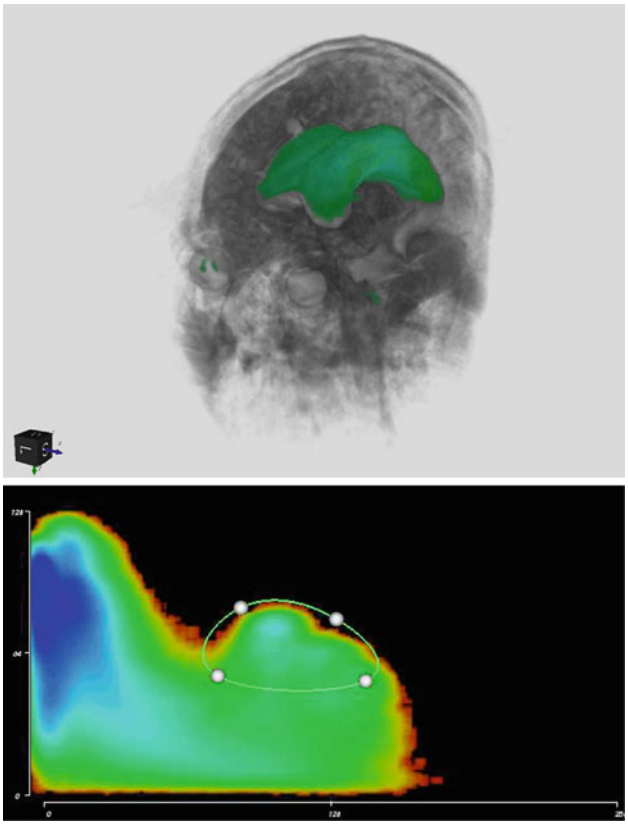

Fig. 8 Using prior knowledge that the *corpus callosum* appears bright in the original image data allows for the correct delineation of an arbitrary region in the SSE histogram corresponding to that structure. (Also a region in the right eye is highlighted which is due to a coarse placement of the curve surrounding the area of interest in the SSE histogram without an additional fine-tuning.) The coloring of the 2D histogram corresponds to the number of voxels represented by each bin

them individually. A prerequisite is that these structures are separated from each other by a boundary of voxels with different intensities. If this is the case, even subcortical structures with similar gray values can be separated employing the SSE histogram. In Fig. 8 an example is shown (image data taken from [23]), where prior knowledge that the *corpus callosum* appears bright in the image data has been used for defining an arbitrary area in the SSE histogram which selectively visualizes this subcortical structure. However, this structure is the largest white matter fiber pathway being represented by homogeneous intensities throughout the bundle. This makes its selective visualization easier compared to smaller and less homogeneous structures.

Thorax

In a clinical thoracic CT data set that mainly contains the heart, we tried to selectively render the lung structures (vessels and parenchyma) as well as the large cardiac cavities. For this data set, we compared how this task can be performed using the IGM as well as the SSE histogram. Selection of the structures in the spatially classified IGM histogram was possible. However, it was a lot of trial-and-error work, since picking clusters with a gradient magnitude value (the y position in the 2D histogram) which corresponds to the structures under consideration was not simple. In the case of the SSE histogram, it was much easier to select large structures with a gray value corresponding to the cardiac cavities employing our knowledge about the relative size of these heart structures. Selecting correct clusters corresponding to the lungs was not as easy, but it was also quickly done after some tries.

In addition, the selection of arbitrary regions in the SSE histogram by manipulating a closed curve was tested. In this case, we started with a color mapping of the 2D bins defined by each bin's frequency. Thus, a color scale ranging from red over green to blue was used for representing increasing bin frequencies. Taking this additional information into account, the two main structures could be selected with more detail compared to the clustering after spatial classification. By applying a color map representing the computed structure size values, different colors are assigned to the lungs and the cardiac cavities (see Fig. 9).

Discussion

In this work, we have presented and investigated a novel approach for the visualization of volumetric medical data sets based on 2D histograms. The novelty regarding the 2D histogram is the use of the *size of anatomical structures* as a second property. We have introduced a threshold-based method for computing this structure size image data, which is a trade-off between computational speed and accuracy.

In contrast to the work of Correa and Ma [5], our structure size computation provides image data with sharper edges compared to the quite smooth appearance of the structures generated by their scale space–based approach. Thus, one single structure is characterized by a rather unique structure size value when our method is employed. In addition, in our case boundaries are represented by very small structure size values which makes a distinction between organ borders and their interior part in the SSE histogram quite easy. Furthermore, in our work a 2D histogram representation is employed instead of two independent 1D histograms (for intensity and feature size) which are used in [5] for the opacity TF definition. In our opinion, the interaction with this 2D SSE histogram is easier and more straightforward than the manipulation of two 1D histograms.

Compared to another recently introduced approach by Hadwiger et al. [8], our structure size computation is done with less computational effort. The computation of the structure size image data is done in a time ranging from a few

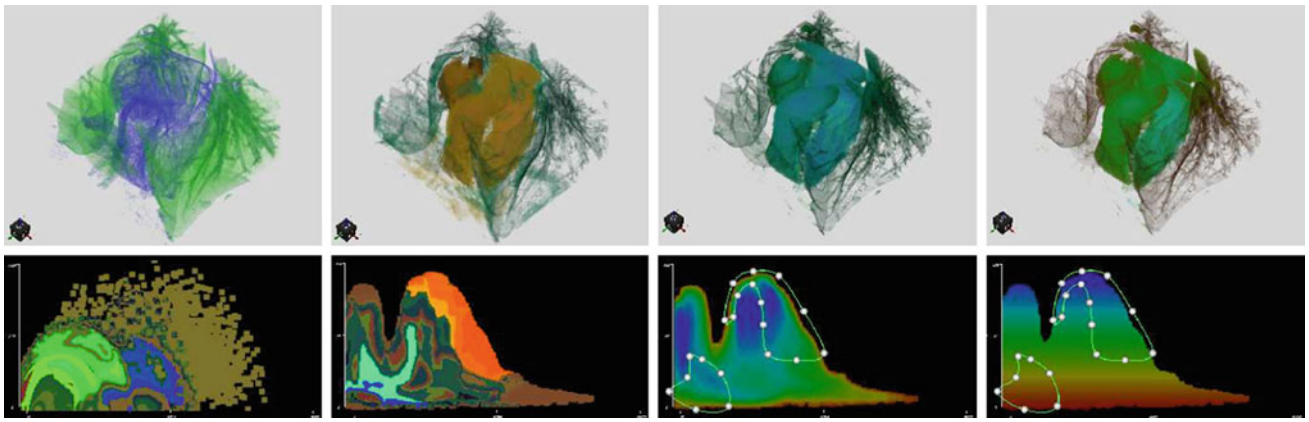

Fig. 9 A thoracic CT containing the lungs and the heart. A selective rendering of these structures is shown. The *left* images show the visualization based on the IGM histogram where the typical appearance of such renderings with strongly emphasized, opaque boundaries can be perceived. The three *right* images show SSE histogram-based visualizations employing spatial classification as well as arbitrary region selection. In the rightmost images color mapping is based on the values of the structure size image, whereas in the images aside the bin frequency determines the color. (See "Visualization of the volume data" for the different types of color mapping)

seconds up to 2 min for larger data sets (e.g. a thoracic CT). The aforementioned method, which employs an unattended region growing approach, takes at least one order of magnitude longer for data sets of comparable sizes.

The VHS introduced by Selver et al. [20] is an alternative method that yields good results for the visualization of abdominal image data. Their free selection of several Gaussian units for grouping different types of tissue gives more flexibility to the user than the spatial classification [19] used here. However, this also makes things more difficult for the unexperienced user. It would be interesting, to compare IGM, VHS, and SSE in a user study mainly performed with radiologists to obtain (objective) measures for each of the methods.

The main benefit of our method is the property of the SSE histogram that the location of a 2D bin can be connected with an intuitive meaning—*voxels of a certain intensity that belong to a structure of a particular size*. This is better adapted to unexperienced users regarding image processing—e.g. clinicians—than conventional 2D histograms which combine intensity and gradient information. In such IGM histograms, homogeneous regions of the same intensity often fall into the same 2D bin—even if they belong to different structures—whereas in the SSE histogram they are more likely to be located at different bin positions. Thus, the SSE histogram discriminates much better between different structures, and makes it possible to selectively highlight them in 3D visualizations.

Furthermore, we have shown how the new 2D histogram can be applied to the definition of TFs for the 3D visualization of volumetric data. An opacity TF can be semi-automatically set up based on the combined parameters of intensity and structure size. We have also presented several approaches for incorporating spatial parameters into the SSE histogram, for the interaction with the histogram data, and how this information can be used for the definition of color TFs. Real world examples using medical image data coming from different modalities have shown the usefulness of our approach for generating 3D visualizations.

For the volume rendering, we employed a simple pre-classified ray casting approach [17]. Using more sophisticated rendering methods like pre-integration techniques [7] which have been used e.g. by Roettger et al. [19] can provide visualizations which are even more pleasing to the eye.

Alternative methods for interacting with the SSE histogram for cluster selection and color assignment should be investigated in the future. An improvement for increasing the method's usability would be a *point-and-click* interaction metaphor for merging different clusters. Furthermore, the assignment of an arbitrary color to one or more clusters using a standard dialog for color selection would be beneficial. The assignment of opacity to each cluster could quickly be done by using a click-and-move interaction the radiologist also uses for setting the window level in 2D image slices. As similar as those parameter settings are to the interfaces the radiologist is used to, as easier it is for him or her to profit from the here presented visualization method.

Certainly, the quality of the SSE histogram and the subsequent 3D visualization heavily depend on a correct computation of the structure size image. Our approach relies on a homogeneity assumption which is fulfilled for a large range of data sets after image smoothing by applying an anisotropic diffusion filter. In the case of extremely noisy MRI data sets or ultrasound images however, this may represent a problem. A handling of these cases would require a more sophisticated method for computing the structure size which

could for instance incorporate a shape prior for the anatomical structures of interest contained in the image data. Consequently, future work will focus on improving the computation of the structure size image.

Another direction of future work could also be to use the SSE histogram for building a prior model for subsequent visualization. Assuming that the gray value distribution in different images of the same part of the body is similar, statistical models could be learned which describe the typical combination of sizes and intensities of anatomical structures. Using this information, *undesired* regions like air—typically a large and dark region—could be excluded from the visualization. Additionally, one could try to reduce noise by suppressing too small structures in the visualization.

Conclusion

Generating meaningful 3D visualizations from acquired medical image data remains a challenging problem. We have presented a contribution to that field by introducing the *size of anatomical structures* as a second feature besides intensity for defining opacity and color transfer functions. Its main strength is the intuitiveness of the approach, which makes it also usable for clinicians.

Acknowledgment Our thanks go to Kevin Schelten for proof-reading this manuscript.

References

1. Bajaj CL, Pascucci V, Schikore DR (1997) The contour spectrum. Visualization 1997, Proceedings, pp 167–173, Oct 1997
2. Birchfield ST, Rangarajan S (2005) Spatiograms versus histograms for region-based tracking. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005, vol 2. pp 1158–1163, June 2005
3. Botha CP, Post FH (2002) New technique for transfer function specification in direct volume rendering using real-time visual feedback. In: Seong K, Mun (eds) Medical imaging 2002: visualization, image-guided procedures, and display, vol 4681. pp 349–356. SPIE
4. Chan M-Y, Wu Y, Mak W-H, Chen W, Qu H (2009) Perception-based transparency optimization for direct volume rendering. IEEE Trans Vis Comput Graph 15(6):1283–1290
5. Correa C, Ma K-L (2008) Size-based transfer functions: a new volume exploration technique. IEEE Trans Vis Comput Graph 14(6):1380–1387
6. Correa C, Ma K-L (2009) The occlusion spectrum for volume classification and visualization. IEEE Trans Vis Comput Graph 15(6):1465–1472
7. Engel K, Kraus M, Ertl T (2001) High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In: HWWS 2001: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on graphics hardware. ACM, New York, pp 9–16
8. Hadwiger M, Laura F, Rezk-Salama C, Hllt T, Geier G, Pabel T (2008) Interactive volume exploration for feature detection and quantification in industrial CT data. IEEE Trans Vis Comput Graph 14(6):1507–1514
9. Kindlmann G, Durkin JW (1998) Semi-automatic generation of transfer functions for direct volume rendering. IEEE Symp Vol Vis pp 79–86, Oct 1998
10. Kniss J, Kindlmann G, Hansen C (2002) Multidimensional transfer functions for interactive volume rendering. IEEE Trans Vis Comput Graph 8(3):270–285
11. Maciejewski R, Woo I, Chen W, Ebert D (2009) Structuring feature space: a non-parametric method for volumetric transfer function generation. IEEE Trans Vis Comput Graph 15(6):1473–1480
12. OsiriX. http://www.osirixmaging.com/resources/, 2010. last visited: 2010/04/22
13. Pekar V, Wiemker R, Hempel D (2001) Fast detection of meaningful isosurfaces for volume data visualization. In: VIS 2001: Proceedings of the conference on visualization 2001. IEEE Computer Society, Washington, pp 223–230
14. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 12(7):629–639
15. Pfister H, Lorensen B, Bajaj C, Kindlmann G, Schroeder W, Avila LS, Raghu KM, Machiraju R, Lee J (2001) The transfer function bake-off. IEEE Comput Graph Appl 21(3):16–22
16. Praßni J-S, Ropinski T, Hinrichs KH (2009) Efficient boundary detection and transfer function generation in direct volume rendering. In: Proceedings of the vision, modeling, and visualization workshop 2009 (VMV09), pp 285–294. Otto-von-Guericke-Universität Magdeburg
17. Preim B, Bartz D (2007) Visualization in medicine. Theory, algorithms, and applications 1st edn. Morgan Kaufmann Series in Computer Graphics, Morgan Kaufmann
18. Reitinger B, Zach C, Bornik A, Beichel R (2004) User-centric transfer function specification in augmented reality. In: Proceedings of WSCG (Plzen, Czech Republic, February 2004), pp 355–362
19. Roettger S, Bauer M, Stamminger M (2005) Spatialized transfer functions. In: Proceedings of EUROVIS 2005: Eurographics/IEEE VGTC symposium on visualization, pp 271–278
20. Selver MA, Güzeliş C (2009) Semiautomatic transfer function initialization for abdominal visualization using self-generating hierarchical radial basis function networks. IEEE Trans Vis Comput Graph 15(3):395–409
21. Sereda P, Bartroli AV, Serlie IWO, Gerritsen FA (2006) Visualization of boundaries in volumetric data sets using LH histograms. IEEE Trans Vis Comput Graph 12(2):208–218
22. Tappenbeck A, Preim B, Dicken V (2006) Distance-based transfer function design: Specification methods and applications. In: Simulation und visualisierung, pp 259–274. SCS-Verlag
23. VolVis. http://www.volvis.org, 2009. last visited: 2009/12/11
24. Weber GH, Scheuermann G (2002) Topology-based transfer function design. In: Villanueva JJ (ed) Proceedings of the second IASTED international conference on visualization, imaging, and image processing, pp 527–532. ACTA Press
25. Wesarg S, Kirschner M (2009) Gradient magnitude vs. feature size: comparing 2D histograms for transfer function specification. In: CGI 2009: Computer graphics international. ACM, New York, pp 115–119
26. Zhou J, Takatsuka M (2009) Automatic transfer function generation using contour tree controlled residue flow model and color harmonics. IEEE Trans Vis Comput Graph 15(6):1481–1488