Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2002)
D. Ebert, P. Brunet, I. Navazo (Editors)

Interactive Previewing for Transfer Function Specification in Volume Rendering

Charl P. Botha and Frits H. Post

Data Visualisation Group
Delft University of Technology, The Netherlands
{c.p.botha,f.h.post}@its.tudelft.nl
http://visualisation.tudelft.nl/

Abstract

This paper presents a new technique for supplying meaningful visual feedback during direct volume rendering transfer function specification. The technique uses meta-data calculated during a pre-processing step to generate interactively an approximate volume rendering that is voxel-registered with a single user-selected slice. Because of the registration, this preview can easily be alpha-blended with a grey-scale image of the data that is being volume rendered. In this way, the user gets real-time visual feedback on her transfer function specification with regards to both *the expected composited optical properties* and *the "fidelity" (how closely the rendering matches the original data) of the resulting rendering.*

1. Introduction

Direct Volume Rendering[1,2] (DVR) is an important and useful technique for visualising structures in volumetric data. However, the difficulty in specifying the DVR transfer function[3] can be prohibitive to the deployment of this technique in practical visualisation applications. The transfer function is a critical component of the volume rendering process that specifies the relation between scalar data (e.g. computerised tomography Hounsfield units), as well as derivative values (e.g. the gradient volume of an MRI dataset), and optical characteristics (e.g. colour and opacity).

Our work attempts to give the user full control over the transfer function specification and to supply real-time visual feedback in such a fashion that the system's reliance on the user's visualisation expertise is mimimised and the leverage of the user's application-specific (e.g. clinical) knowledge is maximised. The visual feedback is supplied with a special volume rendering preview technique.

The preview technique has been designed to give a good indication of the composited optical properties in the resultant volume rendering. In addition, our technique yields explicitly data-registered feedback, meaning that the fidelity of the resultant rendering (and the transfer function) can be checked. On a higher level, the one-to-one correspondence between data and volume rendered structures is accentuated.

In section 2 we discuss existing methods for finding transfer functions. Section 3 documents our new scheme and in section 4 we show some examples that demonstrate its effectivity. We summarise our work and point out possible avenues for future research in section 5.

2. Related Work

Finding good transfer functions has been listed among the top ten problems in volume visualisation[3]. Consequently, much effort has recently been spent on improving this situation. Existing schemes range from fully manual to fully automatic techniques for finding transfer functions.

Probably the oldest method of finding a transfer function is by trial and error. This usually involves manipulating a transfer function whilst periodically checking the resulting volume rendering. If special volume rendering hardware that can do this at interactive rates is not available, this can be a very laborious and time-consuming process.

Another very interesting technique that is based on the design galleries paradigm[4] generates many volume renderings simultaneously, each representing a different configuration

of the transfer function. The user selects the renderings that satisfy her requirements and thus implicitly optimises the transfer function. The considerable challenges here are to generate automatically the different transfer functions that are going to generate a wide spread of dissimilar output renderings[5] and to present these renderings to the user in an effective way.

As potentially hundreds of different renderings have to be made, this technique does rely on fast rendering hardware being available to reach its full potential as an interactive method.

Also take into account that many of the optimised software volume rendering techniques such as shear-warp factorization[6] require pre-calculated transfer function lookups, which makes them less applicable to this problem where the transfer functions are being continuously modified.

The work of König and Gröller combines elements of the design galleries and trial-and-error techniques with the required use of real-time raycasting hardware[7]. They present transfer function specification as a simplified three step process: First the user indicates scalar ranges of interest, then she assigns colours to these ranges and finally she assigns opacities. Numerous feedback renderings are performed during this process in order to guide the user's choices. This technique simplifies the specification to quite an extent.

In a previous paper[8], we proposed a fast feedback-based method that can be implemented without real-time raycasting facilities. It transforms the current slice with the current transfer function (as it is being edited) and then alpha-blends a mapping of this transformed slice with a greyscale slice of the data that is about to be rendered. Because it is essentially visualising instantaneous optical properties (i.e. not accumulated as in real volume rendering) it is less effective at previewing lower opacities. A user-tunable exponential opacity compensation is utilised to remedy this problem. Although this works satisfactorily, we required a mathematically defensible foundation. The work presented in this paper is based on the same overlay idea, but adds a predictive raycasting accumulation and optional shading.

Bajaj's contour spectrum[9] consists of metrics that are computed over a scalar field. This spectrum, presented as a user interface component, can be used to assist the user in finding a suitable transfer function. It offers an alternative and condensed way of examining the volume of data that reveals global characteristics which can be very helpful in creating a transfer function.

The semi-automatic method of Kindlmann[10] is a highly-regarded technique for generating transfer functions from volumetric data. This method makes the reasonable assumption that the features of interest in the data are the boundary regions between areas of relatively homogeneous material.

Bajaj's and Kindlmann's methods are designed for creating transfer functions with minimal or no user-interaction. According to our references, they do not address the problem of providing meaningful feedback during manual transfer function specification or fine-tuning. Because our method focuses on providing meaningful and fast feedback, it does not replace these schemes, but could be used very profitably in augmenting them.

The design galleries approach, trial-and-error scheme and their derivatives can be considered as being focused on providing feed-back in a user-driven transfer function specification process and this is where our scheme excels.

Even if interactive volume rendering facilities are available that can cope with continuous changes in the transfer function, it is often difficult to identify the fuzzy structures in a feedback volume rendering that can result from a particular transfer function and accurately relate them to the structures in the data that are being rendered. This makes it very difficult to quantify the impact that a small modification to the transfer function has.

It is desirable to have some kind of directed search process where the user is gradually but purposefully moving towards an optimal transfer function. The difficulty in quantifying the impact of a transfer function change, especially with regards to structures visible in the unprocessed data, hinders this process and necessitates a high level of visualisation experience from the user. Our method yields explicit and real-time feedback on the relationship between the structures in the data that are being rendered and the structures that can be expected in the resultant volume rendering.

3. The Data-Registered Interactive Previewing Technique

The central idea of this technique is to predict the optical result of casting a ray through a voxel and parallel to the view direction, through *all* the material represented by that particular voxel. The term "material" is defined as the set of optical properties assigned to a given scalar value by the DVR transfer function.

The amount of a specific material intersected by a ray can be estimated for each voxel position by making use of fast run-time processing of pre-calculated per-ray scalar frequency distributions (i.e. histograms of scalar values). The estimated amount, measured as a number of voxels, can be used in a simplified form of the volume-rendering integral to predict the optical outcome.

This prediction is done for all voxels in a particular slice of the raw data and all predicted results are then alpha-blended with that slice.

What the user sees is an estimate of what the volume rendering would look like with her currently configured transfer functions, super-imposed on a grey-scale slice image of the raw data. When the user makes any changes to the transfer

functions or when she "moves" (by changing the slice position) through the data, the visual update can be done in real-time, as relatively little processing is required.

This method is based on the assumption that the data consists of a finite number of contiguous volumes of optically homogeneous material, as is often the case with medical datasets. The closer the truth is to this assumption, the more accurate the preview is.

In the following sub-sections, we will first derive the simplified form of the volume-rendering integral, then explain the method we use to predict the expected amount of homogeneous material in the path of a ray and finally show how this has been implemented.

3.1. Mathematical Preliminaries

We start with the well-known absorption and emission model volume-rendering integral[11,12]. The light intensity at the eye (position D) is found by integrating from $s = 0$ at the opposite side of the volume that is being rendered to $s = D$ at the eye:

$$I(D) = I_0 e^{-\int_0^D \tau(t)dt} + \int_0^D C(s)\tau(s)e^{-\int_s^D \tau(t)dt}ds \qquad (1)$$

where I_0 is the light intensity at the opposite side of the volume, $\tau(x)$ is the extinction coefficient ("material density") at position x and $C(x)$ is the emitted colour at position x.

This relation models absorption (first term) and emission (second term). In the absorption term, already present light is attenuated by the density of the volume material obscuring it from the eye. In the emission term, light reflected and/or emitted by the material at any point is weighted by the material's density at that point and then attenuated by the density of all material between that point and the eye.

The emitted and/or reflected light $C(s)$ is simulated with the Phong shading model[13]:

$$\begin{aligned} I_\lambda(s) \;=\; & I_{a\lambda}k_a O_{d\lambda}(s) + \\ & I_{p\lambda}[k_d O_{d\lambda}(s)\,(\mathbf{N}(\mathbf{s})\cdot\mathbf{L}) + \\ & k_s\,(\mathbf{N}(\mathbf{s})\cdot\mathbf{H})^n] \end{aligned} \qquad (2)$$

The λ subscript denotes wavelength dependence so that we are not limited to a specific colour model. For example, in the case of RGB-raycasting, we will calculate respectively $I_R(s)$, $I_G(s)$ and $I_B(s)$ and these would then represent $C(s)$.

The symbols in equation 2 are as follows:

$I_{a\lambda}$	ambient light intensity
k_a	ambient reflection coefficient
$O_{d\lambda}(s)$	object diffuse colour at s
$I_{p\lambda}$	point light source intensity
k_d	diffuse reflection coefficient
$\mathbf{N}(\mathbf{s})$	surface normal at s
$\mathbf{L}$	direction of light source
k_s	specular reflection coefficient
$O_{s\lambda}(s)$	object specular colour at s
$\mathbf{H}$	vector halfway between view and light source vectors
n	specular reflection exponent

We make use of this model in our direct volume rendering preview to determine $C(s)$ (as it is in the raycasting implementation), but we refer the reader to the textbooks for a detailed discussion of Phong shading.

It is desirable to be able to evaluate the volume rendering integral numerically for practical application. Discretising equation 1 with distance step Δs yields:

$$I(D) = I_0 \prod_{i=1}^{D/\Delta s} e^{-\tau(s_i)\Delta s} + \sum_{i=1}^{D/\Delta s} C(s_i)\tau(s_i)\Delta s \prod_{j=i+1}^{D/\Delta s} e^{-\tau(s_j)\Delta s}$$

Without loss of generality, we can substitute $\tau(s_i)\Delta s$ with the opacity of the ith segment along the ray, i.e. $\alpha(s_i)$ and reverse the order of summation and multiplication (i.e. $s = 0$ at the eye and $s = D$ at the opposite side of the volume):

$$I(0) = I_D \prod_{i=0}^{D/\Delta s-1} e^{-\alpha(s_i)} + \sum_{i=0}^{D/\Delta s-1} C(s_i)\alpha(s_i) \prod_{j=0}^{i-1} e^{-\alpha(s_j)} \qquad (3)$$

Usually a two-term Taylor series approximation of the exponential terms, $e^{-\alpha(s_j)} \approx 1 - \alpha(s_j)$, is employed to yield a computationally efficient form that can be incrementally calculated (an exponential term can be updated with a single multiplication at each discretisation step) as a ray is being traversed. However, we will deviate from the usual derivation and continue with the more exact discrete form in equation 3.

If a ray were to pass through a known number N of voxels such that the voxels have constant object diffuse colour $O_{d\lambda k}$ and they have constant opacity α_k, i.e. the N voxels are optically identical, and we assume that

1. the ray sampling distance Δs is of the same dimension as a voxel, i.e. $D/\Delta s = N$ and
2. surface normals are constant for the N voxels

then equation 2 would yield constant C_k and we could simplify equation 3 to yield a closed-form solution:

$$
\begin{aligned}
I(0) &= I_D \prod_{i=0}^{N-1} e^{-\alpha_k} + \sum_{i=0}^{N-1} C_k \alpha_k \prod_{j=0}^{i-1} e^{-\alpha_k} \\
&= I_D e^{\sum_{i=0}^{N-1} -\alpha_k} + C_k \alpha_k \sum_{i=0}^{N-1} e^{\sum_{j=0}^{i-1} -\alpha_k} \\
&= I_D e^{-N\alpha_k} + C_k \alpha_k \sum_{i=0}^{N-1} e^{-i\alpha_k} \\
&= I_D e^{-N\alpha_k} + C_k \alpha_k \frac{1 - e^{-N\alpha_k}}{1 - e^{-\alpha_k}} \qquad (4)
\end{aligned}
$$

3.2. Algorithm

An orthogonal previewing direction is selected by the user. For each voxel position in the plane normal to the viewing direction, a frequency distribution of voxel scalar values is calculated for all voxels intersected by a ray through that voxel position and parallel to the viewing direction. In our case, the frequency distribution is a listing of the frequency of occurrence of voxel scalar values as a function of ranges (classes) of scalar values. The number of constant-size ranges (classes) is chosen so that the average number of voxels per class remains small (e.g. by choosing the number of classes to be a fifth of the number of voxels in the orthogonal previewing direction, we can expect an *average* of five voxels per class) and the classes are maintained in a sorted order determined by the scalar values at their limits. A gradient volume is also calculated. All of this is done once per previewing session.

Whenever a new super-imposable preview slice has to be generated (i.e. when the user modifies a transfer function or selects another slice position), the following procedure is executed: For each voxel in the current view-direction-orthogonal slice, its scalar value is transformed with the current transfer functions. The frequency distribution corresponding to its position is selected and the class to which this voxel belongs is found with a binary search.

Recall that these classes have been chosen to contain on average a small number of voxels. In addition, these are classes of *scalar* voxel values, not optical properties. However, by merging a group of neighbouring classes if the optical properties that they represent are equal (or very similar) we can obtain a much better estimate of the number of voxels in the path of the ray that have this set of optical properties. The merging is performed as follows:

The class that we have just found is merged with its neighbouring class to form a larger class *if* the value corresponding to the farthest edge of the neighbouring class, after having been transformed by the current transfer functions, is optically equal (or very similar) to the current voxel's transformation. Four-dimensional Euclidean distance in the colour-opacity space (in our case hue, saturation, value and scalar opacity) is used as a metric for optical similarity. This merging is continued until a neighbouring class's edge is optically distinct from the current voxel's transformation. Merging is performed in both directions.

After merging, we have an estimate of how many adjacent (spatially as well as in a scalar sense) voxels, with optical characteristics equal (or very similar) to our selected voxel, we can expect to be intersected by a cast ray that passes through the current voxel position and is parallel to the viewing direction. We will use this estimate as N in equation 4 to calculate the shaded light intensity that would result from a ray cast through just this optical material. As explained in section 3.1, we use the normal of the selected voxel to compute the shading for the ray that passes through it.

After having done this for each voxel in the current slice, the resulting image can be super-imposed on a slice of a grey-scale image of the original data by using $1 - e^{-N\alpha_k}$ as the opacity (representing absorption) and the second term of equation 4 as the colour (representing emission and reflection).

To summarise: Whenever the slice is changed or the transfer function is modified, each voxel in the current slice is transformed, bin merging is performed on the frequency distribution corresponding to its position and the predicted optical characteristics for that position are calculated. It is obvious that the processing overhead is highly dependent on the amount of merging that has to be done as the rest of the computations can be performed in logarithmic time.

Storage requirements are modest when the algorithm has access to the data-structures required by the final raycasting itself, for example the gradient volume. In our implementation, the previewing requires, per previewing direction, a fifth of the data-set size for the frequency distributions. This requirement is dependent on the number of frequency distribution classes that an implementation deploys.

3.3. Implementation

The DSCAS1 platform[14] was used to create a test implementation of the described method as it offers the necessary data structures and communication and rendering functionality.

By using the GUI, the user can set up objects for the data itself, the transfer function, the slice-viewer and the DVR previewer. After having connected these objects the necessary pre-processing will be done and the user can start manipulating the transfer function by interacting with hue, saturation, value, scalar opacity and gradient opacity functions by making use the UI-element shown in figure 1. UI-elements for adjusting all shading parameters are also supplied.

Throughout this process, the UI remains completely interactive and the DVR preview is updated in real-time. Once a satisfactory preview has been created the data can be volume rendered with the resultant transfer function.

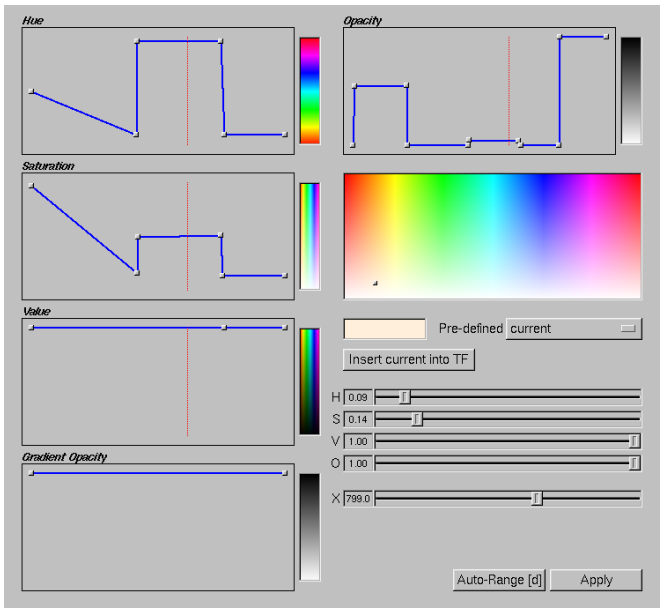

Figure 1: *Our implementation of a DVR transfer function editing widget. The user can add and manipulate an arbitrary number of points in the piecewise continuous lines that represent the hue, saturation, value, scalar opacity and gradient opacity transfer functions. Our method does of course extend to other more complex transfer function representations, e.g. Bézier curves.*

4. Results

We have applied the previewing scheme to make transfer functions for volume rendering a clinical CT-dataset of a shoulder as well as the tooth and sheep datasets used in the Transfer Function Bake-Off[3].

To appreciate the efficacy of this method one actually needs to interact with it or view a video recording which displays the interactivity of the preview. However, in the following figures one can clearly see the similarity between the preview and the raycasting resulting from the same transfer function as well as the subjective quality of the resultant raycasting. Keep in mind that the preview is updated in real-time as the user interacts with the transfer function and moves through the data. It can be viewed overlaid on a grey-scale image of the original data as well as by itself. Due to the flexible nature of the implementation, an arbitrary number of preview slice positions can be viewed from all orthogonal view directions simultaneously, overlaid and non-overlaid.

Figure 2 (see colour section) shows a preview and volume rendering of clinical CT-data of a shoulder. Figure 3 (see colour section) shows (from left to right) a preview with shading, a preview without shading (i.e. equation 2 with $I_{a\lambda} = 1$, $k_a = 1$ and $k_d = k_s = 0$) and a volume rendering of industrial CT-data of a tooth. Figure 4 (see colour section) shows (from left to right) a preview with shading, a preview without shading and a volume rendering of MRI-data of a sheep heart.

The data can be investigated with a point probe to get some idea of the different scalar values involved. By clicking on any point in any slice the scalar voxel value at that point is returned. In all cases, setting up an initial transfer function and then converging on a "good" transfer function takes less than ten minutes of session time.

5. Conclusions

We have presented a volume rendering preview method that yields real-time slice-based predictions of a raycasting with a given dataset and transfer function. This method, which is based on a the volume rendering integral and the Phong shading model, is used in a framework where the user manipulates the volume rendering transfer function and the slice-based preview is updated in real time. The preview is usually shown super-imposed on the corresponding slice of a grey-scale image of the "raw" data. The user is also allowed to continue investigating the results by continually changing the current slice position, i.e. "moving" through the data.

This scheme has some advantages which differentiates it from the other feedback-based methods:

- The preview is sliced-based and data-registered with the data that is being rendered, so it can be overlaid on a grey-scale image of the data and the user can easily relate structures in the volume rendering to structures in the "raw" data. Because of this, the fidelity of the resulting volume rendering can easily be checked and the (clinical) user's

experience in working with slice-based medical data is leveraged.
- The preview is very fast so the effect of small changes are instantly fed-back to the user. This aids tremendously in effectively converging on a good transfer function.
- The method is easy to implement.
- The method does not require special hardware.

The method that we have developed gives a very good data-registered preview of a potential volume rendering and is invaluable in the transfer function specification process. However, there is definitely room for improvement in the way in which the user edits a transfer function (and this is so in most current volume rendering applications). We will investigate novel and different approaches to this interaction problem.

Acknowledgements

This research is part of the DIPEX (Development of Improved endo-Prostheses for the upper EXtremities) program of the Delft Interfaculty Research Center on Medical Engineering (DIOC-9).
The clinical CT-images of the shoulder were provided by R.E. van Gelder of the Academic Medical Center of Amsterdam.

References

1. M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, pp. 29–37, May 1988.
2. M. Meissner, J. Huang, D. Bartz, K. Mueller, and R. Crawfis, "A Practical Evaluation of Popular Volume Rendering Algorithms," in *Proc. Volume Visualization and Graphics Symposium*, pp. 81–90, 2000.
3. H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, K. Raghu, R. Machiraju, and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics and Applications*, vol. 21, pp. 16–22, Jan-Feb 2001.
4. J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber, "Design galleries: a general approach to setting parameters for computer graphics and animation," in *ACM SIGGRAPH Conference on Computer Graphics*, pp. 389–400, 1997.
5. T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of transfer functions with stochastic search techniques," in *Visualization '96*, pp. 227–234, 489, IEEE, 1996.
6. P. Lacroute and M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," in *Proc. SIGGRAPH '94*, pp. 451–458, July 1994.
7. A. H. König and E. M. Gröller, "Mastering Transfer Function Specification by using VolumePro Technology," in *Proceedings of the 17th Spring Conference on Computer Graphics (SCCG)*, 2001.
8. C. P. Botha and F. H. Post, "New technique for transfer function specification in direct volume rendering using real-time visual feedback," in *Proceedings of the SPIE International Symposium on Medical Imaging* (S. K. Mun, ed.), vol. 4681 - Visualization, Image-Guided Procedures, and Display, 2002. To appear.
9. C.Bajaj, V.Pascucci, and D.Schikore, "The Contour Spectrum," in *Visualization '97*, pp. 167–173, IEEE, 1997.
10. G. Kindlmann and J. W. Durkin, "Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering," in *IEEE Symposium on Volume Visualization*, pp. 79–86, 1998.
11. N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 99–108, June 1995.
12. W. Krueger, "The application of transport theory to visualization of 3D scalar data fields," in *Proceedings of Visualization '90*, pp. 273–280, 481–2, IEEE, 1990.
13. B. T. Phong, "Illumination for Computer Generated Pictures," *Communications of the ACM*, vol. 18, pp. 311–317, June 1975.
14. C. P. Botha and F. H. Post, "A Visualisation Platform for Shoulder Replacement Surgery," in *Interactive Medical Image Visualization and Analysis (IMIVA) - Satellite Workshop of MICCAI 2001* (S. D. Olabarriaga, W. J. Niessen, and F. Gerritsen, eds.), pp. 61–64, October 2001.

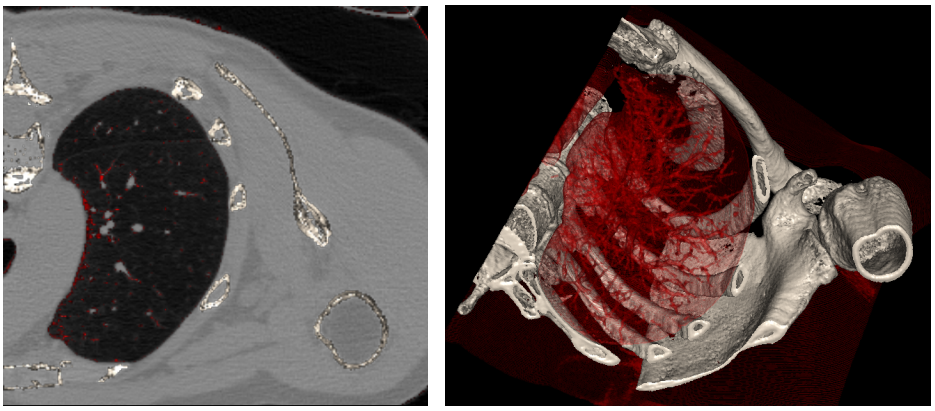

Figure 2: *Preview and direct volume rendering of shoulder CT-data showing bony and bronchial structures.*

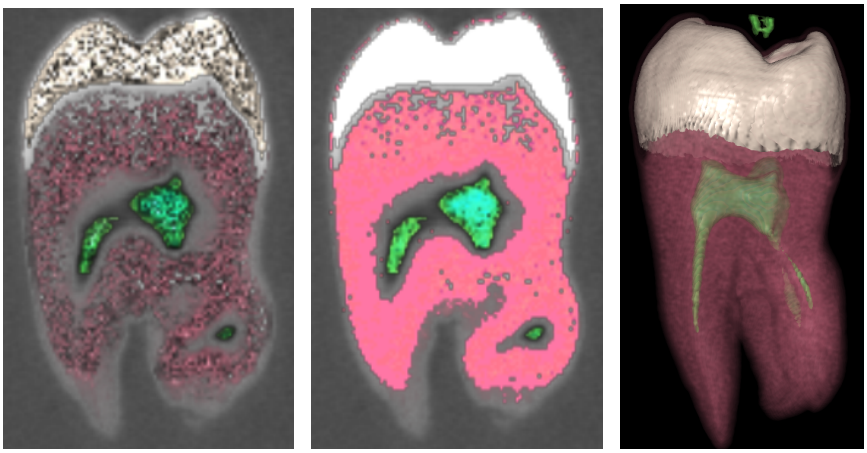

Figure 3: *Previews with and without shading and direct volume rendering of industrial CT-data of tooth.*

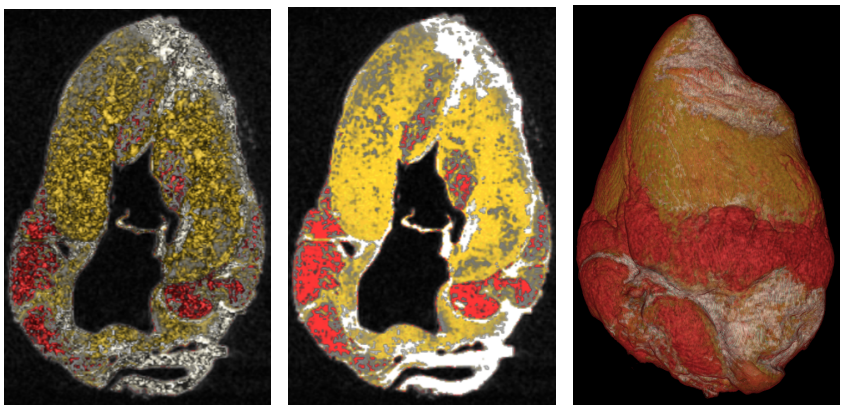

Figure 4: *Previews with and without shading and direct volume rendering of MRI-data of sheep heart.*