Boundary Aware Reconstruction of Scalar Fields

Stefan Lindholm *Student Member, IEEE*, Daniel Jönsson, *Student Member, IEEE*,
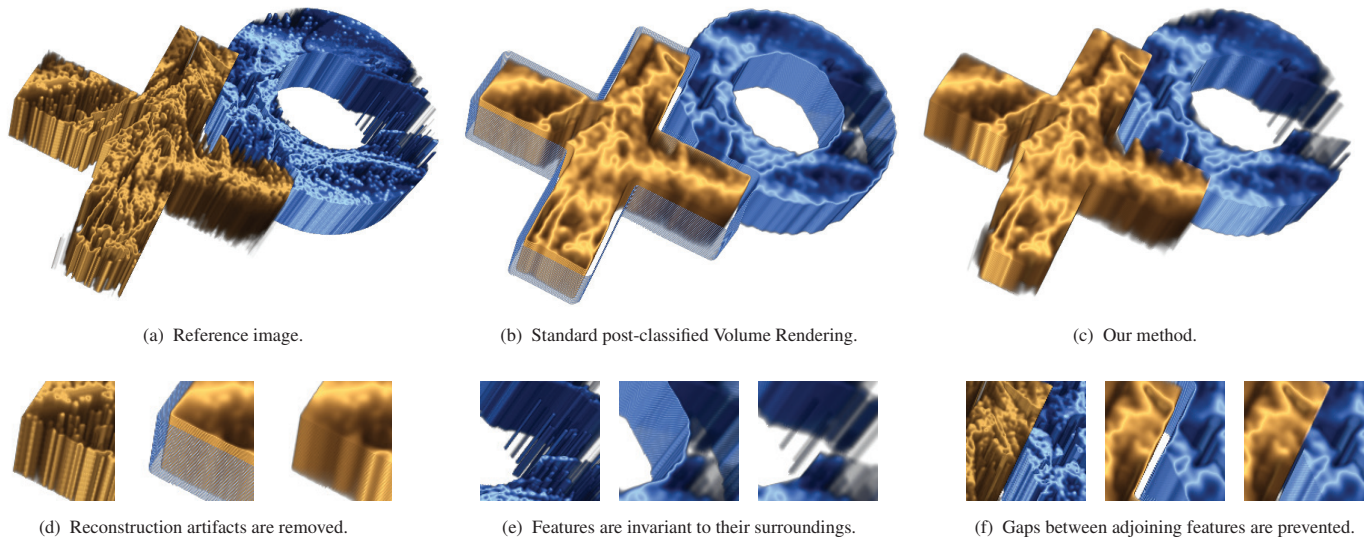Charles Hansen, *Fellow, IEEE*, and Anders Ynnerman, *Associate Member, IEEE*

(a) Reference image.

(b) Standard post-classified Volume Rendering.

(c) Our method.

(d) Reconstruction artifacts are removed.

(e) Features are invariant to their surroundings.

(f) Gaps between adjoining features are prevented.

Fig. 1: Continuity between data points is commonly assumed in visualization—yet our understanding of many objects is that they consist of disjunct features with distinct boundaries. The data set used in this figure was created specifically to highlight and categorize three types of misrepresentations that often arise in relation to boundary areas. In this paper, we present an approach that takes feature boundaries into account during reconstruction and improves the visual representation in transition regions.

Abstract—In visualization, the combined role of data reconstruction and its classification plays a crucial role. In this paper we propose a novel approach that improves classification of different materials and their boundaries by combining information from the classifiers at the reconstruction stage. Our approach estimates the targeted materials' local support before performing multiple material-specific reconstructions that prevent much of the misclassification traditionally associated with transitional regions and transfer function (TF) design. With respect to previously published methods our approach offers a number of improvements and advantages. For one, it does not rely on TFs acting on derivative expressions, therefore it is less sensitive to noisy data and the classification of a single material does not depend on specialized TF widgets or specifying regions in a multidimensional TF. Additionally, improved classification is attained without increasing TF dimensionality, which promotes scalability to multivariate data. These aspects are also key in maintaining low interaction complexity. The results are simple-to-achieve visualizations that better comply with the user's understanding of discrete features within the studied object.

Index Terms—Reconstruction, signal processing, kernel regression, volume rendering

◆

1 INTRODUCTION

Classification is one of the core steps in the visualization pipeline and is commonly applied either explicitly through dedicated off-line classifiers or implicitly through the visual mapping provided by, for example, a transfer function (TF). Yet, classification based on reconstructed data is never perfect and designing cost-efficient solutions to minimize misclassification is an active area of research.

It is common practice in visualization applications to reconstruct a fully continuous signal, i.e. that given two neighboring values the signal is assumed to pass through all other values that exist between. This practice comes with a significant risk that the reconstructed data values, after the visual mapping, end up representing materials at locations where they were not present in the studied object. As an example, consider a medical Computed Tomography (CT) data set in which the attenuation values in the data represent different materials. Bone and air are two materials that are far apart in the attenuation spectrum. Then let a volume rendering ray pass from air into bone. Under the assumption of continuity, the reconstruction will create values that correspond to all intermediate attenuation levels and thus take the appearance of skin, fat, muscles, contrast agent, and other tissues. Fig. 2 highlights this problem where the visual appearance of contrast-enhanced vessels, observed at 400–600 Hounsfield units (HU), is not confined to the vessels, but appear in every ray that passes through both air (less than -800 HU) and dense bone (greater than 850 HU).

Rendering of transitions between differentiable materials has been addressed through methods based on derivative expressions such as gradient magnitudes. We will show that those methods are less ideal in classifying individual materials and also inevitably lead to an increase of the dimensionality of the TF with one or two dimensions.

- *S. Lindholm, D. Jönsson, and A. Ynnerman are with the Department of Science and Technology, Linköping University. E-mail: {stefan.lindholm|anders.ynnerman}@liu.se.*
- *C. Hansen are with the Scientific Computing and Imaging Institute, University of Utah. E-mail: hansen@cs.utah.edu.*

This adds interaction complexity, prevents the use of inherently multi-variate TFs, and does not resolve the misclassification problem as well as the herein proposed method. We also show that the use of methods based on segmentation masks and labeled volume rendering leaves several issues unresolved.

Our approach to address the problem above is centered around the reconstruction step in the visualization pipeline. Given an *object of study* that consists of separate features our goal is to achieve continuous visualizations within each feature while representing discontinuous boundary regions between features. We thus change the objective of the reconstruction to no longer be restricted to the reconstruction of a single "optimal" signal, but rather to provide the overall best representation of the object of study based on prior knowledge of its features and their visualization.

The first step in our feature-constrained reconstruction method is to obtain a probabilistic classification (see Section 4) which is used to assign importance weights to the original grid points. We then reconstruct one sample value for each feature that is within the reconstruction kernel range using the importance values as reconstruction weights. We use a kernel regression framework that provides reconstruction validity along with reconstructed values (see Section 5). The reconstruction validity is used during the intermixing stage to limit the visual contributions from all features to their respective signals' spatial support as they are combined to a final visual output (see Section 6). As can be seen in Fig. 1 and Fig. 3, the output of our method resembles *post-classified* rendering within features and *pre-classified* rendering between features.

It should be noted that, in contrast to previous work, we represent features in discontinuous boundary regions through weighted extrapolation, not by explicitly modeling or classifying data behavior across these regions. This has several benefits. First, the classification of a single feature is less dependent on its surroundings, less dependent on well defined gradients, and also less sensitive to noise. Second, misclassification can be addressed without increasing the dimensionality of the TF, which allows us to maintain an easy-to-use user interface and also increases the scalability of our method to multivariate data such as Dual-Energy CT (DECT) data.

The main contributions of this paper can be summarized as:

* Presentation of a boundary aware approach to reconstruction of data that handles continuity within features as well as discontinuities in boundary regions.

* A novel extension of the kernel regression framework to handle reconstruction of multiple features using anisotropic kernels.

* Reduction of the impact of partial volume effects while maintaining high frequency details within features through adaptive kernel sizes.

* A simple-to-use approach that seamlessly combines probabilistic classification, adaptive reconstruction and visual mapping without increasing interaction complexity.

2   RELATED WORK

When the TF was introduced by Drebin *et al.* [5] it was done so with an explicitly probabilistic classification of samples. Bayesian estimates were used to yield material mixture percentages for each voxel, which subsequently were used to combine visual contributions from the participating material using 'accumulation level intermixing' as described by Cai and Sakas [3]. Our work uses the same level of intermixing to combine the materials, i.e. at each step along the ray, and a similar form of classification. However, we separate the signal reconstruction to each material and use a validity-based weighting to combine the material estimates, which solves many of the misclassification issues of the original approach.

Since the work of Drebin misclassification and partial volume problems have mainly been addressed in three ways. One category relies on the TF as a visual classifier which produces good visualizations but does not classify voxels. A second category aims to computationally
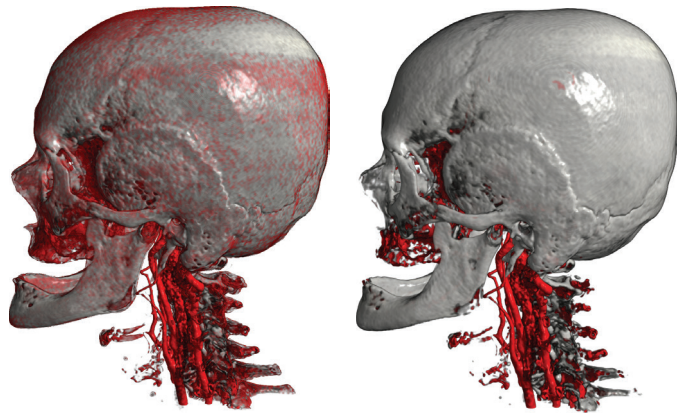

Fig. 2: This example highlights one of the problems that arise when data is assumed to be continuous and no prior knowledge about the scene is used in the rendering. *Left:* The result of standard DVR with continuous reconstruction. *Right:* The proposed method, which achieves continuous reconstruction within each feature while preventing interpolation between features. The red "sheet" visible in the left image is an artifact caused by the interpolation. The problem is made more difficult by the fact that the data is acquired through Dual-Energy CT and therefore is inherently multi-variate.

solve partial volume effects through statistical methods using models for all materials, or combinations of materials, that may exist in the data. A final category of works rely on explicit segmentation masks to limit the area-of-effect of the TF.

Visual classification: Gradients of the scalar field are used by Levoy [17] to better discriminate the material boundaries. Kindlmann and Durkin [11] introduce the boundary emphasis function that explicitly targets material boundaries. It is based on an abstraction using the scalar value, the first and the second derivatives to better visualize isosurfaces, i.e. material boundaries, through opacity mapping. This method is similar to ours in that a simplified user interface allowed ease of visual classification of the underlying volume. Kniss *et al.* [12] extend this idea to widget-based TF design. Their user interface is, however, more complex than our technique, and we also seek to classify individual voxels, not only creating visual representations of material boundaries. A TF method capable of identifying sheet like structures, e.g. material transitions, is proposed by Sato *et al.* [22]. The method improves classification for recognizable shapes but carries the same weaknesses as the derivative based methods due to the increase of the dimensionality of the classification space.

Statistical models: A comprehensive overview of decision theory and probabilistic models is given in the work by Bishop [1] and an overview on image processing methods by Gonzalez and Woods [6]. Probabilistic classification for visualization is summarized by Kaufman and Mueller [9] and its advantages are demonstrated in several publications, e.g. [8, 13, 14, 19, 21, 23]. Variations of the approach have mostly been applied to pre-segmented data or incorporated in off-line classification schemes with complete data models, i.e. all materials need to be known a priori. Laidlaw *et al.* [15] use a variation of this approach in an analysis of feature space designed to estimate percentages of material occupation inside voxels in Magnetic Resonance (MR) data. The method excels in situations where a complete segmentation of all materials is desired. A similar mixture model, specific for DECT, is proposed by Heinzl *et al.* [8], which potentially could be used in place of our interaction based classification. Prilepov *et al.* [19] propose an order dependent method for multi-material interface reconstruction based on such percentage data. Their work focus mainly on surface representations of binary decision boundaries where our work retains the probabilistic nature of the data throughout the rendering process. Kniss *et al.* [13] take a more visual approach by delaying the final classification decision until render time to facilitate uncertainty exploration and risk analysis. Their method rely on
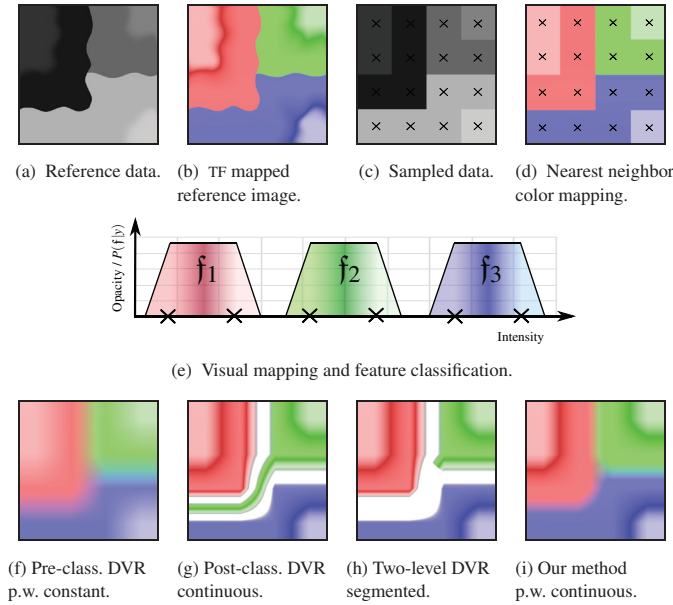
(a) Reference data. (b) TF mapped reference image. (c) Sampled data. (d) Nearest neighbor color mapping.

(e) Visual mapping and feature classification.

(f) Pre-class. DVR p.w. constant. (g) Post-class. DVR continuous. (h) Two-level DVR segmented. (i) Our method p.w. continuous.

Fig. 3: *Top row:* A piecewise continuous object of study (a–b) composed of three features ($\mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}_3$) is point-sampled to a 4×4 grid of raw data (c). Three widgets define the visual mappings and feature classifications (e). The mapping is also exemplified (d). *Bottom row:* Reconstruction of the object performed under the assumption of piecewise constant data (f), globally continuous data (g), segmented data (h), and piecewise continuous data (i).
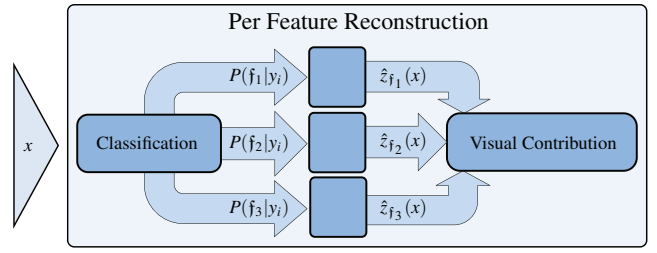

Fig. 4: Overview of the proposed method. In order to avoid artifacts stemming from incorrect interpolation across features, the processing is divided into feature-specific reconstruction operations (here $\mathfrak{f}_1, \mathfrak{f}_2, \mathfrak{f}_3$). Finally, the visual contributions from each reconstructed value are recombined in order to produce a visual representation of the data.

initial probabilistic expressions for each voxel and for all classes of materials in the data. Visual mapping is then performed over these classes instead of the raw data. An important difference between this method and ours is that theirs ultimately result in a piece-wise constant expression that does not allow for variations inside materials.

Explicit Segmentation: Explicit binary segmentation, or labeled-volume rendering, has been used to spatially constrain the TF response, thereby successfully reducing misclassification artifacts. Label volumes by Tiede *et al.* [25] and two-level volume rendering by Hadwiger *et al.* [7] are two examples that provide smooth, but binary, decision boundaries. The methods do not, however, account for signal degeneration in the surrounding regions and can therefore only be used to solve some of the classification artifacts described earlier. This is studied in more detail in Section 3. Also, despite extensive research progress, accurate segmentation masks are often not available unless time consuming manual procedures are used.

Kernel regression: A key part of our work is based on the kernel regression framework [24] (a generalization of, among other methods, bilateral filtering [26] and moving least squares [2]). A number of techniques for reconstruction based on such methods have been presented within the context of volume rendering, e.g. [16, 27]. These works exclusively assume continuity in the signals.

We chose to use the kernel regression framework since it offers an intuitive connection between the probabilistic classification, weighted reconstruction, and reconstruction certainty within a unified framework. Although kernel regression has traditionally not been used for discontinuous signals we will show how this can be done and how it can easily be integrated into existing frameworks. It should also be noted that the kernel regression framework supports unstructured grids even if this aspect is not utilized within this work.

3 DVR APPROACHES AND DISCONTINUITIES

To illustrate our approach in relation to the most commonly used Direct Volume Rendering (DVR) methods we provide a small synthetic point-sampled data set in Fig. 3. The reference is a piecewise continuous object of study consisting of three separate features, $\mathfrak{f}_{1,2,3}$ (red,

green and blue). The objective is to create a representation of the reference from sub-sampled data. Data and TF mapped images for both the reference and the sub-sampled data (4×4 sample points) used to create the representations are shown on the top row (a–d). The TF, which also acts as a classifier, is shown in (e). On the bottom row (f–i), four different results are shown: *pre-classified* DVR, *post-classified* DVR, *two-level* DVR, and *our method*.

As evidenced by the images each method produces different representations of both the boundaries and internal regions. *Pre-classified* DVR effectively treats intermediate space between samples as uncertain. This provides a representation of lost information in the boundary area, which expresses uncertainty by combining contributions from each of the neighboring samples. Unfortunately, this also prevents the variations within the materials from being represented (dark streaks). *Post-classified* DVR reconstructs the signal as fully continuous. This captures the variations within the materials well, but also creates both gaps (white areas) and sheet artifacts (e.g. the green line) in the boundary regions. *Two-level* DVR utilizes segmentation masks to spatially localize the TF widgets to their respective regions (post-classified DVR is used within each region). This successfully removes the sheet artifacts, but interpolation of the signal near boundary regions is still an issue, causing gaps in the rendering.

We have chosen to design our method such that variations within features are preserved, and a representation of uncertainty in boundary regions is obtained, avoiding the sharp decisions made by many existing approaches. The approach builds on systematic modifications of the classification, reconstruction and visual mapping stages of the DVR pipeline as shown in Fig. 4. In the following sections we present the details of these stages and how we can achieve the desired result without further input beyond the existing TF.

For the remainder of the paper, x and y denote spatial position and data value(s) respectively while index i indicates a grid point.

4 PROBABILISTIC CLASSIFICATION

This section explains two new ways to acquire feature classifiers $P(\mathfrak{f}_j \mid y_i)$, i.e. the probability that feature $\mathfrak{f}_j$ occurs given the data value at grid point i. A key aspect of the two new methods is that they use a widget-based graphical interface similar to a standard TF editor, which is highly desirable thanks to the low latency between editing and visual result. In the first method each widget provides both classification and visual properties (illustrated in Fig. 5(a)) and in the second the two qualities are separated into individual widgets (illustrated in Fig. 5(b)). When acquiring both visual properties and classification from the same widget we derive the classification for feature $\mathfrak{f}_j$ from the widget's alpha channel. To further decouple the classification from the visual mapping we normalize the response such that the classifier returns 1 across the maximum of the widget:

$$\widetilde{P}(\mathfrak{f}_j \mid y_i) = \sqrt{\frac{\alpha(y_i)}{\alpha_{\max}}},$$

(a) Combined Editor.
Feature classifiers are extracted automatically.

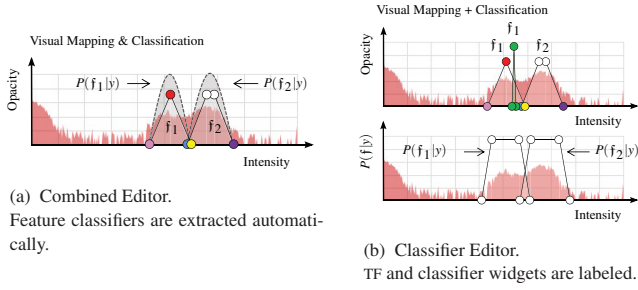(b) Classifier Editor.
TF and classifier widgets are labeled.

Fig. 5: Feature classifiers can be designed as separate primitives (for full flexibility) or extracted from the visual primitives (for minimal interaction complexity).

where α_{max} is the maximum opacity of the widget associated with feature $\mathfrak{f}_j$. In our experience, this approach often provides sufficient feature delineation (Fig. 9 and 10 are good examples), is especially easy to implement and does not require any additions to the user interface.

Unfortunately, the above approach imparts a set of restrictions on widget design. The most obvious such restriction occurs when a classifier is meant to capture an entire feature while the visual mapping is only meant to display a smaller part of the full feature. An example of such a case is illustrated in Fig. 5(b) where the green widget represents a thin range of values and thus would be too narrow to classify the full feature.

Using dedicated classifier widgets increases the flexibility of the system at the cost of interaction complexity due to the overall increased number of widgets. Classifier widgets and visual property widgets also need to be explicitly linked in an additional step. As the interpretation for a classifier widget is strictly probabilistic it holds only a scalar value. Fig. 5(b) illustrates a setup with two stacked editors. Specificity, interaction complexity and preset capabilities follows that of a standard TF.

Relying on the user to provide classification knowledge to the system through interaction often leaves the probabilistic model incomplete. To address this issue we use an approach outlined by Lundström *et al.* [18] based on a *null* class to make the final model representative for all data. In short, the user provided probabilities are first normalized before the *null* class is created

$$P(\mathfrak{f}_j \mid y_i) = \frac{\widetilde{P}(\mathfrak{f}_j \mid y_i)}{\max_i \sum_j \widetilde{P}(\mathfrak{f}_j \mid y_i)} \qquad (1)$$

$$P(\mathfrak{f}_0 \mid y_i) = 1 - \sum_j P(\mathfrak{f}_j \mid y_i). \qquad (2)$$

While the methods presented above are the only forms of classification considered in this work other sources can also be applicable. For example, off-line classifiers such as those employed by Kniss *et al.* [13] and Saad *et al.* [21] or modality specific methods such as those employed by Laidlaw *et al.* [15] or Heinzl *et al.* [8] can also be considered.

5 A FRAMEWORK FOR RECONSTRUCTION OF PIECEWISE CONTINUOUS DATA

Our reconstruction framework is based on kernel regression but extends it to handle multiple features. We will therefore briefly explain kernel regression before detailing how it is combined with our feature based approach. Furthermore, to handle partial volume effects, we will describe how to adaptively choose the filter kernel size. We simplify the notation of a single feature $\mathfrak{f}_j$ by omitting the index j to improve the presentation.

5.1 Extending Kernel Regression to Feature-Constrained Reconstruction

This section details how the kernel regression framework is used to achieve continuous signal reconstruction within individual features.

The original kernel regression framework was presented for image processing by Takeda *et al.* [24] to provide optimal estimations of an unknown signal between sampled grid points. The framework describes how a series of constraints can be used to influence a reconstruction. The framework also supports non-regular grids as well as variations in the local expansions. We use their notation, where y_i denote the values of the grid points and the reconstructed signal estimate is denoted $\hat{z}(x)$.

To preserve the details within a feature but discern between different features, the kernel regression framework in our approach relies on local approximations by Taylor expansion and is thus not defined for reconstructions across discontinuities. Our assumption here is that the Taylor expansion is valid within each feature and that the reconstruction will therefore be valid as long as it is performed within the continuity of a single feature. Furthermore, weight functions are often enforced to be strictly greater than zero over their entire domain to avoid the risk of undefined reconstructions. We can relax this requirement such that weight functions may be zero since the utilization of the kernel regression framework will ensure that the visual impact goes to zero when the reconstruction becomes undefined.

Weighted Reconstruction

For reference, the reconstruction with a single normalized spatial kernel takes the following form

$$\hat{z}(x) = \sum_{y_i \in \Omega_K} w_i y_i, \qquad (3)$$

where w_i are the spatial weights of the neighboring samples and the neighborhood Ω_K is defined by the kernel size.

Feature-Constrained Reconstruction

In this work, the spatial kernel is given by a generalization of trilinear interpolation with separable weights computed in each dimension for all spatial weights:

$$w_i = w_X w_Y w_Z, \text{ with } w_{X|Y|Z} = 1 - \frac{|x - x_i|_{X|Y|Z}}{2\sigma}. \qquad (4)$$

Here, w_X, w_Y and w_Z are the respective weights in each spatial dimension and w_i denotes the final spatial weight for grid point i. The function is scaled such that the tent shape approximates a Gaussian which is truncated at $\pm 2\sigma$.

Constructing a feature-specific weight function for kernel regression is straightforward once the conditional probabilities $P(\mathfrak{f} \mid y_i)$ are known. Using the framework notation, a feature weight is defined as

$$w_{\mathfrak{f}} = P(\mathfrak{f} \mid y_i). \qquad (5)$$

This is combined with the spatial weight function from Equation 4 to form the feature specific reconstruction for feature $\mathfrak{f}$

$$\hat{z}_{\mathfrak{f}}(x) = \frac{\sum_{y_i \in \Omega_K} w_i w_{\mathfrak{f}} y_i}{\sum_{y_i \in \Omega_K} w_i w_{\mathfrak{f}}}. \qquad (6)$$

Assuming a regular grid, normalized spatial kernels will always have support and integrate to one. The feature kernels on the other hand cannot be assumed to do so. To address this we will introduce the concept of reconstruction validity during the visual mapping stage described in Section 6.

Extending the framework to multiple features is done by computing one weighted reconstruction per feature of interest defined by the user. That is, Equation 6 is evaluated for each $\mathfrak{f}_j$. The result is a set of estimated values $\hat{z}_{\mathfrak{f}_j}$ (one for each feature of interest), which are processed in the visual mapping stage described in Section 6.

5.2 Partial Volume and Kernel Size Selection

As our data model we assume an object of study to consist of a series of continuous features (independent signals) divided by discontinuities (step-functions). In case the data has been convolved with some form of point spread function (PSF) causing partial volume effects the PSF is modeled as a Gaussian with an area of influence of σ_{PSF} in accordance with Kindlmann and Durkin [11]. By using the fact that the influenced regions are restricted in the spatial domain and that their extent is often a global property that is independent of the participating materials [10, 20] we can calculate a kernel size that captures the effect of partial volume. We observe that, for a kernel with a size equal to the PSF, the combined weight of all voxels that have been misclassified due to partial volume is less than the combined weight of a solid feature. However, using a large kernel across the whole domain can suppress details in high frequency areas. Therefore, to enable our reconstruction to handle partial volume effects and at the same time preserve details in high frequency data, we utilize a variable kernel size. The maximum kernel size is based on the PSF, $\sigma_{max} = 2\sigma_{PSF}$, which also serves as an initial estimate. In order to reduce the kernel size in regions dominated by solid features we first introduce a measure of kernel support:

$$\Phi(x, \mathfrak{f}, \Omega) = \sum_{y_i \in \Omega} w_i w_{\mathfrak{f}}. \qquad (7)$$

Note that Ω is directly related to the σ used in Equation 4. In particular, we use the term *feature support* for an estimation within a neighborhood whose size is related to the observed PSF:

$$\mathcal{F}_{\mathfrak{f}} = \Phi(x, \mathfrak{f}, \Omega_{max}). \qquad (8)$$

This feature support is used to differentiate between real features and partial volume misclassifications. The variable kernel size is computed as a linear combination of the minimum (σ_{min}) and maximum (σ_{max}) kernels based on the feature support from Equation 8:

$$\sigma_{\mathfrak{f}} = (\mathcal{F}_{\mathfrak{f}})\sigma_{min} + (1 - \mathcal{F}_{\mathfrak{f}})\sigma_{max}, \qquad (9)$$

where we use $\sigma_{min} = 0.25$ for a smallest possible kernel size of 2^3 voxels. If a feature $\mathfrak{f}$ has no support, i.e. $\mathcal{F}_{\mathfrak{f}} = 0$, the reconstruction of the feature is terminated since no visual output will be generated due to a lack of valid samples. Note that the size of the reconstruction kernel does not need to be adjusted for point-sampled data without partial volume.

6 VISUAL MAPPING

A widget-based TF is used to assign color and opacity properties to primitives. Section 4 presented how each TF widget is explicitly or implicitly linked to a feature. The TF here expresses the user's interest in a particular part of a feature and more than one widget can be assigned to the same feature. We will now go through how the multiple visual contributions are combined into a joint visual response.

6.1 Reconstruction Validity

Having reconstructed an estimated value for a specific feature we want to know how valid this value is to use in the latter parts of the pipeline.

As a measure of validity we use the support of active reconstruction kernel. Including the kernel validity is necessary in order to smoothly avoid ill-posed reconstructions. For example, consider a region where $P(\mathfrak{f} \mid y_i) \approx 0$ for a majority of samples in Ω_K, i.e. almost no values were valid to use. This may be caused by situations where only a few far-away voxels have an impact on the reconstructed value. The kernel regression framework still provides the best possible estimate from a least squares' perspective, but it will do so using a strong normalization as the denominator in Equation 6 will be very small. To address this issue, we interpret the strength of this normalization as a measure of kernel validity

$$\mathcal{V}_{\mathfrak{f}} = \Phi(x, \mathfrak{f}, \Omega_K). \qquad (10)$$

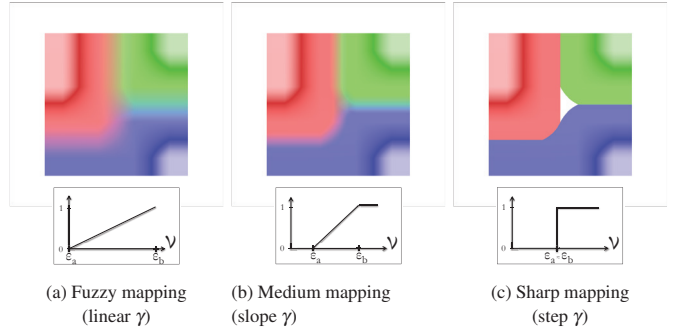| (a) Fuzzy mapping (linear γ) | (b) Medium mapping (slope γ) | (c) Sharp mapping (step γ) |

Fig. 6: The mapping function determines how sharp the transition boundary should be between different features. The presented images in this work use a mapping similar to (b) unless otherwise specified.

Note that the kernel validity differs from the feature support in that it is computed on the potentially smaller neighborhood Ω_K using $\sigma_{\mathfrak{f}}$ for spatial weights (cf. Equation 8).

This is similar to the interpretation as a reconstructed conditional probability $\widetilde{P}(\mathfrak{f} \mid x)$ at the point of reconstruction as proposed by Kniss *et al.* [13]. Due to its use as a normalization factor in the reconstruction, we prefer the interpretation of the expression as reconstruction validity within the context of this work.

6.2 Combining Multiple Visual Contributions

The maximum number of visual contributions that need to be combined is equal to the number of widgets in the TF. In practice, however, only one or two of the contributions will typically have non-zero opacity for a given data value (i.e, there are few places where more than two visible features overlap).

As an example, consider the visual editor in Fig. 5(b) in which two features ($\mathfrak{f}_1$ and $\mathfrak{f}_2$) and three visual primitives, TF_p (red, green and white) have been defined. The red and green primitives are associated with $\mathfrak{f}_1$, the white primitive with $\mathfrak{f}_2$. Computations at any point x along the ray would then include two reconstructions and result in three visual contributions, which all need to be taken into consideration for the final visual response at point x.

Previous works have targeted the problem of combining multiple, uncertain or partial visual contributions. Cai and Sakas [3] provide a categorization of intermixing schemes. Other approaches include solutions based on decision- and risk-boundaries, fuzzy representation and probabilistic animation [4, 13, 18]. It is, however, outside of the scope of this paper to provide a complete overview of these approaches.

In our work, we employ an accumulation level intermixing [3] where color contributions from a single sample point are computed individually. Each contribution is additionally weighted by its reconstruction validity, $\mathcal{V}_{\mathfrak{f}}$, to separate between desired and undesired contributions. For our purposes we have chosen a fuzzy representation of uncertainty based on reconstruction validity and user interest as expressed in the alpha channel of the TF. We first add a linear mapping function to the computed validity

$$\gamma(\mathcal{V}) = \frac{\mathcal{V} - \varepsilon_a}{\varepsilon_b - \varepsilon_a}, \text{ with } \gamma \text{ clamped to } [0, 1]. \qquad (11)$$

The visual contributions for each feature are subsequently blended into the accumulated color, C_{dst}, and opacity, α_{dst}, using front-to-back compositing:

$$\begin{aligned} C_{dst} &= C_{dst} + (1 - \alpha_{dst})C_{src} \\ \alpha_{dst} &= \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src}\gamma(\mathcal{V}_{\mathfrak{f}}), \end{aligned} \qquad (12)$$

where C_{src} and α_{src} are the color and opacity from the mapping of the reconstructed value $\hat{z}_{\mathfrak{f}}$ through each associated primitive, $TF_p(\hat{z}_{\mathfrak{f}})$. The compositing in Equation 12 is thus performed once for each widget. The blending order corresponds to the order of the widgets in the TF.

(a) Intensity   (b) Intensity/gradient   (c) Intensity/gradient   (d) Intensity/gradient   (e) Ours
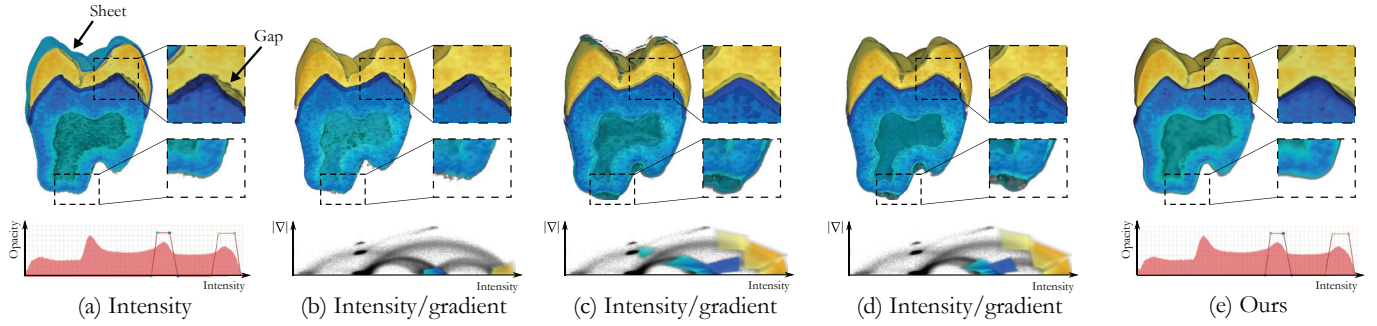
Fig. 7: Comparison of reconstruction for intensity TF, 2D intensity/gradient TF, and our method using the Tooth data set. (a) 1D TF introduces misclassified boundaries (blue sheet) and large empty space near features boundaries (top magnification). (b–d) 2D intensity/gradient TF with minimal to high user interaction, illustrating the difficulty of achieving a desired classification through complex interfaces. None of them manages to perform the desired classification. (e) Our method applied to the 1D TF correctly classifies the materials and handles empty space near feature boundaries, as shown in the top and bottom magnification, with minimal additional user input.

Appropriate values for ε_a and ε_b naturally depend on the characteristics of the data. For point sampled data without partial volume the mapping is reduced to a stylistic expression of certainty, as exemplified in Fig. 6, ranging from full uncertainty visualization to a best-guess representation. All renderings based on point sampled data in this paper use $\varepsilon_a = 0.25$ and $\varepsilon_b = 0.75$. For imaging data with partial volume, the less precise classification often lowers the expected reconstruction validity. For this type of data, we use $\varepsilon_a = 0.25$ and $\varepsilon_b = 0.5$ for all renderings, which suppress most partial volume effects while also providing full opacity within features.

For non-expert users, ε_a and ε_b can remain hidden. The only parameter that we then choose to expose to the user is the option to manually fine-tune σ_{max} in case the initial estimate proves incorrect. The interpretation of the parameter is straightforward as a 'high frequencies' vs. 'partial volume artifacts' tradeoff discussed earlier.

7 RELATION TO EXISTING METHODS

The problem of correctly classifying features or transitions between features is not new. One of the dominant ways to address the problem has traditionally been to rely on derivative models of the data, i.e. local descriptions for the gradient and curvature. This section provides an analysis of how such methods can be used to achieve the piecewise continuous representations of an object of study that are the target of our work. Gradient models offer greater specificity in delineating different transitional regions, e.g. the blue sheet observable in Fig. 7(a). Traditionally, this information has been used to visualize specific transitions in an isosurface like manner. This information can, however, also be used to classify a single solid feature.

As a first example, one can create a minimal intensity/gradient TF that only classifies and visualizes those parts of the feature that have low gradient magnitudes.[1] Unfortunately, this results in severe gaps in the visualization between the two visualized materials, as illustrated in Fig. 7(b) top magnification. To fill the gaps and to create a more reasonable representation of the feature one may extend the classifier widgets in the intensity/gradient domain to include the nearest parts of all transition arcs associated with the targeted feature. However, as illustrated in the bottom magnification of Fig. 7(c), this can introduce a different form of undesirable artifacts as the specificity between different transitions is worse near an arc base than at its apex. Crossing arcs also contribute to similar misclassification problems. While some of the misclassification introduced by the arcs can be removed by manually adjusting the widgets, misclassification still appears, as illustrated in the bottom magnification in Fig. 7(d). This adjustment is also both ad-hoc and time consuming. Furthermore, the number of arcs that need to be adjusted is dependent on the number of *other* features

---

[1]Low here refers to gradient magnitudes caused by natural variations within features which are assumed to be smaller than the magnitudes observed in transitional regions, as assumed by Kindlmann and Durkin [11].
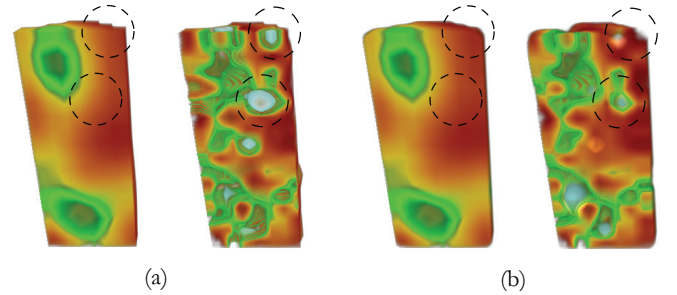

(a)   (b)

Fig. 8: Displaying renderings of original synthetic data (left) and with added noise (right) for post-classified DVR in (a) and our method in (b). With post-classified DVR, noise-disturbed samples will affect also their surroundings (see outlined regions). The same artifacts are much less prominent using our reconstruction method. Thus, our method can help reduce the visual impact of noise.

Table 1: Specifications of data sets and visualizations employed. "Feature overlap" refers to overlap in value ranges.

Fig.	Source	Partial volume	Feature overlap	Size, reference size	σ_{max}
1	Synthetic	No	No	$128 \times 64 \times 1$, $1024 \times 512 \times 1$	0.5
2,9	DECT scan	Yes	Partial	$512 \times 512 \times 512$, N/A	1.25
3,6	Synthetic	No	No	$4 \times 4 \times 1$, (procedural)	0.5
8	Synthetic	No	Yes	$16 \times 16 \times 16$, N/A	0.55
7	CT scan	Yes	Yes	$128 \times 128 \times 128$, N/A	1.75
10	DECT scan	Yes	Yes	$256 \times 256 \times 256$, N/A	1.25

the current feature-of-interest is neighboring. This means that we may end up having to manually adjust a potentially large number of arcs even if our visualization only concerns a single feature. Whereby the visualization becomes implicitly dependent on a complete data model that includes all features that may exist in the data.

8 IMPLEMENTATION

The presented feature-constrained reconstruction is summarized in Algorithm 1. The prototype implementation that has been used to present the images in this paper is based on GLSL code and required only minor changes to an existing software implementation of post-classification DVR. TF widgets are rendered individual layers in look-up tables (1D- and 2D-array textures, respectively) to allow each widget to be sampled independently. To demonstrate the ease of implementation, and further promote the reproducibility of our approach, we provide annotated GLSL code in the supplementary material. The file contains two functions: globally continuous reconstruction and feature-constrained reconstruction.
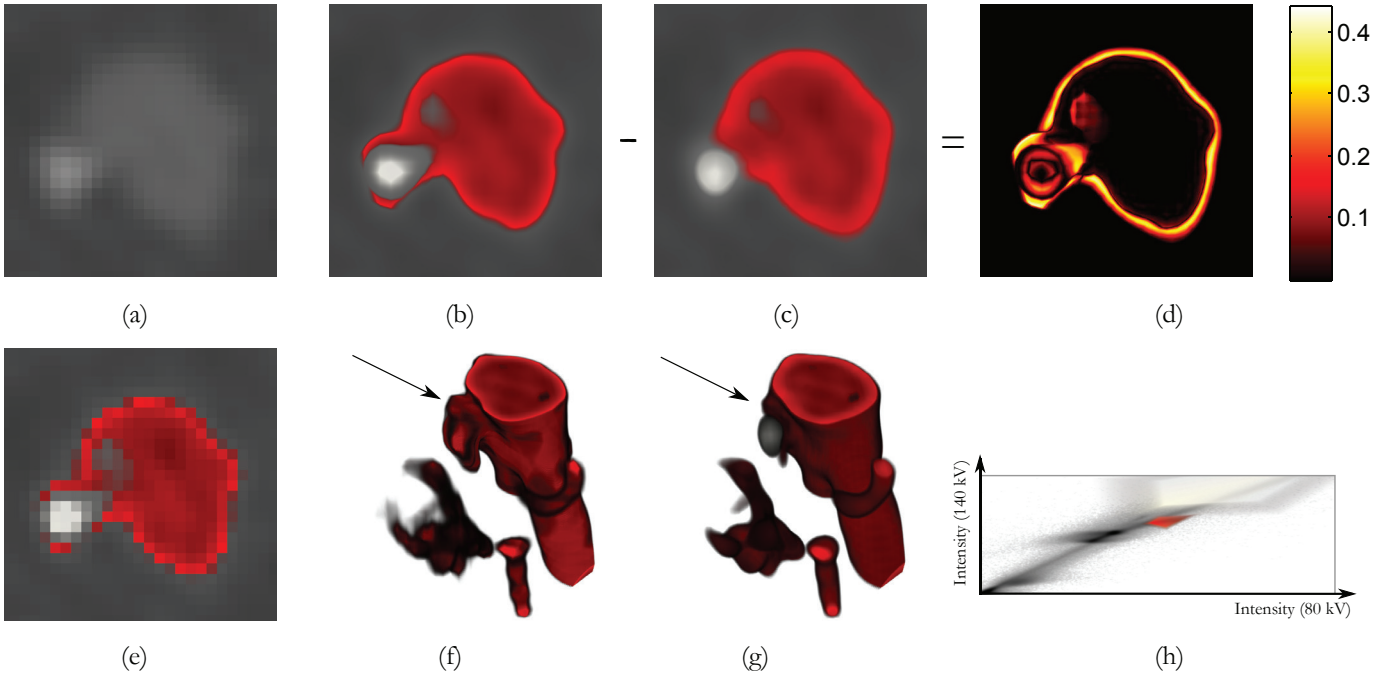
Fig. 9: These renderings exemplify a small region of interest (ROI) of the CT angiography scene introduced earlier in Fig. 2. (a) shows the raw data of the ROI which contains a cross section of a main artery neighbored by a smaller dense material. The following two sub-images show the data as mapped through the TF with; (b) linear interpolation and (c) our feature constrained reconstruction. The visual difference (CIE76 Lab distance) between linear interpolation and our method is highlighted in (d). For reference, we provide the TF mapping with no data interpolation (e) and 3D renderings of and an extended region with linear data interpolation (f) and and our method (g). As seen in both the 2D and 3D renderings, our method prevent the smaller pieces of dense material from appearing as if they were covered in a layer of contrast agent.

Algorithm 1 *Feature-Constrained Reconstruction.*

Require:
  A set of visualized features f_j and a set of TF primitives TF_p
  Access to $P(f_j \mid y_i)$, either precomputed or as a set of parametric functions (see Section 4)
Algorithm:
  for all points x along view ray **do**
    for all features f_j **do**
      1) Compute feature support $\mathcal{F}_{f_j}$ (Eq. 8)
      2) Select reconstruction kernel size based on $\mathcal{F}_{f_j}$ (Eq. 9)
      3) Reconstruct $\hat{z}_{f_j}(x)$ w.r.t. $P(f_j \mid y_i)$ (Eq. 6)
      4) Compute reconstruction validity $\mathcal{V}_{f_j}$ (Eq. 10)
      if Validity $\mathcal{V}_{f_j} > 0$ **then**
        5) Extract visual properties through $TF_p\left(\hat{z}_{f_j}(x)\right)$
        6) Compute visual contributions (Eq. 12)
        7) Blend to buffer
      end if
    end for
  end for

---

9 RESULTS

We have applied the framework to several synthetic and real world data sets. Specifications of the data sizes and origin as well as parameters for the visualizations can be found in Table 1. Dedicated classifier widgets in a separate editor were used to generate Fig. 1 and 8. For all other figures, classifiers derived from the TF widgets were sufficient.

Initially we focus on the highlighted regions in Fig. 1 representing erroneous visual effects that commonly arise in a continuous model. The first highlight (Fig. 1(d)) shows the well-known sheet artifact that appears in transitional regions between features where a visual mapping of an unrelated feature incorrectly is shown. The same problem has been discussed in relation to Fig. 2.

The second highlight (Fig. 1(e)) demonstrates a similar sheet artifact but this time manifesting itself within, as opposed to outside, the visualized feature. In this region of the data set, the values within the 'O' are higher than what the user has selected to show. Outside the 'O' the values are all zero, which the TF also defines as transparent. However, due to the continuous interpolation, a blue sheet appears at the boundary even though the user expects a transparent output in this region. For point sampled data, our method removes this artifact through the weighted reconstruction. For data with partial volume, the effect is hard to separate from what may be natural value shifts within the feature. The effect is dependent on the number of voxels that have been erroneously labeled in the initial classification and can thereby be addressed through improved classifier specificity.

A different type of artifact is shown in the third highlight (Fig. 1(f)). Gaps are formed between adjoining features due to a shrinking effect of both features. This effect always occurs when their respective visual primitives in the TF are disjunct, since reconstructed intermediate values are mapped to transparency. The artifact is even more noticeable in the presence of partial volume as illustrated in Fig. 7 and discussed in Section 7. As highlighted in the inset images of Fig. 7, our method improves the representation of both visualized features without introducing the misclassification that appears in other methods.

The effects of adapting the kernel size to preserve high frequencies within features can be seen in Fig. 7. Here, the kernel size is increased for both features near the transition, yet the solid materials still show the high frequency variations that exist when a global small kernel is used (compare sub-figures (a) and (e)). Another example of how increased kernel size is used to reduce small scale effects of partial volume is evident in Fig. 9 where the high frequencies near transitional regions are misleading. In Fig. 8, a synthetic data set with two features are given overlapping value ranges by adding noise. The sheet-type artifact discussed above causes single noise-disturbed samples to also affect their respective local neighborhoods. Our method avoids the sheet effect by restraining the spatial spread and therefore reduces the visual impact of the noise.

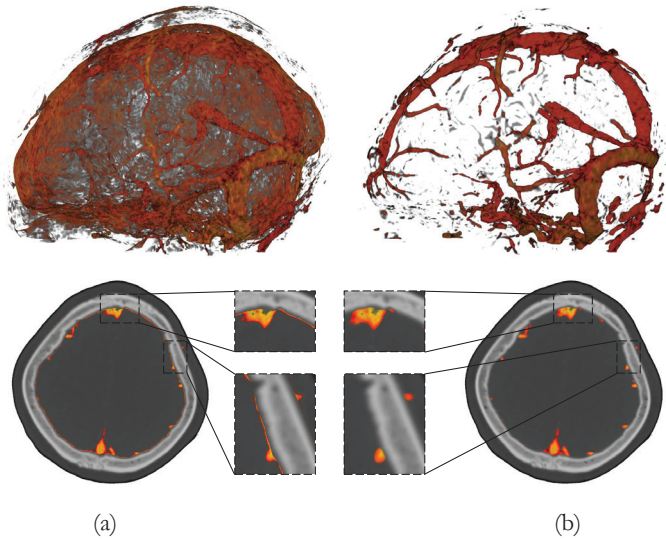(a)   (b)

Fig. 10: This example is rendered from a contrast enhanced DECT data set of a human head. Using a second energy level in the CT acquisition provides improved differentiability between the contrast enhanced vessels and the skull. Even so, classification is still problematic, resulting in severe misclassifications, primarily along the inside of the skull (a). Our method (b) is capable of eliminating much of the thin artifacts, resulting in a more easily interpretable image.

Table 2: Performance measurements for 2D and 3D scalar fields. Interactive performance can be achieved for 2D scalar fields and small 3D scalar fields.

Data set size		Post (HW) $\Omega_K = 2^3$	Post (SW) $\Omega_K = 2^3$	$\sigma_{max} = 0.5$ $\Omega_{max} = 2^3$	$\sigma_{max} = 1.0$ $\Omega_{max} = 4^3$	$\sigma_{max} = 1.5$ $\Omega_{max} = 6^3$
512^2	(2D slice)	0.002 s	0.002 s	0.003 s	0.004 s	0.010 s
16^3	(3D ROI)	0.002 s	0.005 s	0.008 s	0.050 s	0.184 s
128^3	(3D focus)	0.002 s	0.006 s	0.008 s	0.031 s	0.094 s
128^3	(3D)	0.004 s	0.022 s	0.036 s	0.171 s	0.516 s

Finally we also provide examples to demonstrate that the approach scales well and supports data of higher dimensions. This is illustrated in Fig. 2, 9 and 10 which are rendered from data acquired through DECT, i.e. inherently dual-variate data. Fig. 2 and 9 originate from the same data set and depict a CT angiography study of human blood vessels which has been infused with a contrast agent. Fig. 10 similarly depicts a study on vessel anatomy focusing on vessels inside a human head. Dual-energy CT is used in both cases to increase the differentiability between the contrast agent and bone. The differentiation is sufficient to derive acceptable material classifiers directly from the TF (as described in Section 4), and our method is thus applied with next to zero overhead in interaction complexity. For the small focus region in Fig. 9, our method prevents red sheets from obscuring important features. The difference image, showing CIE76 Lab color distances (pixel values in the $[0,1]$ range), highlights that the vessel is also less affected by thinning due to partial volume. In Fig. 10 the vascular tree is made more visible by our method as the artifacts near the skull are filtered away. Reconstruction of the two energy levels was performed separately while the classification was performed in the full 2D space.

When the kernel size is fixed the proposed framework only requires a small number of extra instructions per primitive compared to an equivalent continuous reconstruction kernel (software implementation). More specifically, feature weights need to be fetched or procedurally generated, validity computed, and the final result normalized (steps 3–7 in Algorithm 1). For variable kernel sizes, our prototype implementation currently uses a brute force approach, which computes the kernel size for each sample position by sampling the neighborhood (step 1–2 in Algorithm 1), and is therefore only interactive for 2D scalar fields or for smaller focus/context lenses (subset of full viewport) or regions-of-interest of 3D scalar fields. We performed performance measurements using variable sized kernels, no shading, an early ray termination threshold of 0.98, and two TF primitives on a system with an Intel Core i7 3.6 GHz CPU with 16 GB RAM and a Nvidia GTX 770 graphics card with a single sample per voxel. A viewport size of 512^2 was used for all cases except the focus/context lens which used a 256^2 viewport with a reduced field-of-view. Performance measurements are available in Table 2. Measurements on 3D data were performed on the Tooth data from Fig. 7 while the 2D measures were performed on a 16-bit CT image. The results show that the method in its current form supports interactive exploration for 2D slices at high frame rates, and that it also can be applicable to 3D scalar fields if the rendering is focused on a subsection of the volume. Rendering a full volume at interactive frame rates would require further performance optimization. A performance analysis shows that the overhead is primarily introduced by the evaluation of Equation 8, which represents the sampling needed to determine the kernel size. Determining the kernel size through methods with lower computational overhead, such as gradient analysis, provides opportunities for performance improvements.

10 CONCLUSIONS AND FUTURE WORK

Separation between the object of study (what the user wants to see), the measurable signal (what can be measured) and the measured signal (the sampled data) in the context of signal reconstruction for visualization is the underpinning concept in this work. We made two crucial points; that the user's conceptual understanding of the object of study in many cases consists of multiple disjoint features, and that applying a continuous model in such cases results in severe artifacts in the rendered image.

To address the identified issues we presented a novel boundary aware visualization model that improves upon previous work by utilizing knowledge about features to be accounted for in the reconstruction stage of the visualization pipeline. The approach is general for all visualization work requiring a color mapping and we have exemplified it using slice based and volumetric rendering.

The new framework enabled the use of a piecewise continuous data model by performing a per-feature reconstruction. The theoretical foundation for this was realized by extending kernel regression to constrain reconstruction within single features. Knowledge about features was introduced using a probabilistic interpretation of classification. As this probabilistic interpretation is common in the field of visualization our work is widely applicable. Since our method does not rely on any single implementation to derive the probabilistic classifications it offers great flexibility in the trade-off between classification specificity versus computational and interaction overhead. Notably, one of the proposed probabilistic classification implementations was entirely based on the TF and thus does not introduce any extra user defined parameters.

Although performance is currently a weakness of our approach, it is applicable for slice based rendering as well as for smaller focus regions of volumetric data. Addressing performance by improving the selection of kernel sizes remains an interesting possibility for future work. Other avenues for future work include gradient reconstruction, semi-automatic creation of support functions, and an extension of representation of transitions between regions.

REFERENCES

[1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, New York, 2006.

[2] N. K. Bose and N. A. Ahuja. Superresolution and noise filtering using moving least squares. *IEEE Transactions on Image Processing*, 15(8):2239–2248, 2006.

[3] W. Cai and G. Sakas. Data intermixing and multi-volume rendering. *Computer Graphics Forum*, 18(3):359–368, 1999.

[4] S. Djurcilov, K. Kim, P. Lermusiaux, and A. Pang. Visualizing scalar volumetric data with uncertainty. *Computers & Graphics*, 26(2):239–248, 2002.

[5] R. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *Computer Graphics (Proceedings of SIGGRAPH 1988)*, 22(4):65–74, 1988.

[6] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice-Hall, Inc., 3rd edition, 2006.

[7] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *IEEE Visualization*, VIS '03, pages 301–308, Washington, DC, USA, 2003. IEEE Computer Society.

[8] C. Heinzl, J. Kastner, T. Möller, and M. E. Gröller. Statistical analysis of multi-material components using dual energy CT. In *Vision, Modeling and Visualization Workshop (VMV).*, pages 179–188, Konstanz, Germany, Oct. 2008.

[9] A. Kaufman and K. Mueller. *The Visualization Handbook*, chapter Overview of Volume Rendering, pages 127–174. Elsevier, 2005.

[10] A. Kayugawa, M. Ohkubo, and S. Wada. Accurate determination of CT point-spread-function with high precision. *Journal of Applied Clinical Medical Physics*, 14(4), 2013.

[11] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium on Volume Visualization*, VVS '98, pages 79–86, New York, NY, USA, 1998. ACM.

[12] 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.

[13] J. Kniss, R. Van Uitert, A. Stephens, G.-S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. In *IEEE Visualization*, pages 287–294, 2005.

[14] D. H. Laidlaw. *Geometric Model Extraction from Magnetic Resonance Volume Data*. PhD thesis, California Institute of Technology, 1995.

[15] D. H. Laidlaw, K. W. Fleischer, and A. H. Barr. Partial-volume bayesian classification of material mixtures in mr volume data using voxel histograms. *IEEE Transactions on Medical Imaging*, 17(1):74–86, 1998.

[16] C. Ledergerber, G. Guennebaud, M. Meyer, M. Bacher, and H. Pfister. Volume MLS ray casting. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1546, 2008.

[17] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[18] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman. Uncertainty visualization in medical volume rendering using probabilistic animation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1648–1655, 2007.

[19] I. Prilepov, H. Obermaier, E. Deines, C. Garth, and K. I. Joy. Cubic gradient-based material interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1687–1699, 2013.

[20] S. Rathee, Z. Koles, and T. Overton. Image restoration in computed tomography: Estimation of the spatially variant point spread function. *IEEE Transactions on Medical Imaging*, 11(4):539–545, 1992.

[21] A. Saad, T. Möller, and G. Hamarneh. Probexplorer: Uncertaintyguided exploration and editing of probabilistic medical image segmentation. *Computer Graphics Forum*, 29:1113–1122, 2010.

[22] Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3D local intensity structure for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.

[23] D. Stalling, M. Zöckler, and H. C. Hege. Interactive segmentation of 3D medical images with subvoxel accuracy. In *Computer Assisted Radiology and Surgery*, pages 137–142, 1998.

[24] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transaction on Image Processing*, 16(2):349–366, 2007.

[25] U. Tiede, T. Schiemann, and K. Hohne. High quality rendering of attributed volume data. In *Proceedings of Visualization '98*, pages 255–262, Oct 1998.

[26] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, pages 839–846, 1998.

[27] I. Viola, A. Kanitsar, and M. E. Gröller. Hardware-based nonlinear filtering and segmentation using high-level shading languages. In *IEEE Visualization*, pages 309–316, 2003.