

# Robust and effective method for automatic generation of one-dimensional transfer functions

Rustam Mesquita, Waldemar Celes

Tecgraf/PUC-Rio Institute  
Computer Science Department  
Pontifical Catholic University of Rio de Janeiro, Brazil  
{rustam,celes}@tecgraf.puc-rio.br

**Abstract**—In direct volume rendering, the use of transfer functions allows the visualization of specific parts of the volume, revealing significant structures such as boundaries between different materials. To interactively specify an adequate transfer function is a very challenging and time-consuming task. Besides, a user can settle with a transfer function that does not reveal the overall structure inside the volume. Hence, the importance of generating adequate transfer functions automatically. Since Kindlmann and Durkin’s seminal work on the semi-automatic generation of transfer functions, many research works have been conducted attempting to improve their proposal. Despite some fine resulting visualizations, many of these works aimed to diminish the problem of boundary overlaps, in the data domain, by using two-dimensional transfer functions, thus decreasing the focus on the automatic generation and leaving to the user, again, the task to manually isolate isosurfaces. Moreover, fine-tuning two-dimensional transfer functions is less intuitive than adjusting one-dimensional ones. This paper redirects Kindlmann and Durkin’s concepts of boundary characterization and proposes a new method to automatically generate one-dimensional transfer functions, more robust to boundary overlaps. This work also suggests a rendering filter to isolate detected isosurfaces in the visualization.

## I. INTRODUCTION

A transfer function is an essential component of direct volume rendering. It specifies the contribution of each voxel to the final rendered image. Usually, it is a mapping from scalar value to color and opacity. With the help of an interactive interface, one may manually find an adequate transfer function that reveals the volume boundaries; however, since this is mainly a trial and error process, it requires some time and minimum knowledge about the investigated data. Not surprisingly, (semi) automatic generation of transfer functions is the subject of ongoing researches over the last 20 years.

In 1998, Kindlmann and Durkin [1] significantly improved the automatic detection of boundaries in volume data. They started by characterizing an ideal boundary as a step function blurred by a Gaussian. By computing directional derivatives of the scalar value within the volume, they proposed a strategy to set opacity values to boundary voxels automatically. Their work is of great importance for the area. However, because their proposal relies on average derivative values, it does not adequately handle overlapping boundaries, where two or more boundaries have overlapping ranges of values in the data

domain. Several later works have tried to improve Kindlmann and Durkin’s method but invariably ended up suggesting the use of two-dimensional transfer functions, thus decreasing the focus on the automatic generation and giving attention to user-friendly guided interfaces. Interactivity became an essential component of these approaches.

In this paper, we revisit Kindlmann and Durkin’s boundary model and propose adjustments to their method, obtaining an improved procedure to detect boundary voxels. Our approach is more robust for handling boundary overlaps and is capable of automatically generating effective 1D transfer functions. The generated transfer function reveals the overall structure within the volume data, assigning a single strength value to each detected boundary. As a consequence, we can offer a simple user interface for turning on/off each isosurface and controlling a Gaussian thickness function associated with the boundaries. The interface provides an intuitive way of adjusting the visualization; in general, no search for unrevealed boundaries is necessary.

For rendering, we set an automatic filter to avoid wrong rendering of overlapping boundaries. We also need a filter based on the gradient threshold to prevent rendering of noisy isovolumes.

The rest of this paper is organized as follows. In Section II, we briefly describe other important works that have been published on (semi) automatic generation of transfer functions. Section III reviews in detail the method proposed by Kindlmann and Durkin [1] while, in Section IV, we present our proposal. The results achieved using our method are shown and discussed in Section V, followed by concluding remarks in Section VI.

## II. RELATED WORK

Kindlmann and Durkin’s proposal [1], further detailed in Section III, defines a conceptual boundary model. By correlating the first and the second average derivatives, their method extracts a 1D transfer function. The volume boundaries are identified by arches in a 2D histogram that stores the occurrences of each pair of a scalar value and first derivative. Kindlmann and Durkin’s proposal [1] immediately showed its application in the area of automatic transfer function synthesis [2]. However, the crossing of two or more arches indicates

boundary overlaps in the data domain, which disturbs the correct boundary detection by their method. The 1D transfer function is replaced by a 2D version to handle boundary overlaps, where a “lasso” tool is suggested to select each boundary individually. In this context, Kniss et al. [3] [4] improved the user interface to manipulate the transfer function better over a 2D domain, which is also proposed by Wang et al. [5] and Zou et al. [6].

Searching for a new data domain to generate transfer functions, Pekar et al. [7] proposed the use of different average curves to identify boundaries, as isosurface area, isosurface curvature, and total gradient divided by curvature. Similarly, Tenginakai et al. [8] indicate boundary isovalues with local high order moments. Both works present a 2D histogram where boundaries are revealed by patterns different than arches, but no transfer function is generated automatically.

Distant from 2D histograms, Park and Bajaj [9] use a ray-tracing to eliminate the averages of Kindlmann and Durkin’s method by locally applying their concepts, thus obtaining a spatial transfer function. This approach is more truthful to volume variations, hardly showing displaced boundaries. However, it is a technique susceptible to noisy volumes. Also, it is difficult to provide a way for the user to fine-tune the results.

Lum and Ma [10] proposed extracting transfer functions from a different 2D histogram, storing the pair of scalar values next to each voxel, in the gradient direction. The new histogram identifies boundaries as lines instead of arches, decreasing overlaps in the histogram space. Nevertheless, the new domain still allows lines to cross each other, which does not entirely prevent missing boundary overlaps. Besides, the proposed method is not automatic, as it requires the user to select regions in the histogram.

Many works followed this search for a new transfer function domain. For instance, Sereda et al. [11] proposed the LH histogram, which presents boundaries as circular regions, while Haidacher et al. [12] proposed to use the mean and standard deviation to generate a 2D statistical transfer function with both line and dot representations.

Takahashi et al. [13] proposed the topological volume skeletonization; they presented a data-centric approach that extracts topological structure from the volume. More recently, Lan et al. [14] use the automatic detection of Kindlmann and Durkin [1] with additional steps that evaluate spatial connectivity to handle boundary overlaps in the data domain. Their method separates every structure that is not spatially connected and allows the user to select connected structures so that they can be separated through a segmentation process. Ponciano et al. [15] propose a method with a similar goal, where the volume is classified automatically in different regions in a 2D histogram, then, using a graph-based approach, histogram regions can be merged or divided, according to user interaction. Despite the potential to generate useful visualizations, Lan et al. [14] and Ponciano et al. [15] do not improve the transfer function generation itself. Therefore, the automatic generation of transfer functions remains a challenging research topic.

Ljung et al. [16] provided an overview of research on transfer functions for volume rendering. They opted for classifying the research into six aspects: dimensionality, derived attributes, aggregated attributes, rendering aspects, automation, and user interfaces. Although all these aspects are all correlated, in this work, we focus on automation: our main goal is to provide a robust automatic procedure to generate an initial transfer function. There are two main methods to automatically generate transfer functions: differential geometry-based (or boundary-based) methods, such as the ones employed in the aforementioned works [1] [7] [8], and differential topology-based (or structure-based) methods, such as the ones adopted in [13], [14], and [15]. Recently, Yu and Yu [17] try to combine the advantages of both boundary-based and structure-based methods introducing the use of the standard deviation of ambient occlusion for transfer function generation. Some methods try to gain knowledge using machine learning to better design transfer functions [18], [19], [20].

Here, we strive on the classical boundary-based approach, searching for a yet simple but more effective method. We also set as a goal to work with 1D transfer functions; it is indubitably easier to comprehend for regular users, and it is also the available tool in software packages [21] [22] [23].

### III. BOUNDARY CHARACTERIZATION

#### A. Semi-automatic transfer function

Kindlmann and Durkin [1] assume the behavior of an ideal boundary between two different materials as a high variation of the scalar value, resulting in an abrupt change of material density, blurred in the data acquisition process. Thus, a mathematical model to express the scalar value variation across a boundary can be achieved by the convolution of the step function with a Gaussian kernel.

The generic mathematical representation of an ideal boundary between two materials, according to Kindlmann and Durkin, follows the equation:

$$f(x) = v_{min} + (v_{max} - v_{min}) \frac{1 + \operatorname{erf}(\frac{x}{\sigma\sqrt{2}})}{2} \quad (1)$$

where  $v_{min}$  and  $v_{max}$  are the scalar values of each material, and  $\operatorname{erf}$  is the error function.

Because of the boundary blurriness, there is a range of values that lies between  $v_{min}$  and  $v_{max}$ . This range can be seen as a set of isosurfaces between two isovolumes, where each one of these isosurfaces is a more or less accurate representation of the boundary. The isosurface that better represents the ideal boundary is the one that holds the maximum variation of the scalar value between the two isovolumes. Therefore, the exact position of the ideal boundary can be found at the point where the first derivative is maximum, and the second derivative is zero. Figure 1 shows an ideal boundary  $f(x)$  and its derivatives. In a volume, these derivatives are evaluated in the gradient direction.

Kindlmann and Durkin’s method uses the first and the second derivatives. Equations (2) and (3) show the generic mathematical expression for  $f'(x)$  and  $f''(x)$ . Note that  $f'(x)$

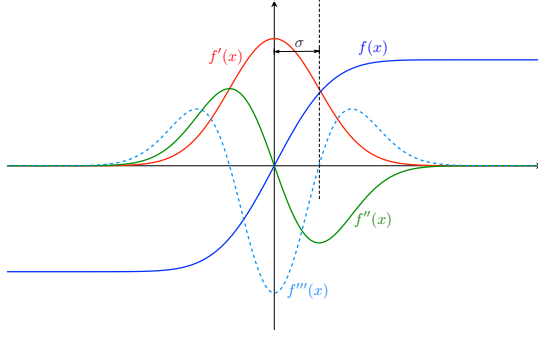


Fig. 1. The ideal boundary function  $f(x)$  and its derivatives.

is a Gaussian function with  $\mu = 0$  and  $\sigma$  representing its standard deviation. Kindlmann and Durkin define the boundary thickness as  $2\sigma$ , since it is enough to capture the value variation across the boundary.

$$f'(x) = \frac{v_{max} - v_{min}}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$

$$f''(x) = -\frac{x(v_{max} - v_{min})}{\sigma^3\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

Combining Equations (2) and (3), one can express the distance  $x$  to the boundary center in terms of  $f'(x)$  and  $f''(x)$ :

$$x = -\frac{\sigma^2 f''(x)}{f'(x)} \quad (4)$$

The standard deviation  $\sigma$  should be constant for each boundary and can also be expressed in terms of the derivatives:

$$\sigma = \frac{f'(0)}{\sqrt{e} f''(-\sigma)} \quad (5)$$

where  $f'(0)$  and  $f''(-\sigma)$  are both maximum values of the first and second derivatives, respectively.

Knowing that  $x = 0$  indicates the exact center of a boundary, one can provide a  $b(x)$  function, mapping distance to opacity. To generate a 1D transfer function that maps isovalues to opacity, Kindlmann and Durkin [1] use a single  $\sigma$  value for the whole volume and compute the average distance to the boundary as a function of the scalar value,  $p(v)$ :

$$p(v) = -\frac{\sigma^2 h(v)}{\max(g(v) - g_{thresh}, 0)}, \quad \sigma = \frac{g(v)_{max}}{\sqrt{e} h(v)_{max}} \quad (6)$$

where  $g(v)$  and  $h(v)$  are, respectively, the first and second directional derivatives, averaged per scalar value, while  $g_{thresh}$  is a threshold parameter that allows the user to decrease  $g(v)$  manually, thus correcting non-zero first derivatives in isovolumes, caused by noise.

The final transfer function is then defined by:

$$\alpha(v) = b(p(v)) \quad (7)$$

which maps the average distance-to-boundary of each data value  $v$  to an opacity value.

Kindlmann and Durkin [1] also show that the 2D histograms of  $g(v) \times v$  and  $h(v) \times v$  hold the same property of indicating a boundary center as the point with the maximum first derivative and second derivative equals to zero, in the gradient direction. Thus, the 2D histogram is a useful way of verifying the presence of ideal boundaries in a volume, as shown in Figure 2.

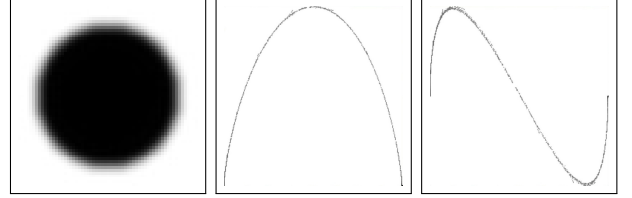


Fig. 2. The slice of a synthetic volume of a sphere and its first and second derivative histograms.

### B. The downsides of averaging

The peaks in  $\alpha(v)$  occur when  $p(v) = 0$ . A careful analysis of the generated transfer function reveals that its peaks occur when  $h(v)$  is closest to zero. This happens because  $h(v)$  is more expressive than  $g(v)$  in determining  $p(v)$ . There are only two situations that leads to  $p(v) = 0$ , either  $h(v) = 0$  or  $g(v) = \infty$ . However, while 0 is inside second derivative's range of expected values,  $\infty$  is not into first's.

This fact becomes a problem when  $h(v)$  achieves zero even though  $v$  does not correspond to a boundary isovalue, which is possible because the second derivative is supposed to be zero at homogeneous regions, as isovolumes. These regions also are supposed to have first derivative equal to zero, thus indicating an isovolume. However, very often, this is not the case when using average values, especially in the presence of boundary overlaps. This problem may cause boundary displacement and an unwanted highlight of isovolumes, depending on how displaced is the detected boundary. Although  $g_{thresh}$  may fix these issues in case of a small overlap, it may also prevent the detection of boundaries with short first derivative.

Besides, in practice, the  $\sigma$  value is not constant to the entire volume. Different boundaries may present different standard deviation values. Also, due to numerical inaccuracies of derivative evaluations, the computed  $\sigma$  value varies even within a single boundary of synthetic volumes.

To better understand these limitations, let us consider the synthetic volume shown in Figure 3. This volume consists of four spheres and presents four boundaries with several overlaps. The boundary ranges are:  $[0, 160]$ ,  $[130, 190]$ ,  $[160, 190]$  and  $[160, 255]$ . Thus, the ideal boundary isovalues are 80, 160, 175, and 207, respectively. Note the isovolume with isovalue equal to 160; this value also characterizes the boundary ranging from 130 to 190.

Figure 3a shows a slice of our synthetic volume and Figure 3b shows the first derivative histogram. The 1D transfer function generated according to Kindlmann and Durkin's method is shown in Figure 4 along with the first and second

average derivatives per scalar value. Both average curves are normalized and smoothed. The four boundaries are correctly revealed by arches in the histogram. However, Kindlmann and Durkin’s method wrongly detected five boundaries: 82, 165, 176, 189, and 211.

Due to boundary overlaps,  $g(v)$  presents the arches distorted and displaced in Figure 4. For instance, the boundary ranging from 130 to 190 is completely lost in  $g(v)$ . The situation is even worst in the second derivative as  $h(v)$  crosses zero at five values, thus indicating five boundaries as can be confirmed by the peaks in the transfer function. The fourth peak highlights the isosurface wrongly detected, as its center value (189) is closer to an isovolume than to a boundary isovalue. Also, the use of  $g_{thresh}$  is not enough to remove the wrong detected isosurface alone. Since the second and third peaks are also centered at  $h(v) = 0$  but with smaller  $g(v)$ , they are the first ones to disappear when increasing  $g_{thresh}$ .

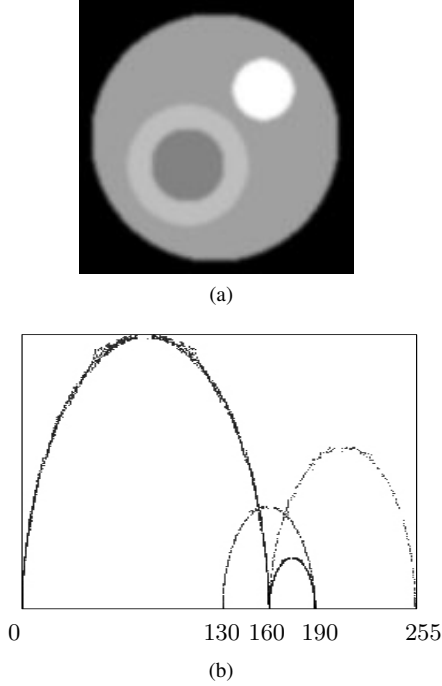


Fig. 3. Volume slice and first derivative histogram of a synthetic volume of four spheres.

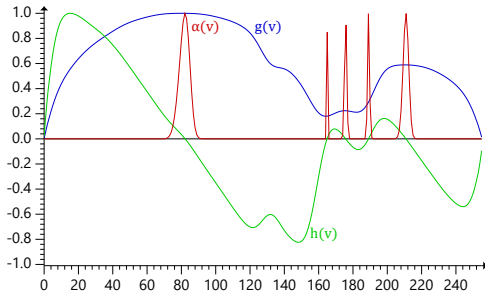


Fig. 4. First  $g(v)$  and second  $h(v)$  average derivatives, and the transfer function  $\alpha(v)$ , according to Kindlmann and Durkin’s method.

To reduce these limitations, Kindlmann and Durkin pro-

posed the use of a 2D transfer function  $\alpha(v, g)$ , assigning opacity per scalar value and first derivative. Despite the improvement in the boundary classification, the 2D transfer function does not prevent boundary displacement caused by the use of average values, because it does not appropriately handle overlapping features. Additionally, the use of 2D transfer functions presents a challenge to the user because its manipulation is much less intuitive than it is with 1D transfer functions.

To improve the generation of 1D transfer functions while preserving the mathematical concepts described by Kindlmann and Durkin [1], we propose the use of a  $\sigma$  value per voxel, a filter to restrict the data used in the histogram computations, and a weighted average to enhance the influence of voxels closer to the boundaries.

#### IV. OUR PROPOSAL

##### A. Handling boundary overlaps

The use of Equation (6) to indicate the distance to the boundary has two main limitations when applied to volumes with different overlapping boundaries. First, the standard deviation varies throughout the volume. Considering a single  $\sigma$  value to the entire volume brings inaccuracy to the computed “distance to the boundary”. Second, the use of average directional derivatives, per scalar value, causes boundaries to be missed or displaced; a condition that is aggravated for volumes where a single scalar value represents both a boundary and an isovolume. This problem suggests the use of spatial transfer functions, which can be effective but hard to be fine-tuned by the user. In this section, we present our proposal to filter voxels for boundary detection while still generating 1D transfer functions.

Our first goal is to consider different  $\sigma$  values throughout the volume. We want to compute a  $\sigma$  value for each voxel but still based only on local computations. To achieve that, we also evaluate the third directional derivative  $f'''(x)$ , which is mathematically expressed as:

$$f'''(x) = \frac{(x^2 - \sigma^2)(v_{max} - v_{min})}{\sigma^5 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (8)$$

Figure 1 also illustrates the behavior of  $f'''(x)$  along an ideal boundary (dashed line). Note that an ideal boundary region can be easily detected by negative values of this third derivative. Combining Equations (2), (3) and (8), we can express  $x$  directly, as a function of the three directional derivatives:

$$x = \frac{f'(x) f''(x)}{f'(x) f'''(x) - f''(x)^2} \quad (9)$$

Similarly, we can evaluate  $\sigma$  per voxel:

$$\sigma = \sqrt{\frac{-f'(x)^2}{f'(x) f'''(x) - f''(x)^2}} \quad (10)$$

Note that, as the first derivative assumes positive values, and the third derivative assumes negative values, in the boundary

region, the radicand value in this expression will always be positive, as expected.

Based on Equations (9) and (10), we can classify each voxel related to its distance to the boundary and then avoid processing voxels in isovolume regions. All the derivatives are calculated by central differences at each voxel.

### B. 1D transfer function generation

Our goal is to identify the best representative isovalue for each boundary in the volume. We also want to associate a weight value to each isovalue as an indication of boundary strength. The final transfer function will be given by Gaussian functions whose mean values are the detected isovalues, with amplitudes given by the associated strength values. The user can then control the overall opacity and boundary thickness by modifying the standard deviation of these Gaussian functions.

As the goal is to generate a 1D transfer function, associating opacity per scalar value, averaging is indispensable. To reduce the side effects of boundary overlaps, we only process voxels close to boundaries. For each voxel, we compute  $x$  and  $\sigma$  according to Equations (9) and (10). We then compute a *normalized distance* to the boundary expressed by  $x/\sigma$ , discarding voxels where  $|x/\sigma| > 1$ . Moreover, to enhance the contribution of voxels closer to ideal boundaries, we compute a *weighted average*, setting the weight proportional to the normalized distance, following a Gaussian distribution. We apply a Gaussian function,  $G$ , to the normalized distance, computing the contribution of this voxel per scalar value:  $\delta(v) = G(x/\sigma)$ , where  $v = f(x)$ . The amplitude of this Gaussian function is set to 1, its mean value is  $\mu_G = 0$  and its standard deviation is  $\sigma_G = 1/3$ ; thus, its image varies from 0 to 1. Note also that choosing  $\sigma_G = 1/3$  would make negligible the contribution of voxels where  $|x| > \sigma$ . The  $\delta(v)$  values are then averaged per  $v$ , resulting in what we call *boundary-ness curve*:  $\bar{\delta}(v) \times v$ . This curve is finally smoothed to deal with noisy data. The local maxima of the *boundary-ness curve* indicate the representative isovalues, and its maximum values the associated weights.

Figure 5 illustrates the boundary-ness curve and the associated 1D transfer function achieved by applying our proposal to the synthetic model of Figure 3: The generated transfer function captures all four boundaries. The detected isovalues are 81, 156, 175 and 210, presenting some displacements (should be 80, 160, 175 and 207, respectively), but still providing a reliable approximation of all boundaries.

### C. Rendering filters

A boundary isovalue can also represent isovolumes or can overlap the data range of other boundaries. Since we propose to work with 1D transfer functions, we need to employ filters to avoid rendering non-boundary regions. We propose a filter that follows the criterion to consider voxels for average computations. We only consider opacity for voxels where  $|x| < \sigma$ , that is, we discard voxels that are not in a boundary region. This filter avoids the wrong rendering of isovolumes and overlapping boundaries and is sufficient for clear visualization

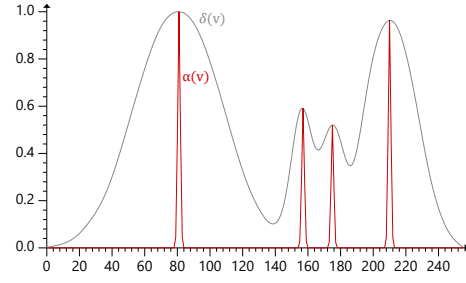


Fig. 5. The boundary-ness curve (gray) and the transfer function (red) of the synthetic model.

of the four detected isosurfaces for the synthetic model, as illustrated in Figure 6.



Fig. 6. Volume rendering of the synthetic model with four boundaries, using our proposed transfer function.

Besides the boundary filter, we need an additional filter to avoid rendering noisy isovolumes. In noisy data, the scalar values in isovolume regions vary significantly, making the condition  $|x| < \sigma$  not sufficient to prevent their rendering. For such cases, we also employ a global filter based on gradient threshold,  $f'(x) > g_{thresh}$ , avoiding the rendering of noisy isovolumes. In general, we observed that setting  $g_{thresh} = 0.10 g_{max}$  is a good starting value, where  $g_{max}$  represents the maximum first derivative in the volume.

## V. RESULTS

We ran a set of computational tests to show and discuss how our proposal performs on different datasets<sup>1</sup>. For each dataset, we compare the results achieved by our method with the results obtained with our implementation of Kindlmann and Durkin's [1] method. For their method, we did our best to adjust an appropriate  $g_{thresh}$  value manually.

The first test was done considering *The Tooth* dataset, a  $256 \times 256 \times 161$  CT Scan volume. The first derivative histogram of this model shows a noisy data with well defined overlapping arches (Figure 7a). Our proposal automatically identifies the five boundaries, generating an adequate 1D transfer function, as shown in Figure 7b. Note that the isovalues 75 and 165 are

<sup>1</sup>The models were obtained from <http://schorsch.efi.fh-nuernberg.de/data/volume/>.



coincident to isovolumes. Nevertheless, the voxels in these isovolumes did not disrupt the identification of the boundaries. Our implementation of Kindlmann and Durkin's method generated the transfer function with displaced isovalues, as showed in Figure 7c. As discussed, such displacements happen because the boundary overlaps biased the average values.

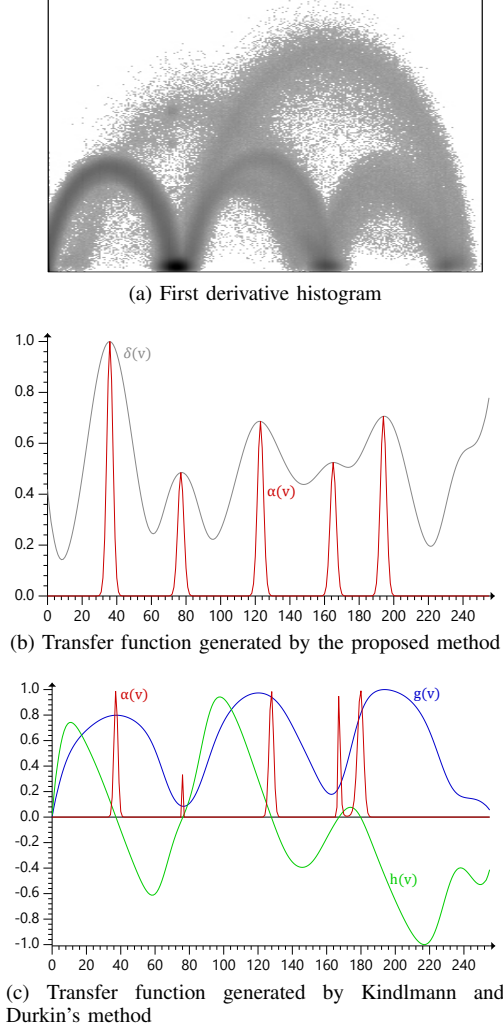


Fig. 7. Achieved results for *The Tooth* dataset.

Figure 8 illustrates the achieved rendered image using the transfer function generated by our method. The rendering filters prevented the isovolumes from being visualized.

Our method considers only voxels close to the boundary, discarding others from overlapping arches. Also, the weighted average eliminates any wrong voxel classification since voxels closer to boundaries have more influence on the average values. The third derivative plays an essential role in this filtering. Our per-voxel evaluation of  $x$  and  $\sigma$  values are less sensitive for being disrupted by overlapping arches.

Visually comparing, one can verify that, for this model, other proposed interactive techniques, using two-dimensional transfer function [4], topology-based procedure [13], or LH histograms [11], do not reveal any other boundary besides the ones detected by our method.

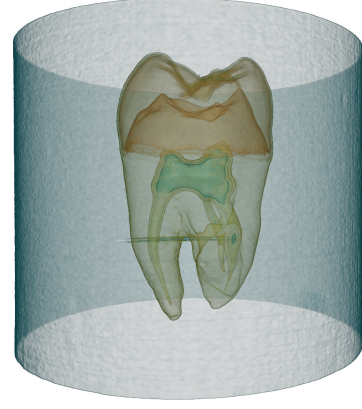


Fig. 8. *The Tooth* dataset rendering.

To provide a better understanding of the proposed approach, Figure 9 shows the first derivative histogram again but now built only with voxels not discarded in the average computations. Note that voxels near the arch bases are discarded. Note also that voxels in the isovolumes regions are discarded too; however, due to noise, a lot of isovolume voxels remain. The applied color scale shows the average values of *normalized distances*. Note how effective is the classification of boundary proximity based on these normalized distances. The weighted average minimizes the influence of the remaining isovolume voxels.

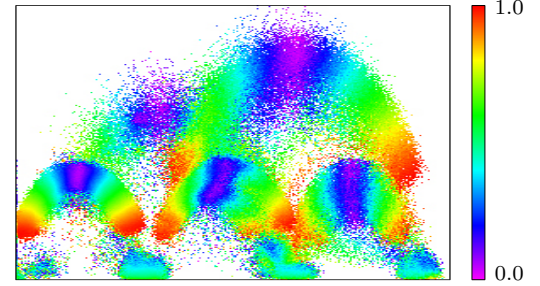


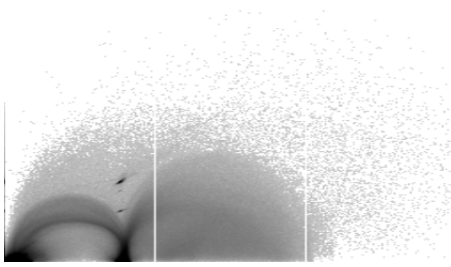
Fig. 9. Filtered histogram of the *Tooth* dataset. The color scale indicates normalized distance to the boundary.

The second dataset that was considered is *The Visible Male*  $128 \times 256 \times 256$  CT Scan. Figure 10a shows the corresponding gradient histogram. Note that it is a noisy data. Our method detected two main boundaries: one representing the skin and another representing the skull. The proposed technique also automatically detected two other with high densities depicting the teeth (see Figure 10b). After manually fine-tuning the  $g_{thresh}$  value, we were able to make Kindlmann and Durkin's method of slightly revealing one tooth boundary (see Figure 10c). The teeth are hard to detect because they are small structures in the data volume, being represented by a small number of hits in the histogram, not easily observed. The ability to automatically revealing such small structures makes our method especially promising for detecting anomalies in medical exams.

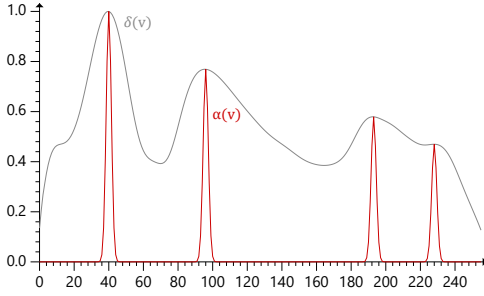
Nevertheless, the skull boundary, essential in the dataset,

was detected displaced by the Kindlmann and Durkin’s method, which can be noted in Figure 11. Our detected boundary reveals a more complete skull structure (Figure 11a), while theirs reveals an incomplete one (Figure 11b). Using a 2D transfer function, Kniss et al. [4] were only capable of revealing a complete structure employing a non-intuitive sheared triangular classification widget.

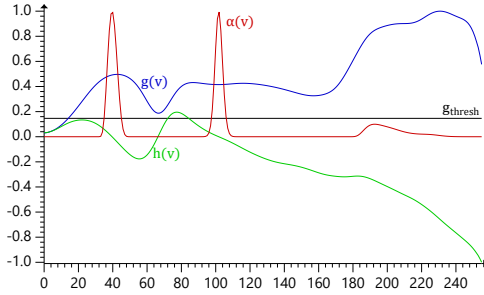
One concern with the proposed method is that the third derivative tends to be more sensitive to noise. This experiment indicates that our proposal performs well also on noisy data. However, an in-depth analysis of how such sensitive may affect the results is still needed.



(a) Histogram



(b) Our method

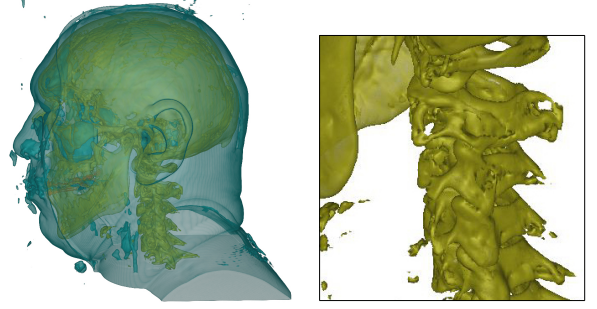


(c) Our implementation of Kindlmann and Durkin’s method

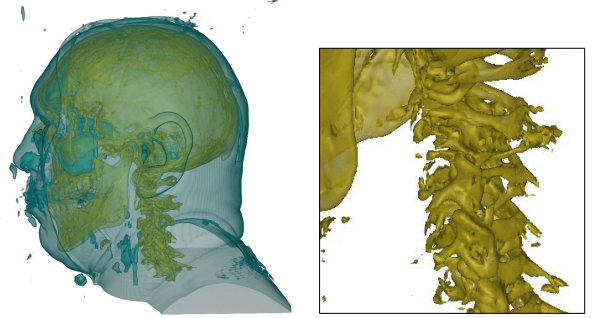
Fig. 10. 1D transfer function of *The Visible Male* dataset.

As an additional test, we considered a  $512 \times 512 \times 59$  dataset of the coronal human upper body. This dataset is also a noisy volume. Our method detected five boundaries as indicated by our *boundary-ness curve* in Figure 12, while Kindlmann and Durkin’s method detected only two main boundaries. The third main boundary, not captured by their method, represents the muscles, as visualized in Figures 13a. The other two not detected represents smaller structures.

Even for the two main boundaries, the proposed method



(a) Our method



(b) Our implementation of Kindlmann and Durkin’s method

Fig. 11. *The Visible Male* dataset rendering from the generated 1D transfer functions.

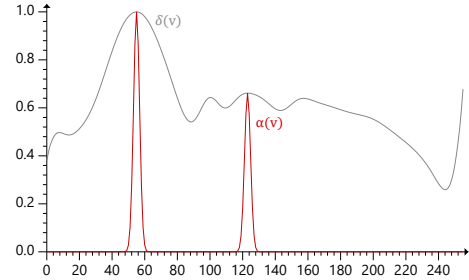


Fig. 12. Generated transfer functions for the dataset of the coronal human upper body. Note that there are five detected boundaries; only the two main boundaries are turned on in the figure.

performed better. Figure 13b and 13c shows the corresponding images obtained using ours and Kindlmann and Durkin’s method, respectively. Note that the spine appears incomplete using their generated transfer function.

## VI. CONCLUSION

In this paper, we presented a simple but yet powerful method to generate 1D transfer functions automatically. We redirect Kindlmann and Durkin’s [1] concepts of boundary characterization and presented an improved approach to detect boundaries in volume data. The main improvements over Kindlmann and Durkin’s method are:

- The filter to discard voxels not close to the boundary in the average computation.

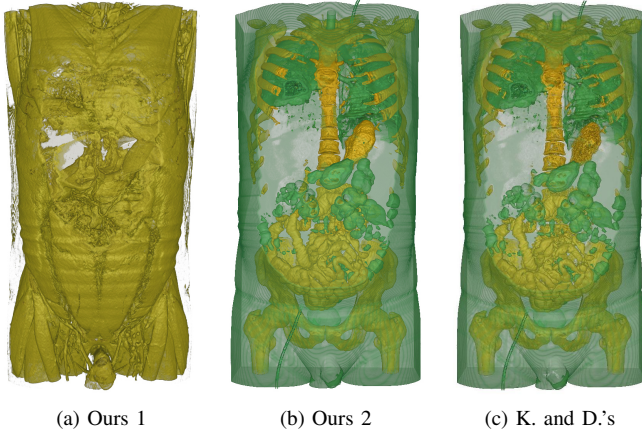


Fig. 13. Visualization for the dataset of the coronal human upper body: (a) visualization of the muscles only revealed by our proposal; (b) the two main boundaries as detected by our method; and (c) the two main boundaries as detected by Kindlmann and Durkin's method. Note that in our method, the spine appears complete.

- The computation of a  $\sigma$  value per voxel, improving the accuracy of calculating distances to the boundary. This computation is supported by the use of the third directional derivative.
- The use of a weighted average, enhancing the contribution of voxels closer to the boundary.
- The idea of identifying isovalues first, with their corresponding strengths. The transfer functions are then obtained by a sequence of Gaussian functions, where the mean values are the detected isovalues, and the amplitudes are the associated strength values.

We have also proposed a rendering filter, essential to isolate the boundaries from the overlapping arches since we are employing a 1D transfer function. Achieved results have demonstrated the robustness and effectiveness of our method on revealing the overall structure of a volume data in a clean image.

A natural extension of this work is its integration with interactive procedures to explore volume data, especially those working with multi-dimensional transfer functions. Our boundary detection procedure could be used to assist on automatic extraction of isovolumes; our rendering filter could be combined with interfaces to edit multi-dimensional transfer functions. Additionally, as mentioned, an in-depth sensitive-to-noise analysis of the proposed method is needed, mainly because of the use of the third derivative in the computation.

## REFERENCES

- [1] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *IEEE Symposium On Volume Visualization*, Oct. 1998, pp. 79–86, 170.
- [2] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. Avila, K. M. Raghun, and R. M. and, "The transfer function bake-off," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 16–22, May 2001.
- [3] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," in *Visualization, 2001. VIS '01. Proceedings*, 2001, pp. 255–562.
- [4] —, "Multidimensional transfer functions for interactive volume rendering," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, 2002, pp. 270–285.
- [5] Y. Wang, W. Chen, G. Shan, T. Dong, and X. Chi, "Volume exploration using ellipsoidal gaussian transfer functions," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, 2010, pp. 25–32.
- [6] Y. Zou, D. Bai, S. Wang, and Y. Kang, "A new two-dimensional transfer functions for volume rendering," in *2010 3rd International Congress on Image and Signal Processing*, Oct 2010, pp. 3733–3736.
- [7] V. Pekar, R. Wiemker, and D. Hempel, "Fast detection of meaningful isosurfaces for volume data visualization," in *Visualization, 2001. VIS '01. Proceedings*, 2001, pp. 223–230.
- [8] S. Tenginakai, J. Lee, and R. Machiraju, "Salient iso-surface detection with model-independent statistical signatures," in *Visualization, 2001. VIS '01. Proceedings*, 2001, pp. 231–238.
- [9] S. Park and C. Bajaj, "Multi-dimensional transfer function design for scientific visualization," in *Proceedings of the Fourth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, 2004, pp. 290–295.
- [10] E. B. Lum and K.-L. Ma, "Lighting transfer functions using gradient aligned sampling," in *Proceedings of the Conference on Visualization '04*, ser. VIS '04. IEEE Computer Society, 2004, pp. 289–296.
- [11] P. Sereda, A. V. Bartoli, I. W. O. Serlie, and F. A. Gerritsen, "Visualization of boundaries in volumetric data sets using LH histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 208–218, Mar 2006.
- [12] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Grller, "Volume visualization based on statistical transfer-function spaces," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, 2010, pp. 17–24.
- [13] S. Takahashi, Y. Takeshima, and I. Fujishiro, "Topological volume skeletonization and its application to transfer function design," *Graphical Models*, vol. 66, no. 1, pp. 24 – 49, 2004.
- [14] S. Lan, L. Wang, Y. Song, Y. P. Wang, L. Yao, K. Sun, B. Xia, and X. Zongben, "Improving separability of structures with similar attributes in 2d transfer function design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 99, pp. 1546–1560, 2016.
- [15] D. Ponciano, M. Seefelder, and R. Marroquim, "Graph-based interactive volume exploration," *Computers & Graphics*, vol. 60, pp. 55–65, Nov 2016.
- [16] P. Ljung, J. Krger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, "State of the Art in Transfer Functions for Direct Volume Rendering," *Computer Graphics Forum*, vol. 35, no. 3, pp. 669–691, Jun. 2016.
- [17] L. Yu and H. Yu, "Boundary-structure-aware transfer functions for volume classification," in *SIGGRAPH Asia 2017 Symposium on Visualization - SA '17*. Bangkok, Thailand: ACM Press, 2017, pp. 1–8.
- [18] Fan-Yin Tzeng, E. Lum, and Kwan-Liu Ma, "An Intelligent System Approach to Higher-Dimensional Classification of Volume Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 3, pp. 273–284, May 2005.
- [19] F. D. M. Pinto and C. M. D. S. Freitas, "Design of Multi-dimensional Transfer Functions Using Dimensional Reduction," *Eurographics/ IEEE-VGTC Symposium on Visualization*, p. 8 pages, 2007.
- [20] L. Wang, X. Chen, S. Li, and X. Cai, "General Adaptive Transfer Functions Design for Volume Rendering by Using Neural Networks," in *Neural Information Processing*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, I. King, J. Wang, L.-W. Chan, and D. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4233, pp. 661–670.
- [21] U. Ayachit, *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.
- [22] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. Weber *et al.*, "Visit: An end-user tool for visualizing and analyzing very large data. high performance visualization-enabling extreme-scale scientific insight," *Insight*, pp. 357–372, 2012.
- [23] C. ImageVis3D, "An interactive visualization software system for large-scale volume data," 2015.