Pacific Graphics (2018)
H. Fu, A. Ghosh, and J. Kopf (Editors)

Short Paper

Robust material graphs for volume rendering

Ojaswa Sharma[†], Tushar Arora, and Apoorv Khattar

Indraprastha Institute of Information Technology - Delhi, India

Abstract

A good transfer function in volume rendering requires careful consideration of the materials present in a volume. In this work we propose a graph based method that considerably reduces manual effort required in designing a transfer function and provides an easy interface for interaction with the volume. Our novel contribution is in proposing an algorithm for robust deduction of a material graph from a set of disconnected edges. Since we compute material topology of the objects, an enhanced rendering is possible with our method. This also allows us to selectively render objects and depict adjacent materials in a volume.

CCS Concepts

•***Computing methodologies*** → ***Machine learning approaches; Rendering;*** *Image segmentation; Volumetric models;*

1. Introduction

The necessity of creating a good transfer function (TF) goes unsaid in visualising 3D volumes. Given the fact that many real-life medical CT/MRI volumes have a complex mix of materials, manual TF creation becomes increasingly difficult. Numerous approaches have been proposed for semi-automatic and automatic TF creation. Majority of automated TF creation algorithms work with material boundaries or interfaces. While there have been excellent feature spaces (like intensity-gradient, and LH) to analyse material boundaries, these do not do justice to the interior of the materials when it comes to visualisation. A complete visualisation must map the interior of all materials and not just their interfaces. Motivated by this, we propose an algorithm for deduction of the material graph from material boundaries. Our graph generation is central to TF creation, and provides fine control over material color and opacities. The by-product of our pipeline is automatic background removal in volumes, which could be difficult to achieve otherwise.

2. Related work

Automatic synthesis of transfer functions has been explored quite well in the volume visualization community. The areas of assisted and interactive volume exploration are also researched extensively and many of such ideas are well summarised in the survey article by Ljung et al. [LKG*16]. High-dimensional TFs were introduced with histograms composed of multiple features derived from the intensity volume [KKH02, ŠBSG06, CM09]. Many approaches have been proposed to automatically deduce higher dimensional TFs from these feature spaces or histograms. Clustering is a popular approach to create groups of similar samples for TF creation. Maciejewski et al. [MWCE09] use non-parametric clustering over intensity-gradient histogram for colour mapping and volume interaction. Cai et al. [CTN*13] seek to automate the transfer function generation process by doing colour and alpha assignment using intensity-gradient magnitude histogram. The authors perform a histogram segmentation using normalized-cut and assign colors by projective mapping. In an interesting approach, Wang et al. [WZL*12] apply Morse theory for creating valley cells from feature space. Their topologic approach to clustering does not depend on the number and shape of materials. The LH histogram is another robust feature space that outlines material edges. Šereda et al. [ŠBSG06] use hierarchical clustering technique on LH histogram space with granularity control based on depth of the dendrogram. Further, Wang et al. [WZL*12] used a modified version of dendrogram for material segmentation and feature exploration. A recent approach by Cai et al. [CNCO17] uses self-organising map (SOM) to get a clustered 2D topology and make graph exploration easy through simple transfer function control.

Several approaches utilise machine learning based algorithms other than clustering. Soundararajan and Schultz [SS15] motivate the use of supervised classification in learning probabilistic transfer functions. The authors compare several classification algorithms and recommend random forests in general. Qin et al. [QYH15] propose a voxel visibility metric based on the Gaussian mixture model and use mathematical optimisation to to minimise distance between the desired and the actual visibility distribution.

In this work, we suggest a graph-based framework for creation of material definitions and volume exploration. Unlike many existing works that focus primarily on clustering material boundaries to deduce a transfer function, we instead create material definitions that can map a solid material interior to a color value. Our algorithm generates a material graph that can be used for further topologic

† Corresponding author, e-mail: ojaswa@iiitd.ac.in

DOI: 10.2312/pg.20181282

analysis of the volume. With our approach, we also show robust background removal and assignment of opacity values for interactive visualisations.

3. Proposed approach

The main problem that we solve in this work is of robustly discovering various materials in a volume and their spatial interconnectivity. We define a material $\mathcal{M}$ in a given volume to have an intensity distribution $\mathcal{V} : (\mu, \sigma)$ and a mean occlusion value of η. $\mathcal{V}$ is represented by a gaussian of mean intensity μ and a standard deviation of σ. The occlusion value indicates how deep inside the volume lies a particular material, which is useful in assigning transparency values for an effective visualisation of interior of the volume. A material also has a shape with solid interior (with possible holes and multiple disconnected components). We denote the boundary of a material by $\partial\mathcal{M}$. Two adjacent materials share a common interface boundary $\mathcal{I}$ which is part of each material boundary. Therefore, it is possible to define the spatial interconnectivity of materials by a graph $\mathbf{G}(\mathbf{M}, \mathbf{I})$ with set of vertices $\mathbf{M}$ containing all materials (including the background material) and set of edges $\mathbf{I}$ containing interfaces between pairs of adjacent materials.

At the onset, $\mathbf{M}$, $\mathbf{I}$, and $\mathbf{G}$ are unknowns. A naïve approach to discover $\mathbf{M}$ could be to perform some sort of clustering or segmentation on the input voxel values. In our experiments, we found that this does not always result in well defined and consistent materials. Instead, we first look at an initial edge set $\hat{\mathbf{I}}$ and simultaneously deduce $\mathbf{M}$, $\mathbf{I}$, and $\mathbf{G}$ from it via a local optimization. An initial edge set is accurately captured by the LH histogram proposed by Šereda et al. [ŠBSG06]. An LH value is computed for an edge pixel by performing a bidirectional traversal along the local gradient to search for a low (L) and a high (H) value in a small neighborhood. These low and high values are indeed samples from two adjacent intensity distributions $\mathcal{V}_{i,1}$ and $\mathcal{V}_{i,2}$, and represent a sample from their interface $\mathcal{I}_i$. The central part of our algorithm is a robust material deduction from $\hat{\mathbf{I}}$ and creation of topology in the form of a material graph $\mathbf{G}$.

3.1. Graph deduction

The initial edge set $\hat{\mathbf{I}}$ consists of samples from multiple interfaces. For a large volume, these samples could be high and therefore to keep the algorithm computationally tractable, we perform subsampling of the interface set from their distribution. We define material interfaces by clustering LH samples with the HDBSCAN [CMS13] algorithm.

There are two primary challenges in material graph deduction from a set of material interfaces. Firstly, a given interface only conveys that the two given materials are adjacent, and there exist no explicit material connectivity. This situation arises since the material distributions are not known yet. Secondly, material boundaries detected by a clustering algorithm may result in repeating boundaries of the same interface of two materials due to noise in volume data and subtle variations of intensities. Our graph deduction assumes that the clustering algorithm provides valid material-material interfaces and that two materials of an interface are not same. As a consequence our algorithm does not try to collapse a detected interface to a single material. However, similar interfaces may be merged into one interface or similar materials from different interfaces may be merged together. We use the Bhattacharyya distance metric [Bha43] in our algorithm to determine relative separation between two material distributions. Noise and intensity variations in our algorithm are considered by modelling material intensities with a normal distribution.

Our algorithm looks at distances between materials of given interfaces to match and merge similar materials. We first create a distance matrix $\mathbf{D}$ of size $2k \times 2k$ where k is the number of initial interfaces. $\mathbf{D}_{ij}$ gives Bhattacharya distance $d_B(\mathcal{V}_i, \mathcal{V}_j)$ between any two intensity distributions $\mathcal{V}_i$ and $\mathcal{V}_j$. Materials of an interface appear consecutively in rows and columns of $\mathbf{D}$ and are assigned a distance value of infinity. Distance to the same material (i.e, $d_B(\mathcal{V}_i, \mathcal{V}_i)$ is also set to infinity. Thus, the distance matrix is computed as

$$\mathbf{D}_{i,j} = \begin{cases} \infty & \text{if } i = j, \\ \infty & \text{if } i = j-1 \text{ and } i \text{ odd}, \\ \infty & \text{if } i = j+1 \text{ and } i \text{ even}, \\ d_B(\mathcal{V}_i, \mathcal{V}_j) & \text{otherwise.} \end{cases}$$

A value of infinity indicates that the connectivity between such material pairs will not be affected by the algorithm. We compute a base distance threshold θ_{base} as minimum of the distances of material pairs in the initial interfaces

$$\theta_{base} = \min_{i \in \{1,\ldots,k\}} \left\{ d_B\left(\mathcal{V}_{i,1}, \mathcal{V}_{i,2}\right) \right\},$$

We choose two multipliers $\varepsilon << 1$ and $\zeta >> 1$ to define a possible range of threshold values. A threshold $\theta \in [\varepsilon\theta_{base}, \zeta\theta_{base}]$ is used to merge materials based on $\mathbf{D}$.

Our graph deduction algorithm is a greedy approach to merge similar materials. Initially the material graph $\mathbf{G}$ contains disconnected edges $\hat{\mathbf{I}}$. For a particular value of threshold θ, we deduce a feasible graph in two steps: finding materials to merge, and reconstructing material graph. In the first step, the algorithm walks through the distance matrix $\mathbf{D}$ to find out similar materials. This is performed by creating sets of materials that are similar to each other. Such a set may include a material from an interface, but not both materials. The later will imply collapsing an interface to a single material. Each set is reduced to into a single material which gives the set of materials $\mathbf{M}$. We implement the same by creating parent-child hierarchy and then merging all children with their respective parents.

The second step constructs a valid and connected graph from $\mathbf{M}$. This is performed by merging nodes of similar materials together and merging repeating interface edges into one interface edge. Since each material intensity is a distribution function with its mean and standard deviation, the resulting material intensities are computed by a weighted average of the constituent material distributions.

3.1.1. Optimal graph search

A particular value of θ may result in a suboptimal graph configuration. We search for a material graph that maximises the separation energy

$$\mathbf{E}_{sep}\left(\mathbf{G}(\mathbf{M}, \mathbf{I})\right) = \frac{1}{|\mathbf{I}|} \sum_{j=1}^{|\mathbf{I}|} d_B(\mathcal{V}_{j,1}, \mathcal{V}_{j,2}). \tag{1}$$

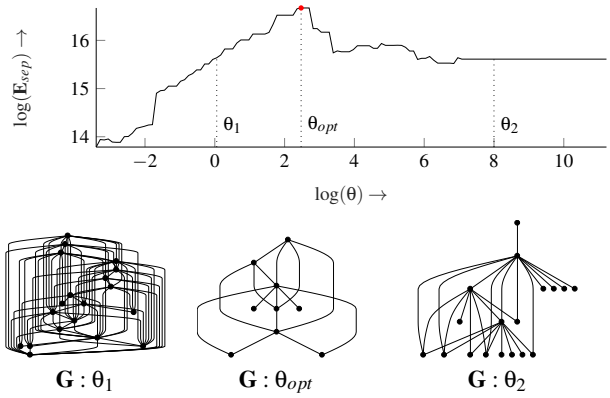

Figure 1: *Log-log plot of separation energy* $\mathbf{E}_{sep}$ *vs. distance threshold* θ. *A robust and optimal graph is obtained by maximizing this energy. The graphs for three values of theta indicate that an optimal theta ensures minimal material nodes.*

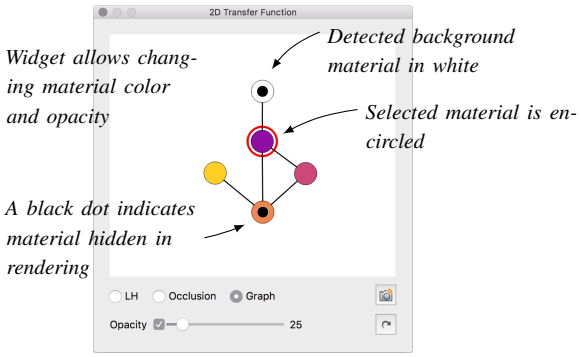

Figure 2: *Graph-based TF widget where vertices represent materials and edges indicate material adjacency.*

Intuitively, maximising $\mathbf{E}_{sep}$ results in a graph that prefers fewer but well separated interface edges. It can be seen in (1) that $\mathbf{E}_{sep}$ is directly proportional to the sum of the all edge distances in the graph and maximising the total distance will result in a better separation for the graph. Further, $\mathbf{E}_{sep}$ does not increase with a large number of small interface edges since the energy term represents average interface separation in the graph. This results in a robust selection of the material graph. Variation of $\mathbf{E}_{sep}$ is shown in Figure 1 for the Bonsai dataset.

3.2. TF design and volume visualization

We use the synthesised material graph to implicitly represent a high dimensional transfer function. We hide complexity of TF manipulation by providing the user with very few and meaningful controls for changing appearance of the volume rendering. Our TF widget consists of materials displayed as graph nodes with edges drawn between adjacent materials (see Figure 2). The user can select any node and change material color and transparency to change the volume rendering. This is particularly useful in hiding a set of materials and displaying only a few or manually changing the opacity for better visualization. To start with, materials are automatically assigned visually separated colors and occlusion derived opacities. Graphically, invisible materials are indicated with a black dot on the corresponding graph node. By default, the detected background material (always colored as a white node) is set to invisible to hide it from the volume rendering.

4. Results

We apply our fully automated pipeline for graph deduction and volume rendering on several volumetric datasets. Our real-time GPU volume renderer is written in C++ with a Python backend. The 3D volume datasets are obtained from The Volume Library [Roe12]. Figure 3 shows renderings of these volume datasets along with their voxel classification and computed material graphs. The occlusion values are scaled to derive transparency of the voxels.

Table 1: *Running times (in sec.) for various steps in the pipeline.*

Dataset	Interface features	Interface subsampling	Interface clustering	Graph deduction	Voxel classification	Total
Bonsai	8.69	11.23	62.83	1.51	19.19	103.46
CT Head	3.67	4.99	40.04	0.44	4.08	53.21
Engine	2.12	2.53	24.35	0.26	1.88	31.05
Mouse	4.33	4.63	32.05	0.37	4.05	45.43
Tooth	1.14	1.11	2.01	0.31	0.86	5.43

Computationally, our approach is tractable which can be seen in Table 1 for running times. These running times are on an Intel Xeon 2.4 GHz processor with 64 GB memory.

The graph deduction algorithm itself is robust due to multi-stage nature of the algorithm. In designing transfer functions, one of the tedious tasks is to look for an intensity value corresponding to the background material. We simplify and automate background removal by detecting the same using minimum of occlusion and intensity values and suppressing the entire background cluster. In our renderer, we give the user flexibility to change colour and opacity of any detected material via our simplified material graph widget. Such a simplified interface gives semantic control to volume rendering and hides the complexity of designing a full-fledged TF with high dimensional controls.

We observed that classical clustering approaches applied directly to intensity values do not provide meaningful clusters for TF creation. On the other hand, volume segmentation approaches tend to be time consuming. Our algorithm performs local optimisation to search for an optimal material graph which yields a robust graph even in the presence of noise. Presented approach is near real-time and gives a reasonable material definition. The most challenging part of our approach is material deduction that merges and connects interface edges. The input to this stage requires well separated interface edges. A good clustering algorithm is crucial in identifying these edges from a feature space like LH.

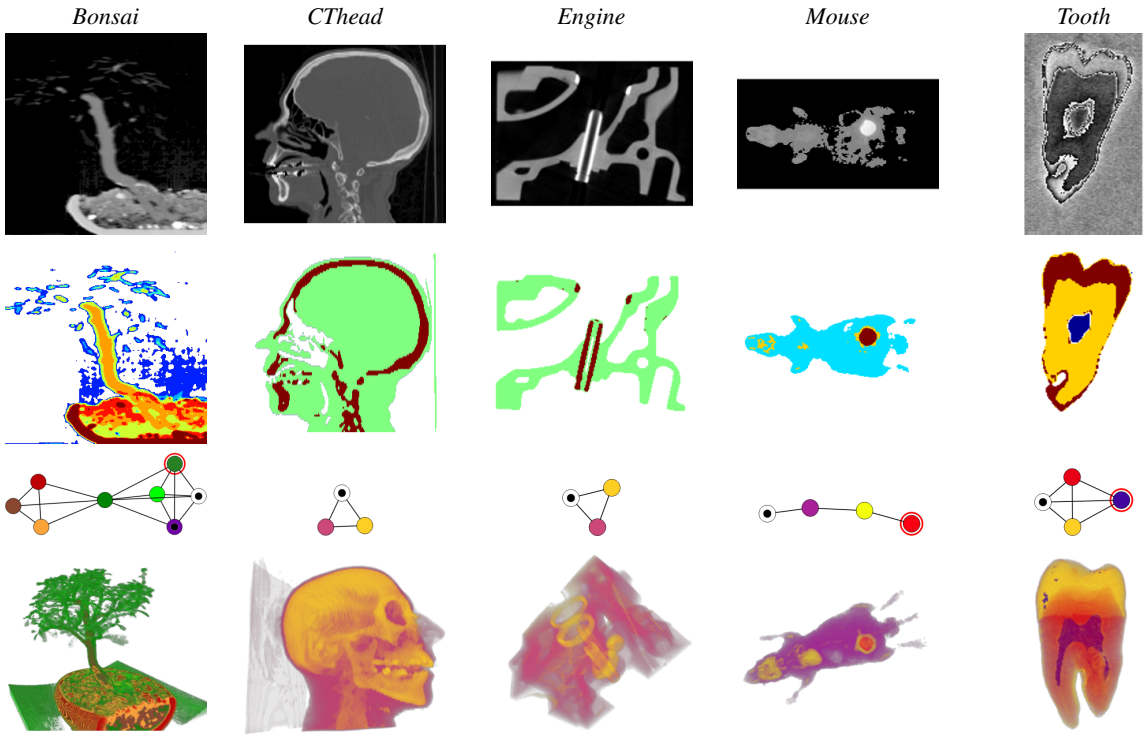

Figure 3: *Results of our automatic volume rendering using material graphs. For each sample data we show the original volume cross-section, materials identified in the cross-section, deduced material graph, and a color volume visualization.*

5. Conclusions

With this work, we have presented a robust approach to simultaneously create material definitions and their connectivity graph in a volume. These two pieces of information are crucial in designing high fidelity interactive TFs. Our TF creation reduces the space of parameters involved in its design. Further our results indicate that the algorithm is capable of segmenting the materials very well and resolution of detection is constrained only by the amount of subsampling performed.

References

[Bha43] BHATTACHARYYA A.: On a measure of divergence between two statistical populations defined by their probability distribution. *Bull. Calcutta Math. Soc* (1943). 2

[CM09] CORREA C., MA K.-L.: The occlusion spectrum for volume classification and visualization. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1465–1472. 1

[CMS13] CAMPELLO R. J., MOULAVI D., SANDER J.: Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining* (2013), Springer, pp. 160–172. 2

[CNCO17] CAI L., NGUYEN B. P., CHUI C.-K., ONG S.-H.: A two-level clustering approach for multidimensional transfer function specification in volume visualization. *The Visual Computer 33*, 2 (2017), 163–177. 1

[CTN*13] CAI L., TAY W.-L., NGUYEN B. P., CHUI C.-K., ONG S.-H.: Automatic transfer function design for medical visualization using visibility distributions and projective color mapping. *Computerized Medical Imaging and Graphics 37*, 7 (2013), 450–458. 1

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on visualization and computer graphics 8*, 3 (2002), 270–285. 1

[LKG*16] LJUNG P., KRÜGER J., GROLLER E., HADWIGER M., HANSEN C. D., YNNERMAN A.: State of the art in transfer functions for direct volume rendering. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 669–691. 1

[MWCE09] MACIEJEWSKI R., WOO I., CHEN W., EBERT D.: Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1473–1480. 1

[QYH15] QIN H., YE B., HE R.: The voxel visibility model: an efficient framework for transfer function design. *Computerized Medical Imaging and Graphics 40* (2015), 138–146. 1

[Roe12] ROETTGER S.:. The volume library [online]. 2012. URL: http://schorsch.efi.fh-nuernberg.de/data/volume/. 3

[ŠBSG06] ŠEREDA P., BARTROLI A. V., SERLIE I. W., GERRITSEN F. A.: Visualization of boundaries in volumetric data sets using LH histograms. *IEEE Transactions on Visualization and Computer Graphics 12*, 2 (2006), 208–218. 1, 2

[SS15] SOUNDARARAJAN K. P., SCHULTZ T.: Learning probabilistic transfer functions: A comparative study of classifiers. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 111–120. 1

[WZL*12] WANG Y., ZHANG J., LEHMANN D. J., THEISEL H., CHI X.: Automating transfer function design with valley cell-based clustering of 2D density plots. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1295–1304. 1