An efficient transfer function design method for volume rendering based on density clustering and dimensionality reduction

SIBGRAPI Paper ID: 99999

***Abstract*—Transfer functions (TFs) are a fundamental component of volume visualization and have been extensively studied in the context of Direct Volume Rendering (DVR). In the traditional DVR pipeline, TFs serve two main roles: material classification and mapping data values to optical properties. The effectiveness of a TF is closely tied to the characteristics of the underlying data. Although multidimensional TFs offer enhanced classification capabilities, defining them remains a complex task, particularly when emphasizing specific volume features. This paper presents an intuitive TF design method that facilitates both TF definition and volume exploration. The method combines dimensionality reduction, clustering and representative selection to identify features of interest within a volume, while ensuring computational efficiency suitable for large volume datasets. Additionally, our approach includes a user-friendly exploration workflow based on an initial TF definition and an enhanced 2D scatterplot interface for interactive visualization.**

I. Introduction

Direct Volume Rendering (DVR) is widely used in scientific and medical applications to visualize three-dimensional scalar data. A key element in DVR is the transfer function (TF), which maps volume data attributes (e.g., density) to visual properties such as color and opacity [1].

Multidimensional TFs can enhance data classification, but their design becomes increasingly complex with more input attributes [1], [2]. Since no universal TF fits all datasets, design is often manual and highly dependent on user expertise [3]. Dimensionality and non-intuitive parameter spaces further complicate the process.

We propose a low-cost, unsupervised approach to simplify TF design. Our method combines clustering, dimensionality reduction, and pivot-based indexing to support semi-automated classification. A modified 2D scatter plot interface facilitates intuitive exploration of volume features.

Our main contributions are:

- A computationally efficient method for generating TFs with minimal manual adjustment.
- A user-friendly exploration interface for interacting with classified volume data.

The remainder of this paper is organized as follows: Section II reviews related work; Section III presents our method; Section **??** describes the exploration interface; results and discussion are given in Sections V and VI; and Section VII concludes the paper.

II. Related Works

Various aspects of transfer functions (TFs) have been extensively discussed in the literature [1]. Our review focuses on methods that support user interaction in multidimensional TF design, particularly those employing machine learning, dimensionality reduction, and information visualization techniques.

A typical multidimensional TF incorporates either multivariate or derived input data. Multivariate attributes originate from the volume acquisition process, while derived attributes are computed from primary data, such as density. The gradient is the most commonly used derived attribute, but others include curvature, size, distance, texture and statistical measures.

Selecting an optimal subset of attributes to maximize material classification accuracy is a complex task. [3] argue that no single TF design is universally effective. Considering all available attributes is impractical, as it may increase computational costs and introduce noise, degrading classification performance. This classical challenge is known as the "curse of dimensionality." Dimensionality reduction is the most common approach to address this issue and several studies have applied such techniques to multidimensional TF design [4]–[8].

Histograms are commonly used as componente for 2D TFs design [9], typically representing intensity–gradient magnitude or low–high histograms. Several approaches have been proposed to automate histogram-based TF design. Röttger et al. [10] group spatially connected regions and associate gradient values with spatial coordinates to classify datasets. Many methods combine histograms with clustering algorithms, including affinity propagation [11], hierarchical clustering [12], and iterative self-organizing data analysis [13].

Approaches to multidimensional TF design generally follow two main strategies. The first provides an interface that allows users to manipulate all data attributes—such as the parallel coordinate plot (PCP). The second applies dimensionality reduction techniques, such as Multidimensional Scaling (MDS) or Principal Component Analysis (PCA), to create simplified visual representations.

[14] employed PCP in their exploration scheme, integrating viewer parameters and TF specification into the interface. [8] followed a similar approach, applying a local linear embedding technique for dimensionality reduction. Likewise, [15] proposed a hybrid interface that combines PCP with a scatter plot generated using MDS.

[7] applied a Self-Organizing Map (SOM) and a radial basis function for TF design. SOM performs unsupervised

learning to reduce dimensionality, producing a map in which neighboring regions represent similar voxels. Users interact with the map by drawing widgets in specific regions. [16] extended this idea using a spherical SOM, enabling interaction on a spherical lattice. [17] proposed a volume exploration space based on subtree structures derived from hierarchical clustering and modified dendrograms. Later, [4] revisited these ideas, augmenting SOM with a normalized cut step to create a "cell map," in which each region encodes volume information tied to meaningful structures. Our method shares similarities with the work of [4], but employs an MDS-based technique and a density-based clustering algorithm to automate material classification, resulting in a modified scatter plot view.

[18] proposed one of the earliest TF design strategies using supervised learning, implementing neural networks and support vector machines. [19] combined SOM with back-propagation neural networks for material classification. [20] used a Generative Adversarial Network (GAN) framework to compute models for TF specification and view selection. More recently, [21] combined GANs with Convolutional Neural Networks (CNNs) to synthesize the exploration process. [22] developed a method using CNNs to generate visualizations from TF colorizations. [2] introduced a deep learning-based gallery approach with differentiable rendering to support user exploration of the design space.

[23] proposed a graph-based method for identifying significant volume structures, which involves clustering features, building a material graph topology, and enhancing the rendering of important structures.

III. Method

This section presents our unsupervised method for transfer function (TF) design, enabling semi-automated material classification and initial TF specification to support intuitive volume exploration.

An overview is shown in Fig. 1. After organizing the multi-dimensional data into a volume grid, our method comprises three main steps: dimensionality reduction, clustering and representative selection.

The techniques used at each step were carefully chosen based on their time complexity, ensuring a balance between computational efficiency and effectiveness for handling large volume datasets. This design choice supports practical scalability and responsiveness, which are critical for interactive volume exploration.

A. Dimensionality reduction

Dimensionality reduction is a critical step for two reasons. First, the clustering technique employed requires a two-dimensional input space for proper functioning (see Section III-B). Second, our TF design interface is two-dimensional (see Section IV).

We adopt FastMap [24] to project high-dimensional voxel data into 2D while preserving data similarity. FastMap operates by selecting two distant pivots and projecting all points onto the line defined by these pivots, recursively reducing the dimensionality.

Let d be the number of attributes and n the number of voxels. The algorithm proceeds as follows:

1) Select two points with maximal pairwise distance as the pivots.
2) Project all points onto a hyperplane orthogonal to the line defined by the pivots.

To mitigate the computational cost of pivot selection, [24] proposed the approach summarized in Algorithm 1.

Algorithm 1: Pivot searching in FastMap.

Input: $\mathbb{O}$
Output: Pivots O_a, O_b
$O_a \leftarrow$ random point $o \in \mathbb{O}$
$O_b \leftarrow$ point $o \in \mathbb{O}$ farthest from O_a
$O_a \leftarrow$ point $o \in \mathbb{O}$ farthest from O_b

B. Clustering

To simplify material classification and enhance the detection of relevant volume structures, we employ the classical density-based clustering technique DBSCAN [25].

DBSCAN is widely recognized for its effectiveness in identifying clusters of arbitrary shapes without requiring prior knowledge of the number of clusters [26]. However, its standard implementation has a worst-case time complexity of $\mathcal{O}(n^2)$ [26]. To ensure practical scalability, we adopt an optimized grid-based variant [27], which reduces the time complexity to $\mathcal{O}(n \log n)$.

As in the original algorithm, two parameters must be tuned by the user: $minPts$, the minimum number of points to form a dense region, and ε, the neighborhood radius.

At the end of this step, each cluster contains a subset of voxels potentially representing distinct features of interest (FOI) within the volume.

C. Representative selection

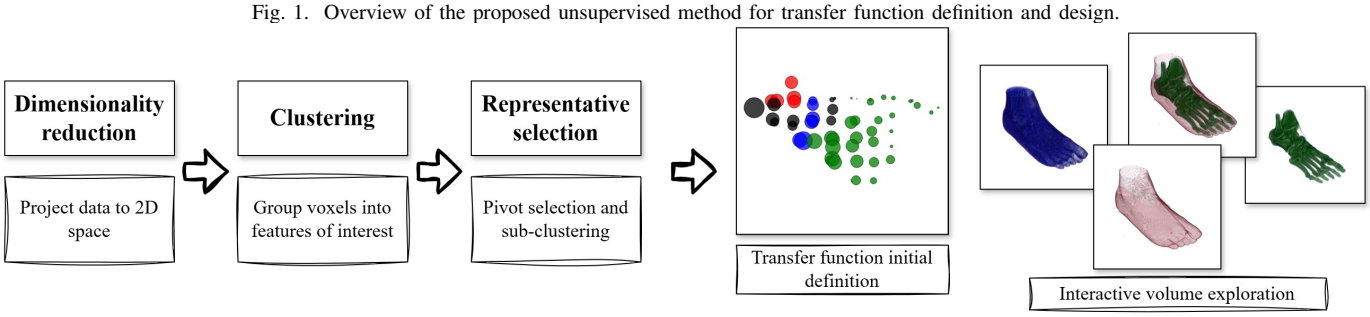
The voxels classified by DBSCAN could be directly projected onto the TF design interface. However, to reduce clutter in the scatter plot and improve interpretability, we apply a representative selection technique within each cluster.

First, representative pivots are selected using Sparse Spatial Selection (SSS) [29]. This technique iteratively adds points as pivots if they are sufficiently distant from all previously selected pivots, controlled by a distance factor α (Algorithm 2).

Next, each cluster is subdivided into sub-clusters by assigning every point to its nearest pivot (Algorithm 3). This step refines the initial data classification and acts as a second-level clustering, improving the granularity of the FOI representation.

The parameter α controls the number of selected pivots: smaller values lead to more pivots and finer sub-clustering, while values closer to 1 yield fewer, broader sub-clusters.

Fig. 1. Overview of the proposed unsupervised method for transfer function definition and design.



Algorithm 2: Sparse Spatial Selection (SSS).

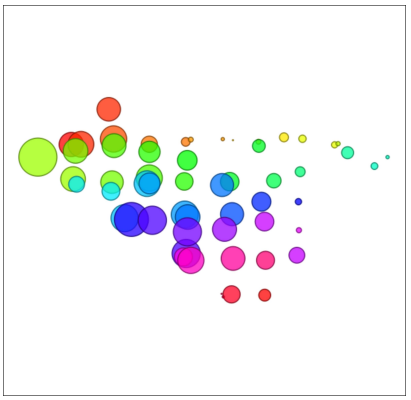
Input: Points $\mathbb{P}$
Output: Selected pivots $\mathbb{P}_s$

$\mathbb{P}_s \leftarrow \{p_1\}$
foreach $p \in \mathbb{P}$ **do**
  if $\forall p_s \in \mathbb{P}_s, dist(p, p_s) \geq M\alpha$ **then**
    $\mathbb{P}_s \leftarrow \mathbb{P}_s \cup \{p\}$
  end
end

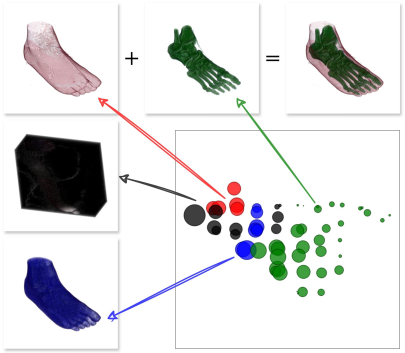
Algorithm 3: Sub-cluster assignment within a cluster.

Input: Points $\mathbb{P}$ of cluster c
Input: Pivots $\mathbb{P}_s$ of cluster c
Output: Points with sub-cluster assignments

foreach $p \in \mathbb{P}$ **do**
  $p_s \leftarrow$ nearest pivot in $\mathbb{P}_s$
  Assign p to p_s's sub-cluster
end



(a) Initial transfer function definition.



(b) Fine-tuned transfer function definition after user adjustment and identified volume features of interest.

Fig. 2. Transfer function design interface and interactive volume exploration of a right male foot dataset.

IV. Volume Exploration

Figure 2 illustrates our TF design interface, a 2D scatter plot where each point represents a feature of interest (FOI). The coordinates of the points are determined by the FastMap projection. Each point corresponds to a pivot (the centroid of a cluster), with its radius proportional to the number of voxels it represents, normalized logarithmically. It is important to highlight that each pivot results from a two-level clustering process, generated by the combination of SSS and DBSCAN.

Our method generates an initial TF definition using a predefined opacity and a rainbow color scale, assigning a unique color to each cluster.

Users adjust the TF definition following the WYSIWYG principle: the pivot's color and opacity directly map to their associated voxels according to clustering. Both selected and unselected elements are fully customizable.

Volume exploration is performed through pivot selection. The system dynamically increases the opacity of selected pivots while decreasing that of others. Users can make arbitrary selections, save them as groups, and interact with pivots, clusters, or groups as selectable entities.

Iterative selection of nearby elements facilitates detailed inspection of volume structures. FastMap and DBSCAN naturally cluster spatially similar FOIs, simplifying this process.

Our approach automates material classification by assuming each cluster or pivot corresponds to a relevant item. If users are unsatisfied, they may select or deselect elements or adjust the following parameters:

- input volume data,
- DBSCAN parameters ε and $minPts$,
- SSS distance factor α.

V. Results

A. Experimental Design

Experiments ran on an Intel Core i5-7200U, 8 GB RAM, Ubuntu 22.04 64-bit, with an NVIDIA GeForce GT 940MX GPU.

We used classical volume ray-casting with Blinn-Phong illumination and trilinear interpolation. Ray step adjusted per voxel spacing. Runtimes are averages of five trials.

The system was implemented in C++ using Qt and CUDA C/C++. The implementation is publicly available.

Table I lists the volume datasets used in experiments.

All volumes contain scalar density values. Multidimensional attributes were derived—13 in total—including intensity, gradient magnitude, Laplacian magnitude, and 10 local histogram statistics (absolute deviation, contrast, energy, entropy, inertia, kurtosis, mean, skewness, standard deviation, variance).

Attribute selection was empirical, tailored per dataset to balance discrimination and computational cost.

TABLE I
VOLUME DATASETS.

Dataset	Grid size	Total voxels
Engine block	$256 \times 256 \times 256$	16,777,216
Knees	$379 \times 229 \times 305$	26,471,255
Tooth	$256 \times 256 \times 161$	10,551,296

B. Runtime

Table II presents runtimes (seconds) for each dataset.

TABLE II
RUNTIME (SECONDS) OF THE PROPOSED METHOD PER DATASET.

	Engine block	Knees	Tooth
Dimensionality reduction	7.50	7.98	36.05
Clustering	51.52	102.77	19.42
Pivot-based indexing	2.23	3.15	1.33
Volume exploration space	1.48	1.86	0.79

C. Data classification

Attribute selection was done empirically by iterative testing and visual assessment, ensuring effective separation of volumetric structures while maintaining computational feasibility.

DBSCAN's $minPts$ was fixed at 4 [25]. Parameter ε varied within $[0.2, 0.35]$, and SSS parameter α within $[0.8, 0.95]$ for volume exploration.

1) Engine block dataset: Figure 3 shows the volume exploration space for the engine block. Each numbered group corresponds to classified volume details in Fig. 4. Parameters: $k = 4$, TF = {intensity, skewness, gradient magnitude, variance}; $minPts = 4$; $\varepsilon = 0.35$; $\alpha = 0.85$.

Figure 5 shows a volume exploration simulation revealing engine components, starting from Fig. 3.

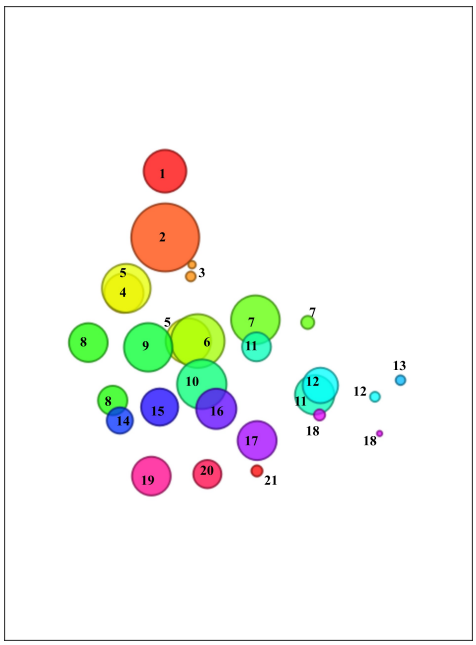


Fig. 3. Volume exploration space for engine block dataset. Parameters: TF = {intensity, skewness, gradient magnitude, variance}; $minPts = 4$; $\varepsilon = 0.35$; $\alpha = 0.85$.

Fig. 4. Rendered classified volume details for engine block. Parameters as in Fig. 3.

2) Knees dataset: Preliminary classification is shown in Fig. 6 with rendered details in Fig. 7. Parameters: TF = {intensity, variance, absolute deviation, energy, contrast}; $minPts = 4$; $\varepsilon = 0.35$; $\alpha = 0.9$.

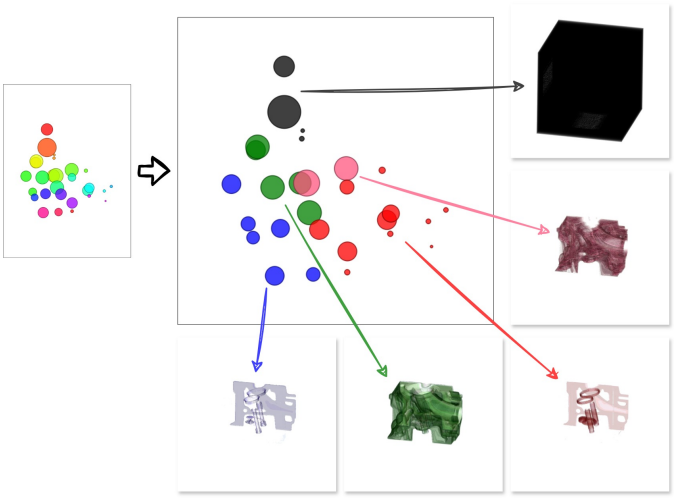


Fig. 5. User-refined transfer function and volume classification for engine block. Groups formed empirically. Parameters as in Fig. 3.

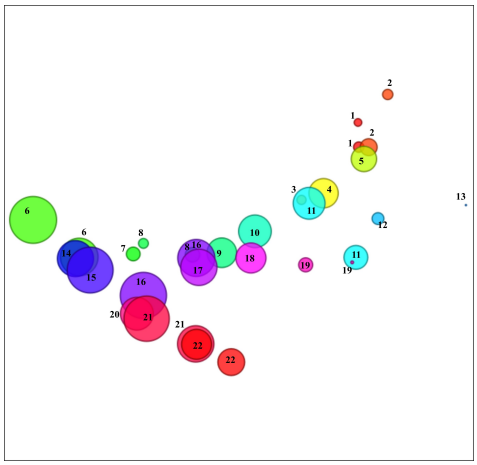


Fig. 6. Volume exploration space for knees dataset. Parameters as above.

Figure 8 illustrates grouped bones and muscles: femur, tibia, patella, fibula, thigh and knee muscles.

3) Tooth dataset: Figure 10 shows the volume exploration space, with rendered details in Fig. 10. Parameters: TF = {intensity, variance, absolute deviation, energy, contrast, entropy}; $minPts = 4$; $\varepsilon = 0.23$; $\alpha = 0.9$.

Figure 11 shows empirically grouped tooth structures: enamel, pulp, dentin, crown, entire tooth, and immersion fluid.

VI. Discussion

The dimensionality reduction approach effectively addresses common challenges in transfer function design by simplifying the data representation. The choice of DBSCAN parameters strongly affects classification outcomes. The $minPts$ parameter can reliably use a default value of 4 [25], given that FastMap reduces the data to a 2D space. However, the ε parameter requires careful tuning: larger values produce fewer but larger clusters, while smaller values result in more numerous, smaller clusters.

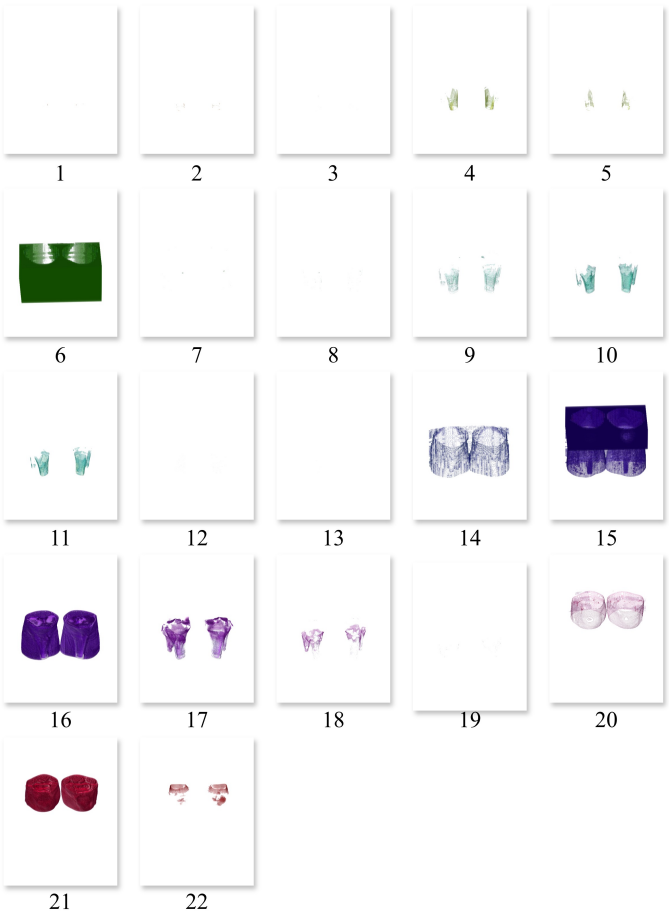


Fig. 7. Rendered classified volume details for knees dataset. Parameters as above.

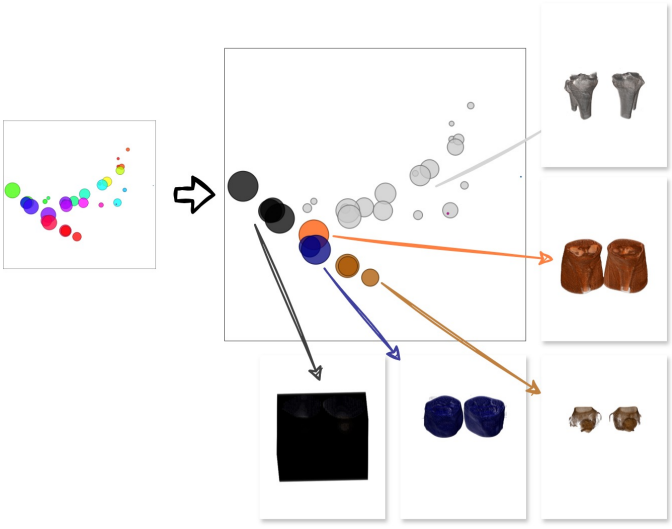


Fig. 8. User-refined TF and classification for knees dataset. Groups formed empirically. Parameters as above.

The SSS distance factor (α) similarly influences clustering granularity, varying inversely with the number of pivots per cluster.

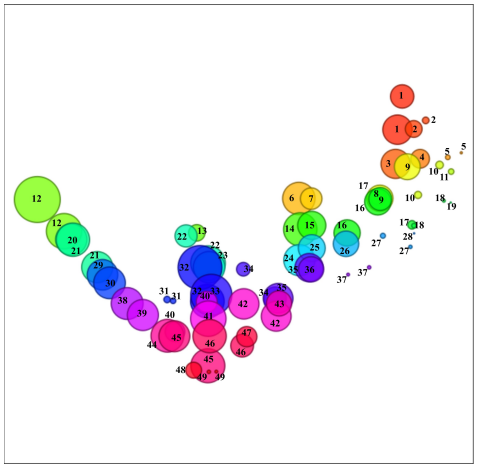


Fig. 9. Volume exploration space for tooth dataset. Parameters as above.

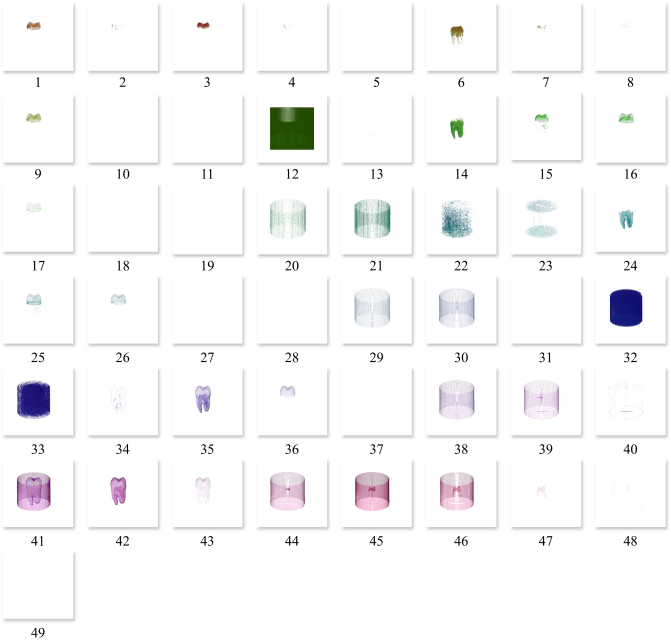


Fig. 10. Rendered classified volume details for tooth dataset. Parameters as above.

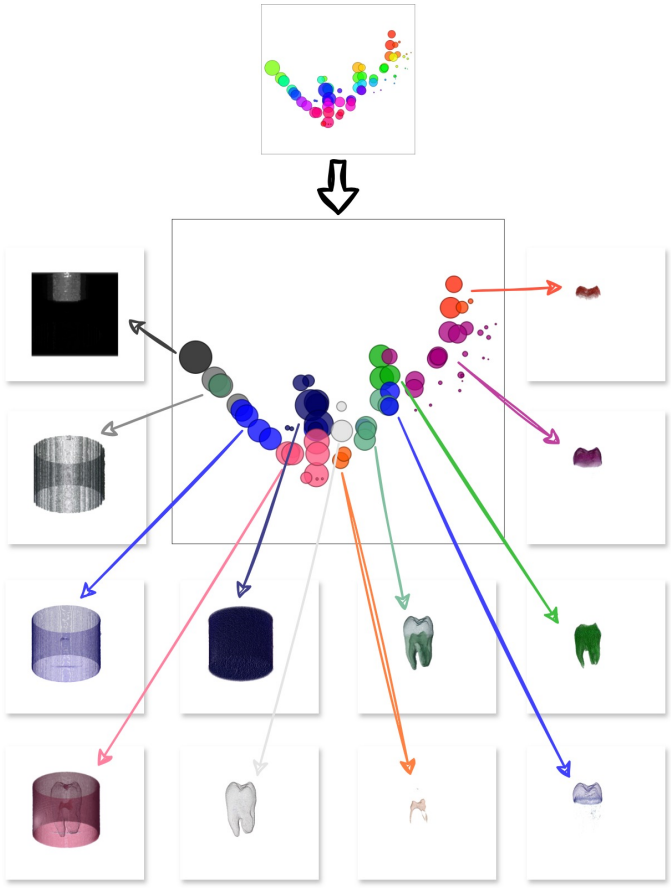


Fig. 11. User-refined TF and classification for tooth dataset. Groups formed empirically. Parameters as above.

Overall, the method incurs minimal computational overhead, demonstrating efficient performance and promising scalability for large volumetric datasets.

VII. Conclusions

We presented a robust method for transfer function (TF) design that integrates feature classification with a simplified volume exploration space. Our approach combines FastMap for efficient feature extraction, DBSCAN for effective clustering, and SSS for pivot-based indexing. These techniques enable semi-automatic classification and initial TF specification, visualized through an intuitive scatter plot interface for volume exploration.

The method exhibits low computational overhead, short runtimes, and minimal storage requirements, highlighting its practicality and scalability for real-world applications. Future work will focus on assessing the method's performance with large, high-dimensional datasets to validate its scalability and effectiveness.

Moreover, we aim to extend our evaluation to multivariate data, enhancing the method's applicability and robustness across a wider range of volume datasets.

Acknowledgements

This work was partially supported by the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

[1] P. Ljung, J. Krüger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, "State of the art in transfer functions for direct volume rendering," in *Computer graphics forum*, vol. 35, no. 3. Wiley Online Library, 2016, pp. 669–691.

[2] B. Pan, J. Lu, H. Li, W. Chen, Y. Wang, M. Zhu, C. Yu, and W. Chen, "Differentiable design galleries: A differentiable approach to explore the design space of transfer functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 1369–1379, 2024.

[3] S. Arens and G. Domik, "A survey of transfer functions suitable for volume rendering." in *VG@ Eurographics*, 2010, pp. 77–83.

[4] L. Cai, B. P. Nguyen, C.-K. Chui, and S.-H. Ong, "A two-level clustering approach for multidimensional transfer function specification in volume visualization," *Vis. Comput.*, vol. 33, no. 2, p. 163–177, feb 2017. [Online]. Available: https://doi.org/10.1007/s00371-015-1167-y

[5] A. Abbasloo, V. Wiens, M. Hermann, and T. Schultz, "Visualizing tensor normal distributions at multiple levels of detail," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 975–984, 2016.

[6] Y. Gao, C. Chang, X. Yu, P. Pang, N. Xiong, and C. Huang, "A vr-based volumetric medical image segmentation and visualization system with natural human interaction," *Virtual Real.*, vol. 26, no. 2, p. 415–424, jun 2022. [Online]. Available: https://doi.org/10.1007/s10055-021-00577-4

[7] F. D. M. Pinto and C. M. D. S. Freitas, "Design of multi-dimensional transfer functions using dimensional reduction," in *Proceedings of the 9th Joint Eurographics/IEEE VGTC conference on Visualization*, 2007, pp. 131–138.

[8] X. Zhao and A. Kaufman, "Multi-dimensional reduction and transfer function design using parallel coordinates," in *Volume graphics. International Symposium on Volume Graphics*. NIH Public Access, 2010, p. 69.

[9] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *IEEE Transactions on visualization and computer graphics*, vol. 8, no. 3, pp. 270–285, 2002.

[10] S. Roettger, M. Bauer, and M. Stamminger, "Spatialized transfer functions," in *Proceedings of the Seventh Joint Eurographics / IEEE VGTC Conference on Visualization*, ser. EUROVIS'05. Goslar, DEU: Eurographics Association, 2005, p. 271–278.

[11] T. Zhang, Z. Yi, J. Zheng, D. C. Liu, W.-M. Pang, Q. Wang, J. Qin *et al.*, "A clustering-based automatic transfer function design for volume visualization," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[12] P. Sereda, A. Vilanova, and F. A. Gerritsen, "Automating transfer function design for volume rendering using hierarchical clustering of material boundaries." in *EuroVis*, 2006, pp. 243–250.

[13] F.-Y. Tzeng and K.-L. Ma, "A cluster-space visual interface for arbitrary dimensional classification of volume data," in *Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization*, ser. VISSYM'04. Goslar, DEU: Eurographics Association, 2004, p. 17–24.

[14] M. Tory, S. Potts, and T. Moller, "A parallel coordinates style interface for exploratory volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1, pp. 71–80, 2005.

[15] H. Guo, H. Xiao, and X. Yuan, "Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates," in *2011 IEEE Pacific Visualization Symposium*. IEEE, 2011, pp. 19–26.

[16] N. M. Khan, M. Kyan, and L. Guan, "Intuitive volume exploration through spherical self-organizing map and color harmonization," *Neurocomputing*, vol. 147, pp. 160–173, 2015.

[17] L. Wang, X. Zhao, and A. E. Kaufman, "Modified dendrogram of attribute space for multidimensional transfer function design," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 1, pp. 121–131, 2011.

[18] F.-Y. Tzeng, E. B. Lum, and K.-L. 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, 2005.

[19] L. Wang, X. Chen, S. Li, and X. Cai, "General adaptive transfer functions design for volume rendering by using neural networks," in *International Conference on Neural Information Processing*. Springer, 2006, pp. 661–670.

[20] M. Berger, J. Li, and J. A. Levine, "A generative model for volume rendering," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 4, pp. 1636–1650, 2018.

[21] F. Hong, C. Liu, and X. Yuan, "Dnn-volvis: Interactive volume visualization supported by deep neural network," in *2019 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2019, pp. 282–291.

[22] S. Kim, Y. Jang, and S.-E. Kim, "Image-based tf colorization with cnn for direct volume rendering," *IEEE Access*, vol. 9, pp. 124 281–124 294, 2021.

[23] O. Sharma, T. Arora, and A. Khattar, "Graph-based transfer function for volume rendering," in *Computer Graphics Forum*, vol. 39, no. 1. Wiley Online Library, 2020, pp. 76–88.

[24] C. Faloutsos and K.-I. Lin, "Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, 1995, pp. 163–174.

[25] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[26] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[27] A. Gunawan and M. de Berg, "A faster algorithm for dbscan," *Master's thesis*, 2013.

[28] J. Gan and Y. Tao, "Dbscan revisited: Mis-claim, un-fixability, and approximation," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 519–530.

[29] O. Pedreira and N. R. Brisaboa, "Spatial selection of sparse pivots for similarity search in metric spaces," in *International Conference on Current Trends in Theory and Practice of Computer Science*. Springer, 2007, pp. 434–445.