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 provides 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 3D 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 as more input attributes are considered [1, 2]. Since no universal TF fits all datasets, the design process is often manual and highly dependent on user expertise. The high dimensionality and non-intuitive parameter spaces further complicate this task.

We propose a computationally efficient TF design method that combines clustering, dimensionality reduction and representative selection. Our approach provides:

- A simplified TF design interface with an initially generated TF definition.
- A semi-automated identification of features of interest (FOIs) based on volume data.
- An interactive exploration interface based on classified volume FOIs.

The remainder of this paper is organized as follows: Section II reviews related work. Section III presents our method. Section IV describes the TF design interface. Results are discussed in Sections V. Finally, Section VI concludes the paper.

II. Related Works

Various aspects of TFs have been extensively discussed in the literature [1]. Our review focuses on strategies for managing the complexity of defining multidimensional TFs, particularly those involving machine learning, dimensionality reduction, and information visualization.

Histograms are frequently used in 2D TF design, typically representing intensity–gradient magnitude. Automated techniques often combine histograms with clustering algorithms such as affinity propagation [3], hierarchical clustering [4], and iterative self-organizing methods [5]. Roettger et al. [6] proposed classifying spatially connected regions using gradient and coordinate information.

Multidimensional TF design commonly follows two strategies: (i) interactive interfaces enabling direct manipulation of attributes, such as parallel coordinate plots (PCP); and (ii) dimensionality reduction techniques like MDS or PCA, producing simplified visual representations [7, 8, 9].

Self-Organizing Maps (SOM) have been used to reduce dimensionality and create interactive maps [10]. Extensions include spherical SOMs [11], hierarchical clustering with modified dendrograms [12], and normalized cuts forming cell maps [13]. Our method builds on [13] by employing MDS and density-based clustering to automate material classification.

Supervised learning has also been explored: neural networks and SVMs [14], SOM with backpropagation [15], and deep learning approaches using GANs and CNNs for TF generation and visualization [2, 16, 17, 18]. Sharma et al. [19] proposed a graph-based method to identify and highlight significant volume structures through feature clustering and topology analysis.

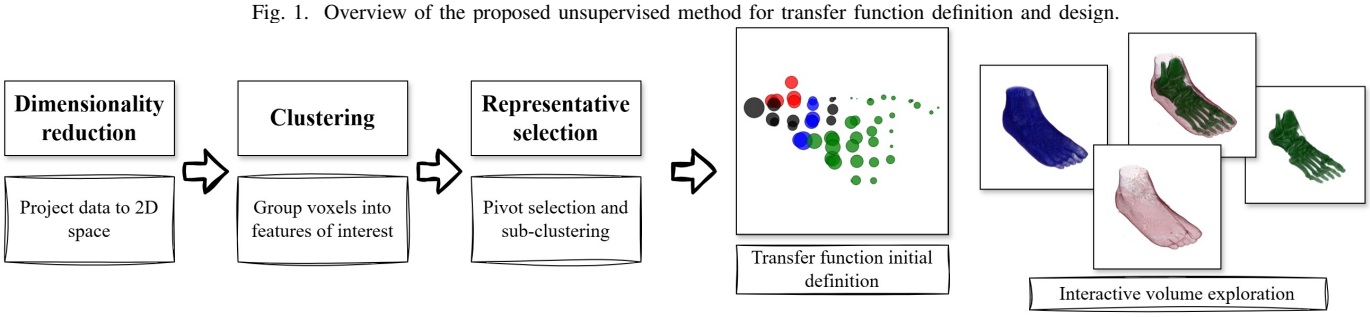
III. Method

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 multidimensional 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.

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



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 [20] 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, [20] 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 [21].

DBSCAN is widely recognized for its effectiveness in identifying clusters of arbitrary shapes without requiring prior knowledge of the number of clusters [22]. However, its standard implementation has a worst-case time complexity of $\mathcal{O}(n^2)$ [22]. To ensure practical scalability, we adopt an optimized grid-based variant [23], 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 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) [24]. 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).

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

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.

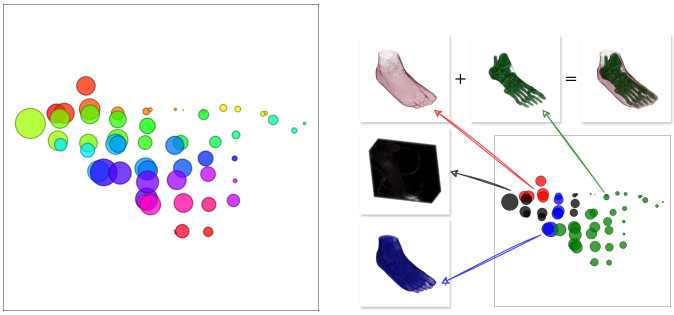
Algorithm 3: Sub-cluster assignment within a cluster.

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

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.

IV. Volume Exploration

Figure 2 illustrates our TF design interface: a 2D scatter plot where each point represents a volume 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.



(a) Initial transfer function definition.

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

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

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 were conducted on a computer with the following specifications: Intel Core i5-7200U, 8 GB RAM, Ubuntu 22.04 64-bit, and an NVIDIA GeForce GT 940MX GPU.

We used classical volume ray-casting with Blinn-Phong illumination and trilinear interpolation. The ray step was adjusted according to voxel spacing. Runtimes reported are averages of five trials.

Table I lists the volume datasets used in the experiments.

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. Volume eploration

All datasets originally contain only scalar density (intensity) values as their primary attribute. From the original data, 12 additional multidimensional attributes were systematically derived to enrich the feature space, including gradient magnitude, Laplacian magnitude, and 10 local histogram statistics (absolute deviation, contrast, energy, entropy, inertia, kurtosis, mean, skewness, standard deviation, and variance).

For each dataset, a specific subset of k attributes was empirically selected through iterative experimentation and visual assessment to compose the TF. These selected groups of attributes aimed to balance the discrimination power of volume structures with computational efficiency. Notably, the attribute selection process varied between datasets, with distinct groups of k attributes chosen for each case.

The DBSCAN parameter $minPts$ was fixed at 4 [21]. The parameter ε varied within the range $[0.2, 0.35]$, and the SSS parameter α was varied within $[0.8, 0.95]$ for volume exploration.

1) Engine block dataset: Figure 3 shows the volume exploration space for the engine block dataset. For this dataset, a set of $k = 4$ attributes was selected: {intensity, skewness, gradient magnitude and variance}. Each numbered group corresponds to a volume feature in Fig. 4. Method parameters: $minPts = 4$, $\varepsilon = 0.35$ and $\alpha = 0.85$.

Figure 5 shows a volume exploration simulation revealing engine components, based on the selected attribute subset.

2) Knees dataset: For the knees dataset, a set of $k = 5$ attributes was selected: {intensity, variance, absolute deviation, energy and contrast}. Preliminary classification is shown in Fig. 6, with rendered details in Fig. 7. Method parameters: $minPts = 4$, $\varepsilon = 0.35$ and $\alpha = 0.9$.

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

3) Tooth dataset: For the tooth dataset, a set of $k = 6$ attributes was selected: {intensity, variance, absolute deviation, energy, contrast and entropy}. Figure 9 shows the volume exploration space, with rendered details in Fig. 10. Method parameters: $minPts = 4$, $\varepsilon = 0.23$ and $\alpha = 0.9$.

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

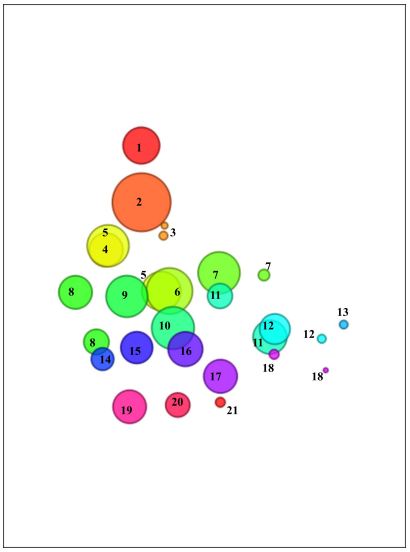


Fig. 3. Initial transfer function definition and volume features for the engine block dataset. Selected attributes: {intensity, skewness, gradient magnitude and variance} ($k = 4$). Method parameters: $minPts = 4$; $\varepsilon = 0.35$ and $\alpha = 0.85$.

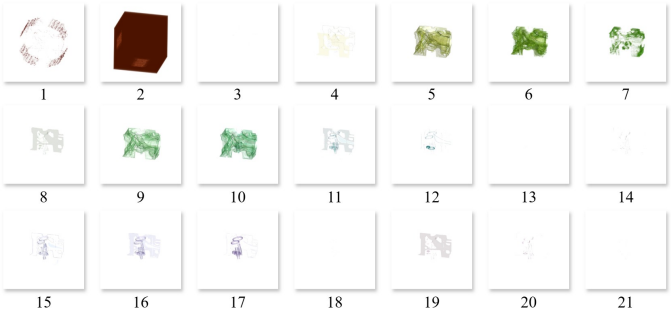


Fig. 4. Rendered volume features for the engine block dataset.

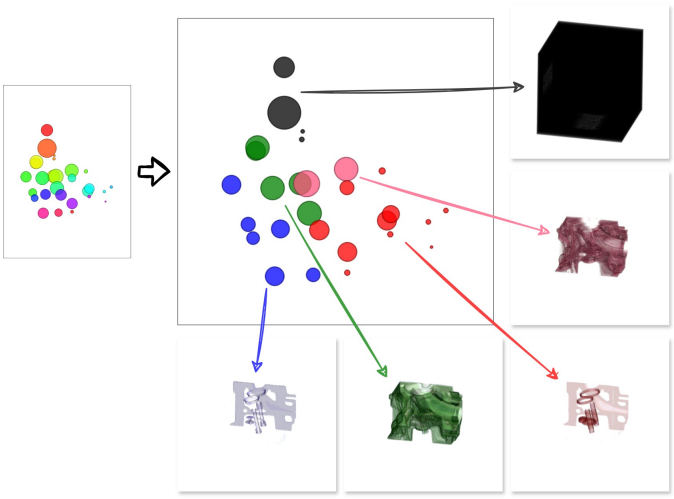


Fig. 5. User-refined transfer function definition and volume features of interest for block dataset.

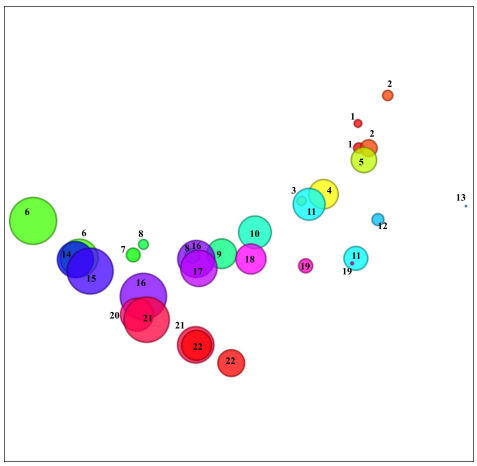


Fig. 6. Initial transfer function definition and volume features for the knees dataset. Selected attributes: {intensity, variance, absolute deviation, energy, contrast} ($k = 5$). Method parameters: $minPts = 4$, $\varepsilon = 0.35$ and $\alpha = 0.9$.

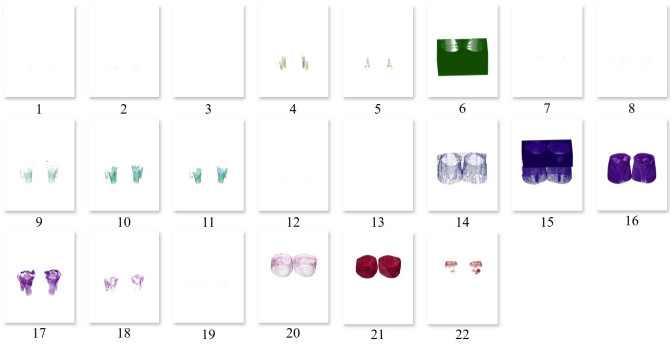


Fig. 7. Rendered volume features for the knees dataset.

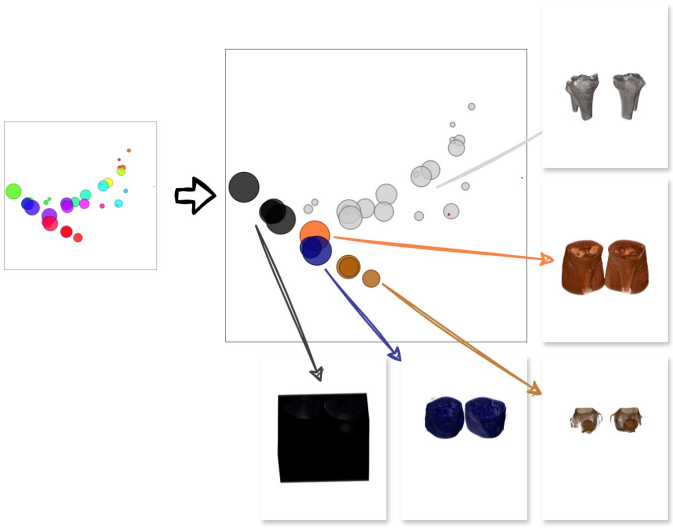


Fig. 8. User-refined transfer function definition and volume features of interest for knees dataset.

C. Runtime

Table II presents the runtimes (in seconds) for each dataset. Dimensionality reduction and clustering are the most time-consuming steps. The reduction time depends on dataset size and dimensionality, while clustering, representative selection,

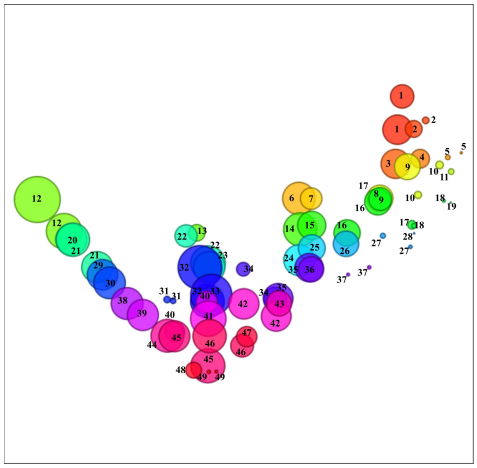


Fig. 9. Initial transfer function definition and volume features for the tooth dataset. Selected attributes: {intensity, variance, absolute deviation, energy, contrast and entropy} ($k = 6$). Method parameters: $minPts = 4$; $\varepsilon = 0.23$; $\alpha = 0.9$.

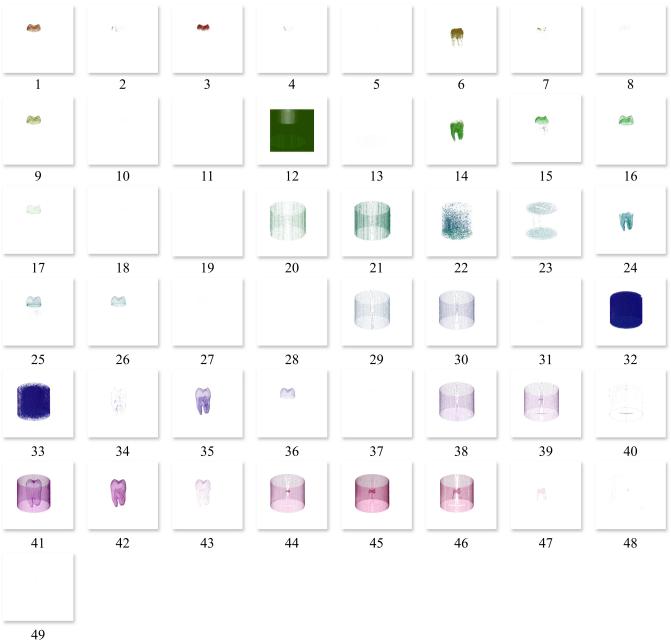


Fig. 10. Rendered volume features for the tooth dataset.

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
Representative selection	2.23	3.15	1.33
Transfer function design interface	1.48	1.86	0.79

and TF interface construction operate in 2D and depend mainly on the number of voxels.

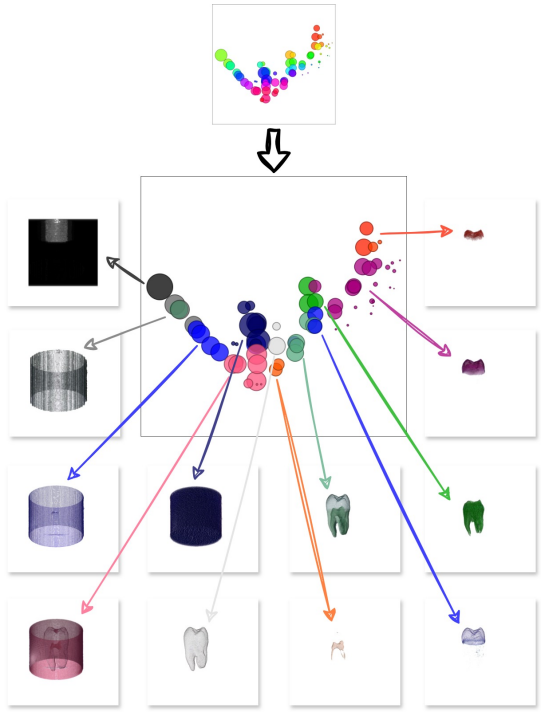


Fig. 11. User-refined transfer function definition and volume features of interest for tooth dataset.

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

D. Parameter choice

The choice of DBSCAN parameters strongly affects classification outcomes. The $minPts$ parameter can reliably use a default value of 4 [21], 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.

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

VI. Conclusions

We presented a robust method for TF design that provides semi-automated feature classification and a simplified volume interaction space. Our method exhibits low computational overhead, as confirmed by short runtimes, highlighting its practicality and scalability for real-world applications.

Future work will focus on assessing the performance of the method with large, high-dimensional datasets to further validate its scalability and effectiveness.

Moreover, we aim to extend our evaluation to multivariate data, enhancing the applicability and robustness of the method 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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[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] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.