Dimensionality reduction and density-based clustering for transfer function design in Direct Volume Rendering

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 multiple roles, primarily material classification and mapping data values to optical properties. The effectiveness of a TF is closely tied to the characteristics of the underlying data. While multidimensional TFs offer increased classification capabilities, their design remains a complex task, especially when aiming to emphasize specific volume features. This paper presents an unsupervised volume classification method that facilitates both the definition and design of TFs. The proposed approach combines dimensionality reduction, clustering techniques and pivot-based indexing to support the specification of meaningful transfer functions. The results include a user-friendly volume exploration workflow based on initial TF parameters, a semi-automated classification mechanism, and an enhanced 2D scatterplot interface for interactive data analysis.**

I. Introduction

Direct Volume Rendering (DVR) is a powerful technique employed in computer science for visualizing three-dimensional scalar data grids, particularly in scientific and medical applications [1], [2]. The transfer function (TF) is a central component of the DVR pipeline and the primary focus of this work. A TF maps volume data (such as density) to visual properties (such as color and opacity) in the rendered images [3].

When interacting with a DVR system, users often adjust TF parameters to reveal specific regions of interest within the dataset. A TF is a function in the mathematical sense, whose input consists of a set of volume data attributes. The accuracy of the resulting classification is directly influenced by the chosen data domain. Several studies [3]–[6] have demonstrated that utilizing multidimensional TFs can significantly enhance discriminative power. However, it is crucial to recognize that simply increasing the number of input attributes does not guarantee improved classification. There is no universal TF suitable for all datasets; consequently, TF design is usually left to the user's expertise and knowledge of the data domain [7].

Even after defining the data domain, adjusting TF parameters remains essential to highlighting desired volume details. TF design is inherently non-intuitive in one-dimensional spaces [5], [8], and this complexity intensifies in higher-dimensional scenarios. While material classification benefits from higher-dimensional data, specifying TFs in such spaces is notoriously difficult [3], [5], [9], [10].

We propose a low-computational-cost method that simplifies TF design, regardless of the dimensionality of the data domain. Our approach adopts an unsupervised-learning perspective, integrating clustering, dimensionality reduction, and pivot-based indexing. We also introduce an exploration scheme wherein users navigate through a set of volume details that are semi-automatically classified and mapped onto a modified 2D scatter plot view.

Our major contributions can be summarized as follows:

- An effective and low-computational-cost TF design approach. We employ semi-automated material classification to generate TFs that require minimal parameter adjustment.
- An intuitive volume exploration scheme. We provide a user-friendly scatter plot view for navigating the classified data.

A. Definition of concepts

In this paper, the term "feature selection" specifically refers to the group of dimensionality reduction (DR) techniques. Although some DVR-related works use the term "feature" to denote regions or structures of interest within a dataset, we avoid this terminology here to prevent ambiguity.

B. Paper organization

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed method. The volume exploration scheme and the TF design interface are detailed in Section IV. Section V presents the results, followed by their discussion in Section VI. Finally, the paper concludes in Section VII.

II. Related works

Various aspects concerning TFs have been thoroughly discussed in the literature [3]. Our review focuses on methods that support user interaction in multidimensional design, especially those that apply machine learning, dimensionality reduction, and information visualization views.

A. Multidimensional data

A typical multidimensional TF consists of multivariate or derived input data. Multivariate attributes are obtained from the volume acquisition process, while derived attributes are usually calculated from material density data. The gradient stands out as the most commonly utilized derived attribute, with other examples including curvature [11], [12], size [13], [14], distance [15], texture [16] and statistical measures [17].

Selecting an optimal subset of attributes to maximize material classification accuracy is complex. [7] contend that there is no universally suitable TF for all cases. Considering all available attributes is impractical, as it may escalate computational costs and introduce noise, thereby diminishing classification results. This classical problem is known as "the curse of dimensionality". Dimensionality reduction is the most common attempt to address this challenge. Several approaches have resorted to such techniques to deal with multidimensional TFs [4], [18]–[21].

B. Transfer function design

1) 2D transfer function design: Histograms usually serve as the user interface for 2D TFs [9]. These interfaces generally represent the intensity-gradient magnitude and the low-high histograms. Various methods for automating histogram-based TF design have been proposed. [6] group spatially connected regions and associated gradient values with space coordinates to classify the datasets. Most approaches have combined histograms with clustering techniques, such as affinity propagation [22], hierarchical clustering [23], and the iterative self-organizing data analysis technique [24].

2) Multidimensional transfer function design: Approaches to designing multidimensional TFs normally conform to two primary strategies. The first entails furnishing an interface that enables the manipulation of all data attributes. One example of this interface type is the parallel coordinate plot (PCP). The second strategy involves leveraging dimensionality reduction techniques, such as those based on Multidimensional Scaling (MDS) and Principal Component Analysis (PCA).

[25] employed PCP in their exploration scheme, integrating viewer parameters and TF specifications into the design interface. [21] uses the same strategy with a local liner embedding technique for dimensionality reduction. Similarly, [26] proposed a hybrid interface design comprising PCP and a scatter plot generated via MDS.

[20] utilized a self-organizing map (SOM) and a radial-based function for TF design. SOM conducts dimensionality reduction through unsupervised learning, resulting in a map where neighborhood regions represent similar voxels. The user interaction involves drawing widgets in these regions. Likewise, [27] utilized a spherical SOM, enabling user interaction with the map lattice. [8] constructed a volume exploration space with subtree structures, using hierarchical clustering in a modified dendrogram. Afterward, [4] revisited these methods, augmenting the SOM result with a normalization cut step. The exploration space transforms into a cell map, with each region representing volume information associated with meaningful volume structures. Our approach shares similarities with the work of [4], but it utilizes an MDS-based technique and a density clustering algorithm to automate material classification, thereby generating a modified scatter plot view.

[28] proposed one of the earliest strategies employing supervised learning, implementing neural networks and support vector machines. [29] combined SOM and backpropagation neural networks for material classification. [30] used a Generative Adversarial Network (GAN) framework to compute models, addressing both TF specification and viewer position. Later, [31] combined GAN with Convolutional Neural Networks (CNN) to synthesize the exploration process. [32] developed an approach that utilizes CNN for generating visualizations from TF colorization. More recently, [10] introduced a design galleries approach employing deep learning and differentiable rendering to assist users in exploring the design space.

[33] developed a graph-based approach to identify significant volume structures, involving clustering features, constructing a material graph topology, and enhancing important structure rendering.

III. Method

In this section, we describe an unsupervised method for TF design. Our method generates a semi-automated material classification and an initial TF specification. These elements are still combined to produce a simplified design interface and an intuitive volume exploration space.

Fig. 1 shows an overview of the proposed method. In a preprocessing step, the dataset is organized into a regular volume grid. Given that the input data is unlabeled, all techniques are applied from an unsupervised perspective.

A. Dimensionality Reduction

The first step of our method is a dimensionality reduction. It serves two primary purposes. Firstly, it structures a 2D interface for the TF design and the volume exploration space. Secondly, it prepares the data for the clustering step, which requires 2D input.

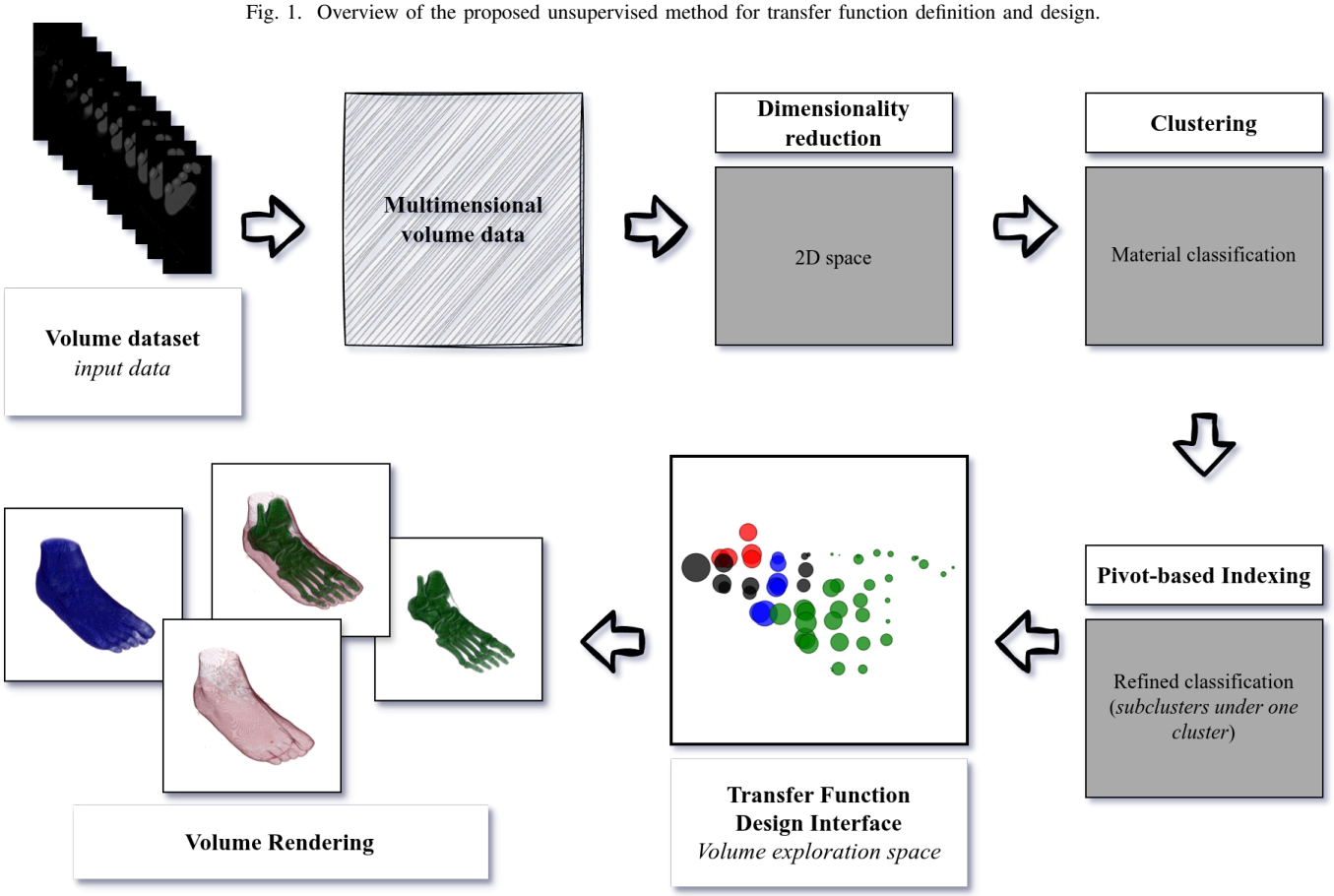
We employed a feature extraction technique in this step. Besides the dimensionality reduction, it minimizes information loss once the feature selection step removes irrelevant and redundant attributes.

FastMap [34], a classical MDS algorithm, is the feature extraction technique used. We take into account the three advantages of applying this technique: low time complexity cost even with large datasets [34], flexibility to handle high-dimensional datasets [34], and ability to preserve the clustering structure of the original data [35], [36]. Another advantage is that no input parameter is required. The time complexity of the FastMap algorithm is $\mathcal{O}(nk)$, where n is the total number of voxels and k, is the dimensionality of the target space.

Let d be the original dimensionality of input data, the algorithm projects n samples into a k-dimensional space, where $k <= d$. Here, n is the total number of voxels, d is the number of original attributes (dimensionality) and $k = 2$. FastMap is a recursive algorithm that can be succinctly described in the following steps:

1) Find the two points, named as pivots, furthest away from each other in a dataset.
2) Project the remaining points onto a hyperplane orthogonally positioned between the pivots.

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



Strategies capable of precisely identifying pivots have at least quadratic time complexity. To avoid compromising runtime, [34] developed a heuristic that is presented in Algorithm 1. It takes a set of points $\mathbb{O}$ and approximately finds the pair of points O_a and O_b that are the farthest from each other. In our approach, each point is a voxel, and thus, $\mathbb{O}$ is the set of all voxels. The algorithm considers only the values of selected attributes. A voxel's 3D position (x, y and z) is not used in any calculation.

Algorithm 1: Pivot searching of FastMap.

Input: $\mathbb{O}$
Output: Pivots O_a, O_b
1 $O_a \leftarrow$ random point $o \in \mathbb{O}$
2 $O_b \leftarrow$ point $o \in \mathbb{O}$ farthest from O_a
3 $O_a \leftarrow$ point $o \in \mathbb{O}$ farthest from O_b

B. Clustering

A major goal of our method is to simplify material classification and facilitate the highlighting of volume details. We address this objective by employing a classical density-based clustering algorithm, the DBSCAN [37].

DBSCAN is a widely utilized algorithm known for its success across various applications [38]. Nevertheless, its adoption in DVR comes with some caveats. The original version [37] exhibits a time complexity of $\mathcal{O}(n^2)$ in the worst case [38]. With practical usability in mind, we implemented a grid-based DBSCAN proposed by [39]. This version claims a time complexity of $\mathcal{O}(n \log(n))$.

Like the original algorithm [37], the 2D grid version also has $minPts$ and ε as input parameters. In this way, the user must fine-tune such parameters to best classify the volume data.

When this method step ends, each cluster comprises a subset of voxels that potentially represent a region of interest.

1) Grid-based DBSCAN of [39]: [39] introduces the concept of a grid to improve the efficiency of the clustering process, especially for high-dimensional datasets. The authors improve the scalability and efficiency of traditional DBSCAN by leveraging grid-based partitioning and density estimation techniques. Detailed explanations are provided in the works of [39] and [40]. The algorithm operates on a cell grid and comprises the tasks summarized next.

1) Grid partitioning. The first step involves partitioning the space into a grid of cells. Each cell represents a small portion of the entire space.

2) Density estimation. Within each cell, the algorithm calculates the density of points. The density is usually estimated using a distance threshold (ε) to determine the neighborhood of each point.
3) Identifying core points. Points with a density above a certain threshold ($minPts$) are considered core points. These core points are potential seeds for clusters.
4) Expanding clusters. Starting from a core point, the algorithm expands the cluster by iteratively adding neighboring points that also qualify as core points. The expansion continues until there are no more core points to be added.
5) Handling border points. Points that are within the ε neighborhood of a core point but do not meet the density requirement to be considered core themselves are classified as border points. Border points are assigned to the cluster of their nearest core point.
6) Handling noise. The points that are not core and do not belong to any cluster are considered noise points.

C. Pivot-based indexing

Our TF design interface revolves around a scatter plot view. Attempting to plot the entire dataset is impractical due to the high cognitive load and computational cost involved. To overcome this challenge, we implement a pivot-based indexing approach within each cluster identified by DBSCAN. Only the pivots within each cluster are plotted, thereby reducing visual density.

This step also serves as a second-level clustering process, refining each classified volume detail. Once the pivots are identified, each point in a cluster is assigned to the nearest pivot, as outlined in Algorithm 2. Therefore, a cluster is divided into sub-clusters represented by pivots.

Let $\mathbb{P}$ denote the set of all points within a cluster c, and $\mathbb{P}_s$ represent the selected pivots of the same cluster, every point $p \in \mathbb{P}$ is assigned to the sub-cluster of the nearest pivot $p_s \in \mathbb{P}_s$.

Algorithm 2: Finding sub-clusters within a cluster.

Input: Set of points $\mathbb{P}$ of a cluster c
Input: Set of pivots $\mathbb{P}_s$ of a cluster c
Output: Set of points $\mathbb{P}$ with associated sub-clusters

```
foreach p ∈ ℙ do
    p_s ← p nearest pivot in ℙ_s
    p assigned to p_s's sub-cluster
end
```

1) Sparse Spatial Selection: We use the Sparse Spatial Selection (SSS) for pivot-based indexing. This algorithm [41] is known for its straightforward implementation and low computational cost compared to other pivot-based techniques.

An overview of the SSS is shown in Algorithm 3. It identifies the points furthest from each other, termed pivots. Let $\mathbb{P}$ be a set of points, $alpha$ a distance factor within the interval $[0, 1]$, the first point $p_1 \in \mathbb{P}$ is added to the pivots $\mathbb{P}_s$. Subsequently, the algorithm traverses $\mathbb{P}$ to find additional pivots. With M representing the maximum distance between two arbitrary points, a point p is added to the pivots $\mathbb{P}_s$ only if $\forall p_s \in \mathbb{P}_s$, $\texttt{dist}(p, p_s) \geq M\alpha$, where $\texttt{dist}$ is a distance function and p_s is a pivot $\in \mathbb{P}_s$.

Algorithm 3: Sparse Spatial Selection.

Input: Set of points $\mathbb{P}$
Output: Set of selected pivots $\mathbb{P}_s$

```
ℙ_s ← p_1
foreach p ∈ ℙ do
    if ∀p_s ∈ ℙ_s, dist(p, p_s) ≥ Mα then
        ℙ_s ← ℙ_s ∪ {p}
    end
end
```

To control the number of pivots, users can adjust the distance factor α. A smaller α value favours selecting a greater number of pivots, while a value closer to 1 results in fewer pivots.

IV. Volume exploration space

Figure 2 illustrates the TF design interface. This system interface is a 2D scatter plot. Each circle represents a pivot selected by SSS. The positions provided in the FastMap results determine the points' position in the scatter plot. A pivot represents a central point within a cluster. It assigns size as a pivot's secondary property. The radius of a circle is calculated based on the number of voxels it represents, normalized using a logarithmic scale.

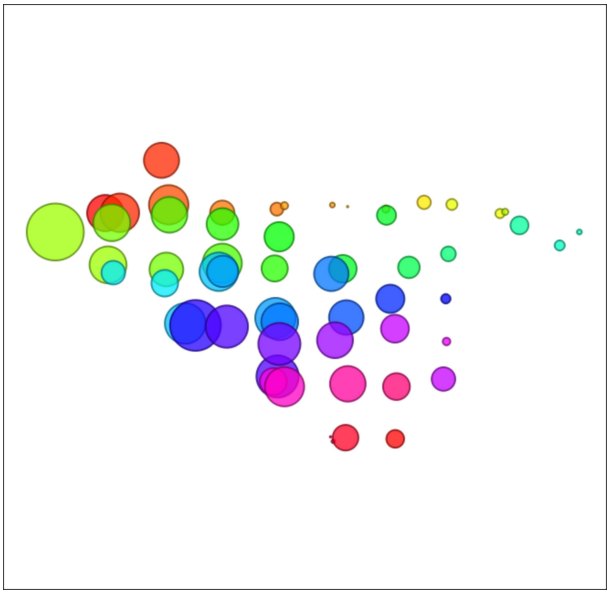
Our approach generates an initial specification for the TF using a predefined opacity value alongside a rainbow color scale. Each cluster is assigned a unique color.

Users can adjust the TF by following the What You See Is What You Get (WYSIWYG) principle. The lookup table is designed in such a way that the color and opacity of each pivot are directly mapped to the associated voxels, taking into account the clustering structure. Users can customize the color and opacity of any selected or not-selected elements.

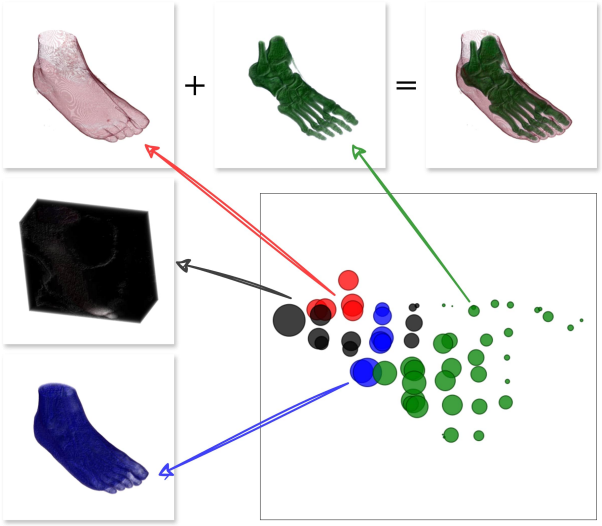
Volume exploration happens through the selection of pivots. Our system dynamically adjusts the opacity of selected pivots to enhance visibility while concurrently reducing the opacity of non-selected elements. Users can interact with the system by making arbitrary selections. Even though hierarchical exploration is not the primary focus, our system supports it by enabling selections to be saved as groups, which can also be treated as selectable entities. This functionality empowers users to select pivots, clusters, and groups.

The iterative selection of nearby elements is essential for identifying details and regions of interest inside a volume. FastMap and DBSCAN naturally group instances that are more similar closer together, simplifying the identification process.

Our approach automates material classification. Therefore, we assume that each cluster, or in a more sophisticated analysis, each pivot, represents a relevant item. If the user



(a) Initial transfer function specification (semi-automatic generated).



(b) Fine-tune material classification after user adjustment.

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

is unsatisfied with the result, they can go back to selecting or deselecting any element or adjust the method parameters, which are:

- multidimensional TF definition,
- DBSCAN ε and $minPts$, and
- SSS distance factor α.

V. Results

A. Experimental Design

We conducted all experiments on a computer equipped with an Intel Core i5-7200U processor, 8 GB RAM, running Ubuntu 22.04 64-bit, and an NVIDIA GeForce GT 940MX GPU.

For image rendering, we utilized a classical volume ray-casting algorithm with Blinn-Phong illumination and trilinear interpolation. The ray step was adjusted according to voxel spacing. Our runtime analysis reflects the average of five trials.

The system was implemented in C++, using the Qt Framework and CUDA C/C++ for parallel processing. The implementation is available in online repositories.

Table I presents the datasets used in our experiments. These datasets are widely recognized within the volume visualization community and are publicly available through online repositories.

All volume data consisted solely of material density, represented as scalar intensity values. Since multidimensional data was not directly available, derived attribute extraction was performed to generate a multidimensional input. We considered 13 attributes, namely: intensity, gradient magnitude, Laplacian magnitude, and 10 statistical measures computed from a local histogram. The statistical measures include absolute deviation, contrast, energy, entropy, inertia, kurtosis, mean, skewness, standard deviation, and variance.

TABLE I
Volume datasets.

Dataset	Grid size	Total of 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 the runtimes for the proposed method applied to each dataset.

TABLE II
Runtime (in seconds) of the proposed method applied to each volume dataset.

	Block engine	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
Transfer function design interface	1.48	1.86	0.79

C. Data classification

Since the same number of attributes is available, we consistently began the investigation with $k = 4$, considering $d = 13$ (where $k \approx \sqrt{13}$) available attributes.

The DBSCAN parameter $minPts$ is default set [37] as 4 in all experiments. With the data normalized, we vary ε between $[0.2, 0.35]$ and SSS parameter α between $[0.8, 0.95]$ to simulate volume exploration.

1) Engine block dataset: Fig. 3 shows the volume exploration space generated for the engine block dataset. Each numbered group is a classified volume detail rendered as presented in Fig. 4. The method parameters are set as follows: $k = 4$, TF = {intensity, skewness, gradient magnitude and variance}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.85$.

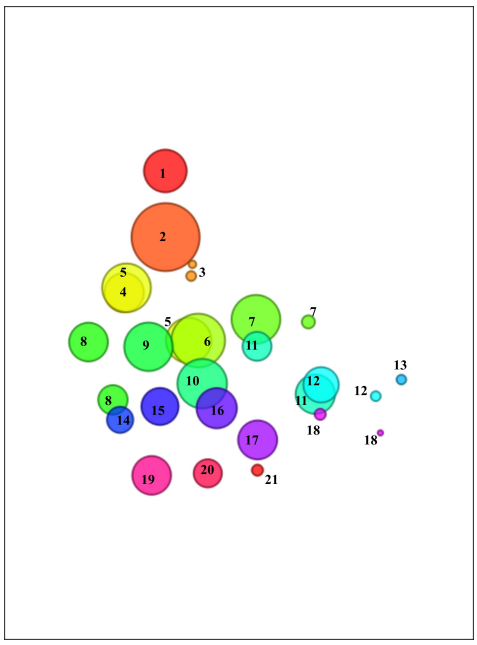


Fig. 3. Volume exploration space for the engine block dataset. Method parameters setup: transfer function = {intensity, skewness, gradient magnitude and variance}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.85$.

A volume exploration simulation is demonstrated in Fig. 5. It reveals different engine block components. The process happens from the initial setup presented in Fig. 3.

2) Knees dataset: A preliminary volume classification for knees datasets is presented in Fig. 6 and the related rendered details in Fig. 7. The method parameters are set as follows: TF = {intensity, variance, absolute deviation, energy and contrast}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.9$.

By exploring the volume details, it is possible to group and identify bones and muscular structures. Fig. 8 illustrates these structures, which include parts of the femur, tibia, patella, fibula, thigh muscles, and knee muscles.

3) Tooth dataset: Fig. 10 presents visualizations of a tooth dataset classification. The method parameters are set as follows: TF = {intensity, variance, absolute deviation, energy, contrast and entropy}; $minPts = 4$; $\varepsilon = 0.23$; and $\alpha = 0.9$.

Manually generated groups of related tooth details are presented in Fig. 11. It shows several discernible structures, including the enamel, pulp, dentin, crown, the entire tooth, and the fluid in which it is immersed.

VI. Discussion

The two-level dimensionality reduction strategy effectively addresses challenges inherent in TFs. The first level offers guidance for TF definition, departing from the conventional approach reliant solely on user domain knowledge. This departure represents a significant advancement in the field. The second level simplifies the design interface.

Fig. 4. Rendered volume classification details for the engine block dataset. Method parameters setup: transfer function = {intensity, skewness, gradient magnitude and variance}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.85$.

While the feature selection heuristic shows promise in all experiments, the task ultimately remains the user's responsibility, which is a major limitation of our work. Investigating other unsupervised feature selection stop criteria may yield proper results, but there is a lack of investigation of these approaches in the TF context, warranting separate consideration in future analyses.

The parameters of DBSCAN significantly influence data classification. The parameter $minPts$ can assume a default value, $minPts = 4$ [37] since FastMap projects the data in a 2D space. The parameter ε requires a fine-tune adjustment, but its behavior is stable. Higher values of ε lead to fewer but larger clusters, while lower values increase the number of smaller clusters.

Similarly, the adjustment of the SSS distance factor (α) follows the same behavior, with α being inversely proportional to the number of pivots within each cluster.

The practical implementation of the method is supported by minimal computational overhead, indicating favorable scalability for large datasets in size aspect. Despite the computational expense of the feature selection step in higher-dimensional cases, the remaining method steps can handle it.

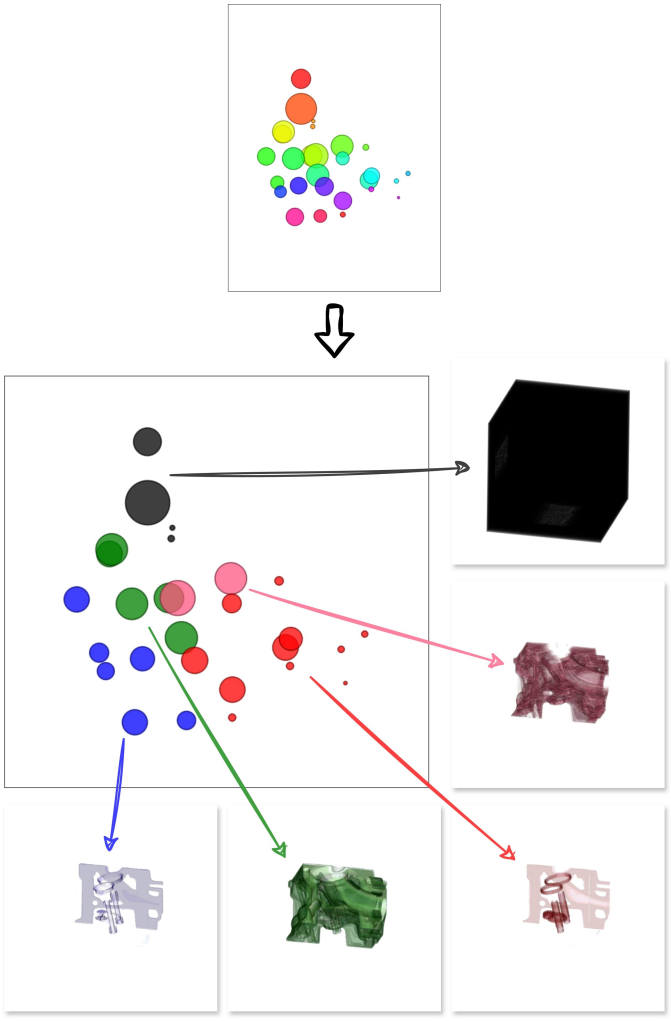


Fig. 5. Visual analysis of user-refined transfer function design and volume classification for engine block datasets. The volume details are manually grouped from an empirical perspective. Method parameters setup: transfer function = {intensity, skewness, gradient magnitude and variance}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.85$.

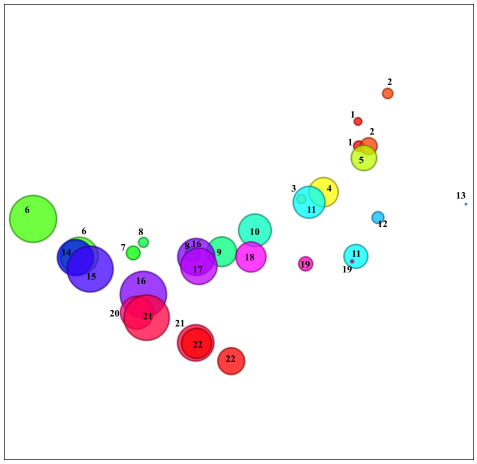


Fig. 6. Volume exploration space for the knees dataset. Method parameters setup: transfer function = {intensity, variance, absolute deviation, energy and contrast}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.9$.

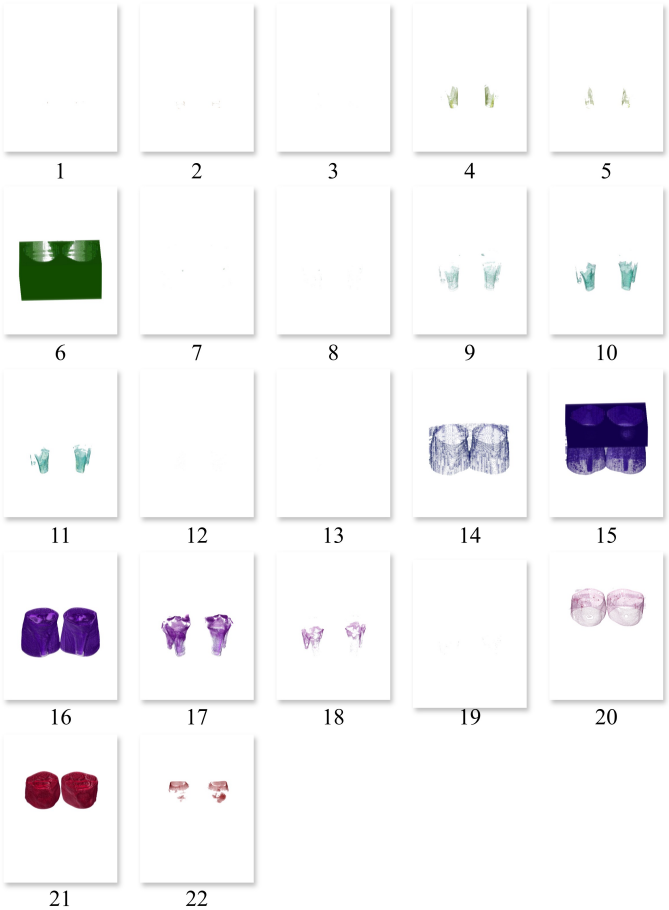


Fig. 7. Rendered volume classification details for the knees dataset. Method parameters setup: transfer function = {intensity, variance, absolute deviation, energy and contrast}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.9$.

VII. Conclusions

In this work, we presented a robust method for TF definition and design from an unsupervised data perspective. Our comprehensive approach covers the entire classification process, from feature selection to establishing a data domain, developing a TF design interface, and creating a simplified volume exploration space. We proposed a heuristic for feature selection based on similarity rankings of attributes, employed FastMap for efficient feature extraction, utilized DBSCAN for effective clustering, and leveraged SSS for pivot-based indexing. These techniques collectively facilitate semi-automatic classification and initial TF specification, forming the basis of our TF design interface and exploration system, as demonstrated through a scatter plot view.

The method exhibits minimal computational overhead, brief runtime, and low storage requirements, highlighting its practicality and scalability for real-world applications. However, the current feature selection process, heavily reliant on user input, presents a limitation due to potential variability introduced by differing levels of user expertise. Nevertheless, the

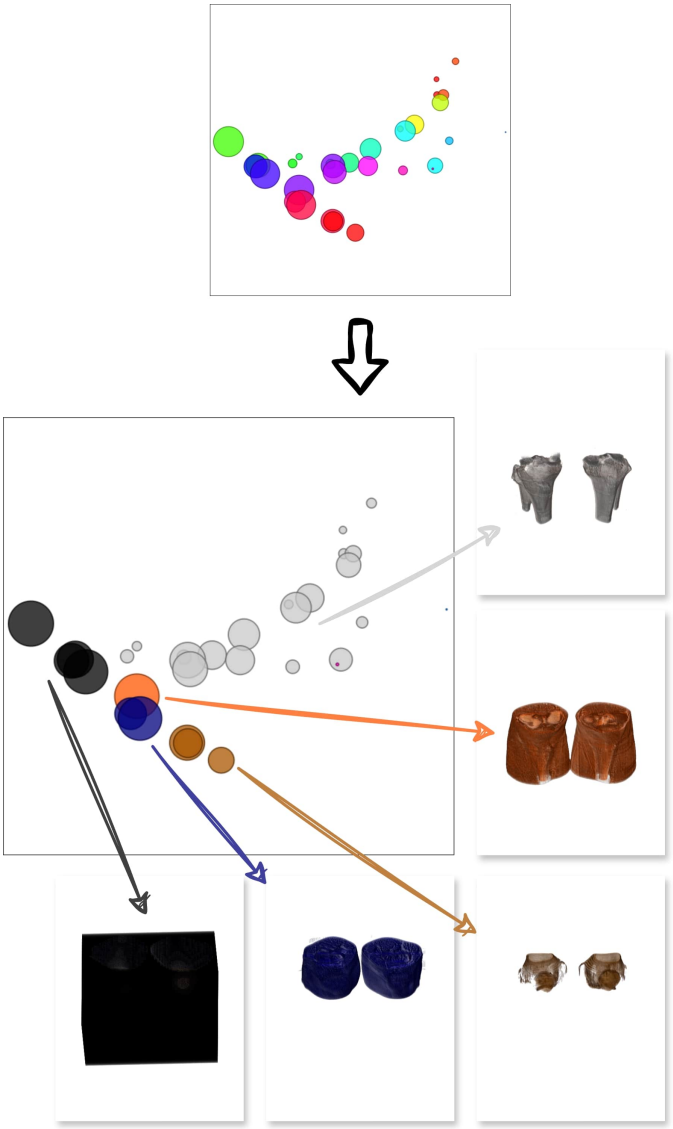


Fig. 8. Visual analysis of user-refined transfer function design and volume classification for knees dataset. The volume details are manually grouped from an empirical perspective. Method parameters setup: transfer function = {intensity, variance, absolute deviation, energy and contrast}; $minPts = 4$; $\varepsilon = 0.35$; and $\alpha = 0.9$.

results are satisfactory in all experiments, given the unlabeled nature of the data. To address this, future work will focus on investigating advanced feature selection techniques and other stop criteria.

Another point of investigation is subjecting the method to handling large and high-dimensional datasets. The techniques included in the method are capable of operating well on datasets with these characteristics, and further practical investigation is needed in future studies to confirm its effectiveness.

Furthermore, we recognize the necessity to evaluate our proposed method using multivariate data, aiming to expand its applicability and robustness across diverse datasets.

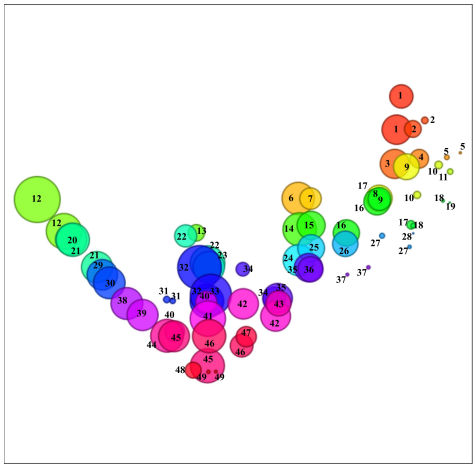


Fig. 9. Volume exploration space for the tooth dataset. Method parameters setup: transfer function = {intensity, variance, absolute deviation, energy, contrast and entropy}; $minPts = 4$; $\varepsilon = 0.23$; and $\alpha = 0.9$.

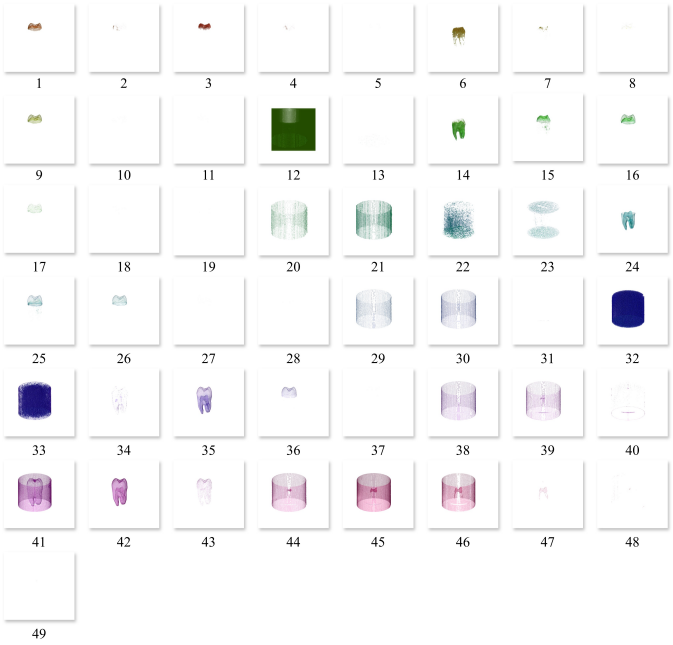


Fig. 10. Rendered volume classification details for the tooth dataset. The method parameters are set as follows: transfer function = {intensity, variance, absolute deviation, energy, contrast and entropy}; $minPts = 4$; $\varepsilon = 0.23$; and $\alpha = 0.9$.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT 3.5 in order to improve readability and language of this paper. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

VIII. Introduction

SIBGRAPI 2025 [**?**].

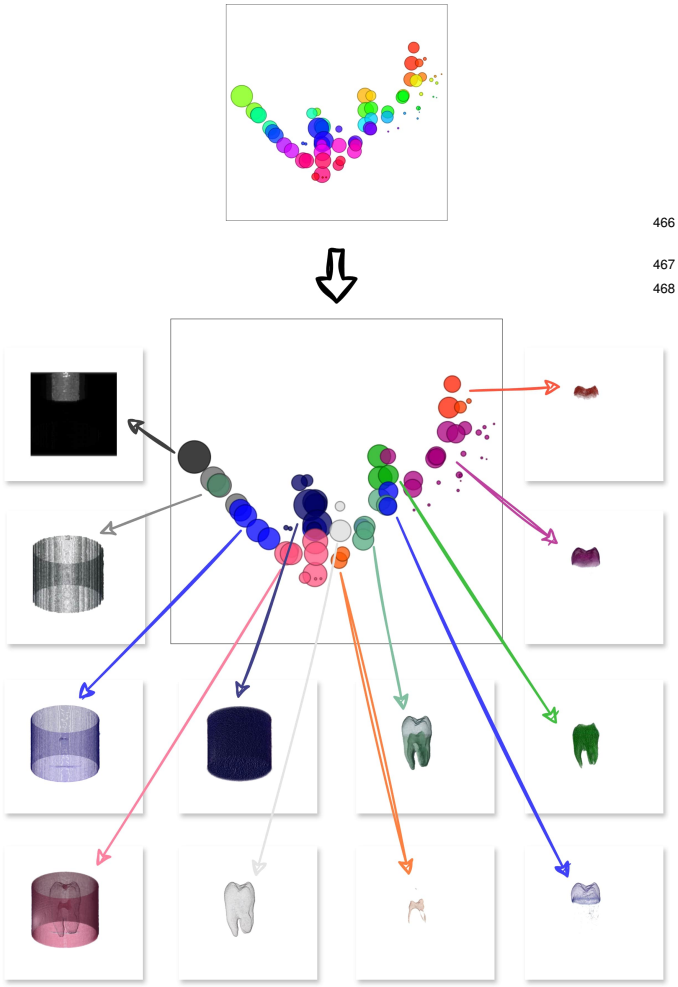


Fig. 11. Visual analysis of user-refined transfer function design and volume classification for tooth dataset. The volume details are manually grouped from an empirical perspective. The method parameters are set as follows: transfer function domain = {intensity, variance, absolute deviation, energy, contrast and entropy}; $minPts = 4$; $\varepsilon = 0.23$; and $\alpha = 0.9$.

Fig. 12. SIBGRAPI - Conference on Graphics, Patterns and Images.

A. *Subsection Heading Here*

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

IX. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

TABLE III
An Example of a Table

One	Two
Three	Four

References

[1] T. T. Elvins, "A survey of algorithms for volume visualization," *ACM Siggraph Computer Graphics*, vol. 26, no. 3, pp. 194–201, 1992.

[2] C. Xu, G. Sun, and R. Liang, "A survey of volume visualization techniques for feature enhancement," *Visual Informatics*, vol. 5, no. 3, pp. 70–81, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468502X21000358

[3] 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.

[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] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. Avila, K. Raghu, R. Machiraju, and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 16–22, 2001.

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

[8] 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.

[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] 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.

[11] J. Hladuvka, A. König, and E. Gröller, "Curvature-based transfer functions for direct volume rendering," in *Spring Conference on Computer Graphics*, vol. 16, no. 5. Citeseer, 2000, pp. 58–65.

[12] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," in *IEEE Visualization, 2003. VIS 2003.* IEEE, 2003, pp. 513–520.

[13] C. Correa and K.-L. Ma, "Size-based transfer functions: A new volume exploration technique," *IEEE transactions on visualization and computer graphics*, vol. 14, no. 6, pp. 1380–1387, 2008.

[14] S. Wesarg and M. Kirschner, "Structure size enhanced histogram," in *Bildverarbeitung für die Medizin 2009*. Springer, 2009, pp. 16–20.

[15] A. Tappenbeck, B. Preim, and V. Dicken, "Distance-based transfer function design: Specification methods and applications." in *SimVis*, 2006, pp. 259–274.

[16] J. J. Caban and P. Rheingans, "Texture-based transfer functions for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1364–1371, 2008.

[17] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Gröller, "Volume visualization based on statistical transfer-function spaces," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2010, pp. 17–24.

[18] 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.

[19] 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

(a) Case I

(b) Case II

Fig. 13. SIBGRAPI - Conference on Graphics, Patterns and Images.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] I. K. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Lab., CA (US), Tech. Rep., 2002.

[36] I. Khan, J. Z. Huang, N. T. Tung, and G. Williams, "Ensemble clustering of high dimensional data with fastmap projection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 483–493.

[37] 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.

[38] 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.

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

[40] 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.

[41] 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.