

Article

Similarity Measurement and Retrieval of Three-Dimensional Voxel Model Based on Symbolic Operator

Zhenwen He ^{*}, Xianzhen Liu and Chunfeng Zhang

School of Computer Science, China University of Geosciences (Wuhan), 388 Lumo Road, Wuhan 430074, China; xzliu@cug.edu.cn (X.L.); 1202211228@cug.edu.cn (C.Z.)

* Correspondence: zwhe@cug.edu.cn

Abstract: Three-dimensional voxel models are widely applied in various fields such as 3D imaging, industrial design, and medical imaging. The advancement of 3D modeling techniques and measurement devices has made the generation of three-dimensional models more convenient. The exponential increase in the number of 3D models presents a significant challenge for model retrieval. Currently, these models are numerous and typically represented as point clouds or meshes, resulting in sparse data and high feature dimensions within the retrieval database. Traditional methods for 3D model retrieval suffer from high computational complexity and slow retrieval speeds. To address this issue, this paper combines spatial-filling curves with octree structures and proposes a novel approach for representing three-dimensional voxel model sequence data features, along with a similarity measurement method based on symbolic operators. This approach enables efficient similarity calculations and rapid dimensionality reduction for the three-dimensional model database, facilitating efficient similarity calculations and expedited retrieval.

Keywords: 3D voxel model; feature representation; similarity measures; model retrieval



Citation: He, Z.; Liu, X.; Zhang, C. Similarity Measurement and Retrieval of Three-Dimensional Voxel Model Based on Symbolic Operator. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 89. <https://doi.org/10.3390/ijgi13030089>

Academic Editors: Sisi Zlatanova and Wolfgang Kainz

Received: 3 January 2024

Revised: 5 March 2024

Accepted: 6 March 2024

Published: 11 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advancement of three-dimensional modeling technology and measurement devices, the generation and acquisition of three-dimensional models have become more convenient [1]. In various fields such as healthcare, architecture, natural sciences, and gaming [1–4], three-dimensional models have extensive application prospects, enriching people's production and daily lives. However, the widespread generation and usage of three-dimensional models have led to a significant increase in their quantity, which has created a higher demand for three-dimensional model retrieval technology. How to effectively extract features from three-dimensional models and measure their similarity, enabling the rapid retrieval of suitable three-dimensional models from a large-scale model database, is a pressing issue in the development of three-dimensional model research.

Feature extraction is a fundamental and critical task in three-dimensional model retrieval [5]. It involves extracting specific attributes and information from models, such as the shape, surface texture, geometric structure, and topological structure [6], to obtain feature descriptors that enable the calculation of similarity between models, facilitating model classification and retrieval. Traditional three-dimensional model retrieval methods face challenges due to the high dimensionality of models [7], resulting in high computational complexity. However, representing three-dimensional models as sequential data [8] can achieve rapid dimensionality reduction and improve the efficiency of similarity calculations.

Therefore, based on the features of three-dimensional models, this paper focuses on researching voxelization methods based on space-filling curves [9], which preserve the aggregation features of models, and utilize an octree structure [10] to store voxel data. Through voxelization, intricate geospatial data can be transformed into structured and manageable data formats, greatly facilitating spatial analysis, retrieval, and visualization.

This approach enables researchers to simulate and analyze the three-dimensional shape, spatial distribution, and attribute characteristics of geological bodies more precisely, thus providing powerful support for geological exploration, resource assessment, disaster warning, and other applications [11–15]. To address the high dimensionality issue in representing three-dimensional voxel models, this paper proposes a feature extraction and sequential representation method based on symbolic operators, mapping three-dimensional model data into one-dimensional sequences. The measurement between models is transformed into the measurement between symbol sequences, enabling similarity calculation based on symbolic operators for three-dimensional voxel models and improving the efficiency of model retrieval. The feasibility of the proposed algorithm is validated through classification and retrieval experiments conducted on the open-source datasets ModelNet10 and ModelNet40. The experimental results demonstrate that the proposed feature extraction method for three-dimensional voxel models effectively preserves the shape characteristics of models. Furthermore, the feature representation method and similarity measurement based on symbolic operators achieve good performance in three-dimensional model classification experiments. The main contributions of this paper are the following:

- Based on space-filling curves and octree structures, this study focuses on voxelization and feature extraction methods for three-dimensional models. A feature extraction method for three-dimensional voxel models based on the Hilbert curve is proposed, enabling the mapping of three-dimensional models to voxel models and further to sequential data. This approach achieves feature extraction and sequential representation for three-dimensional voxel models.
- We propose a method for representing the features of three-dimensional voxel models based on symbolic operators and their similarity measurement. Symbolic operators are mappings from a function space to a symbolic space. To mitigate the curse of dimensionality, the sequential data obtained from the three-dimensional voxel model are mapped to a hexadecimal symbolic space, yielding the feature description of the three-dimensional model (VSO, representation of 3D voxel model based on symbolic operators). Based on this representation, a similarity measurement method is proposed.

The remainder of this paper is organized as follows: the next section is the related work. Section 3 provides a detailed description of the Symbol Sequence Feature Representation (VSO) for 3D models and its similarity measurement methods. Section 4 presents the experiments and analysis of the results. Finally, in Section 6, we present our conclusion.

2. Related Work

The current 3D model retrieval methods can be divided into text-based [15] and semantic-based [16] retrieval and 3D model structure content retrieval [17]. The method of using text keywords to complete 3D model retrieval relies too much on the manual annotation of 3D models, which is highly subjective and cannot adapt to the rapid growth in the number of 3D models, so more research focuses on content-based 3D model retrieval [18]. The key technology of the content-based 3D model retrieval method lies in the feature extraction [19] of the model. Existing content-based feature extraction methods for 3D models can be divided into two main categories: manual design-based and learning-based, as shown in Table 1.

Manual design involves designing feature extraction algorithms based on various attributes of 3D models such as statistical data [20], geometric shape [21], topological structure [22], and local features. Statistical data-based retrieval methods involve statistical analysis and calculations to match and retrieve 3D models. They are computationally simple and have strong adaptability to models with a higher level of noise. However, they provide relatively coarse descriptions of models, resulting in a generally lower similarity strength between models. Feature extraction based on geometric shape allows for the more comprehensive extraction of high-level information from the models. [23] However, most methods require complex computations and slow conversion speeds due to the need for model transformation. Additionally, they require a significant amount of storage space.

Topological structure-based retrieval algorithms can differentiate between the primary and secondary structures of 3D objects [24]. However, before feature extraction, a series of model normalization processes must be performed, which consume substantial computational resources and are sensitive to noise. Local feature-based retrieval methods have lower feature dimensions and computational requirements [25]. However, they are sensitive to occlusion and missing data.

Learning-based methods, on the other hand, utilize data-driven deep-learning techniques to automatically extract features from 3D models using end-to-end models [26]. Deep-learning-based 3D model retrieval methods can automatically learn more representative high-level semantic features through training on large-scale data [27]. However, this approach requires a large amount of annotated data for training and relies on manual annotation or limited existing datasets. Updating the database and retraining the models can be time-consuming and costly.

Table 1. Content-based 3D model retrieval methods.

Feature Extraction	Algorithm	
Manual design-based	Based on statistical data	GD [28] D2 [29] LFD [19]
	Based on geometric shape	SPH [30] ICP [31]
	Based on topological structure	Reeb [32] FPFH [33]
	Based on local feature	SHOT [34]
	Based on voxelization	VoxelNet [35]
Learning-based	Based on multiple viewpoints	3D ShapeNet [36] MVCNN [37]
	Based on raw point cloud	PointNet [38]

Three-dimensional models have the characteristic of high dimensionality. Extracting features from the models can yield descriptors for three-dimensional models. Sequence data representation enables the rapid dimensionality reduction of three-dimensional models, enhancing efficiency in similarity computation. Therefore, sequence data representation and similarity measurement have also become research topics in three-dimensional model retrieval.

Currently, commonly used sequence representation methods can be divided into three categories: non-data-adaptive methods, data-adaptive methods, and model-based methods [39], as shown in Table 2. Non-data-adaptive methods use the same transformation parameters for each sequence during sequence transformation from high-dimensional to low-dimensional space. They are often used to represent equi-length sequences or segmental sequence data with the same transformation parameters. For example, the classical Discrete Fourier Transform (DFT) can transform sequences into the frequency domain [40] to address the curse of dimensionality. The Discrete Wavelet Transform (DWT) simultaneously represents both the time- and frequency-domain information of sequences [41]. Piecewise Aggregate Approximation (PAA) utilizes segmental average values [42] to represent sequence data, with the dimensionality reduction ratio determined by the number of segments. PAA exhibits better performance in terms of indexing speed and flexibility. Data-adaptive methods allow transformation parameters to vary with sequences but are influenced by both individual sequences and the entire dataset. For instance, Symbolic Aggregate Approximation (SAX) [43] performs dimensionality reduction and standardization on data information based on the segmental aggregate approximation method. This method not only effectively eliminates noise interference and retains important feature information in sequences but also achieves high compression rates and significant dimensionality reduction effects on data. Singular Value Decomposition (SVD) is mainly used for decomposing non-square matrices [44], driven by principal component analysis to transform high-dimensional sequence data into low-dimensional data through numer-

ical decomposition methods. Model-based methods assume that sequence data can be generated by a certain model and appropriate parameters are set to represent sequences accordingly. For example, the Hidden Markov Model (HMM) defines relationships between sequence variables [45], and the Markov Chain Model (MCs) is a stochastic process model that can discretize sequences [46].

Table 2. The primary representation method for sequential data.

Type	Feature Representation Methods	Time Complexity
Non-data-adaptive methods	DFT	$O(n \log(n))$
	DWT	$O(n)$
	PAA	Fastest $O(n)$
Data-adaptive methods	SAX	$O(n)$
	SVD	$O(Mn^2)$
Model-based methods	HMM	
	MCs	

Similarity measurement methods involve measuring the distance between two different sequences to determine whether they are similar to each other [47]. Currently, research on similarity measurement methods mostly builds upon existing approaches, proposing improvements for applications in data mining. The Euclidean distance maps two original sequences into points in an n -dimensional vector space, then calculates the distances between point values in a pairwise manner [48]. Its advantages lie in its simplicity and ease of implementation. However, due to its point-to-point calculation nature, the Euclidean distance requires the compared sequences to be of equal length and is more sensitive to deformation and noise. The Manhattan distance is derived from the distance between city blocks, where the values of corresponding feature vectors in each dimension are subtracted to obtain the accumulated differences as the sum of absolute differences between the two points' axes. This method is relatively simple but exhibits instability. The Mahalanobis distance is a modification of the Euclidean distance that corrects issues of inconsistent and correlated scales in each dimension, making it suitable for addressing non-independently distributed dimensions in high-dimensional linearly distributed data. Dynamic Time Warping (DTW) effectively addresses the limitation of the Euclidean distance in only being able to compute distances for equi-length sequences [49]. This method allows for warping, stretching, and shrinking through point-to-point calculations. It can compute the similarity of sequences with different lengths reasonably. The Longest Common Subsequence (LCSS) distance measurement method exhibits strong noise-handling capabilities, but further improvement is needed to address stretching and amplitude shifting issues, and it does not support the triangle inequality [50].

3. Architecture of 3D Model Retrieval Based on VSO

The proposed 3D model retrieval method in this paper, as shown in Figure 1, first combines Hilbert space-filling curves [51] and octree to voxelize the original 3D model, obtaining a 3D voxel model. Then, the voxel model is subjected to feature extraction and dimensionality reduction using a sequential representation method. Based on symbolic operators, the 3D voxel model is transformed into a series of symbol sequences, resulting in a symbolic sequence feature descriptor for the 3D model. Finally, based on the symbolic sequence representation method, a corresponding similarity distance measurement is proposed to calculate the similarity between different 3D models and achieve the fast retrieval of 3D model data. This section will be divided into three parts to provide detailed explanations of the three key steps of the method proposed in this paper: voxelization of models based on the Hilbert curve, 3D voxel model feature representation based on symbolic operators, and the similarity measurement method based on VSO.

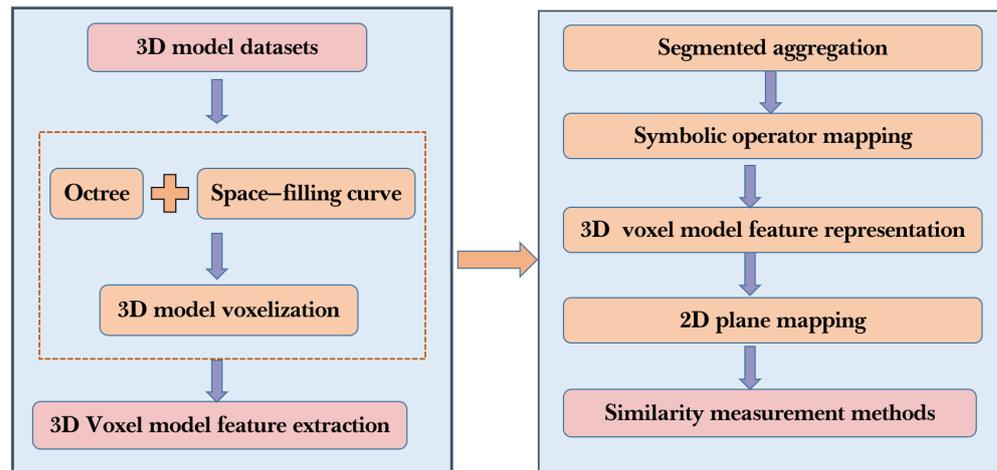


Figure 1. Three-dimensional model retrieval based on VSO architecture.

3.1. Voxelization of Models Based on the Hilbert Curve

Voxelization of a 3D model is the process of converting polygon mesh data into a discrete voxel representation that closely approximates the original model while preserving its accuracy. The accuracy of voxel representation depends on the resolution of the voxels, where a higher resolution results in larger storage space and a higher level of detail preserved from the original model. By traversing the file data of a 3D model, the maximum and minimum values of the vertex coordinates are obtained, and these values are, respectively, denoted as x_{max} , y_{max} , z_{max} and x_{min} , y_{min} , z_{min} , determining two vertices $(x_{max}, y_{max}, z_{max})$ and $(x_{min}, y_{min}, z_{min})$. By taking these two vertices as diagonal points, a three-dimensional cubic space can be determined. This cubic space represents the minimum bounding box of a three-dimensional model. The three-dimensional model is divided into voxel models of $n \times n \times n$ using an AABB bounding box. Here, “ n ” represents the number of voxels in a certain direction of voxel space during the voxelization process, and it can also be referred to as voxelization granularity. If the size of the minimum bounding box is denoted as $L_x \times L_y \times L_z$, and assuming the smallest voxel unit size is $l_x \times l_y \times l_z$, the calculation method for the geometric information of each voxel is as follows:

$$l_x = \left\lfloor \frac{x_{max} - x_{min}}{n} \right\rfloor \quad (1)$$

$$l_y = \left\lfloor \frac{y_{max} - y_{min}}{n} \right\rfloor \quad (2)$$

$$l_z = \left\lfloor \frac{z_{max} - z_{min}}{n} \right\rfloor \quad (3)$$

The values of the voxel units in the voxel space formed by the AABB bounding box can be calculated using the following method:

$$x_v = \text{ceil} \left(\frac{x - x_{min}}{l_x} \right) \quad (4)$$

$$y_v = \text{ceil} \left(\frac{y - y_{min}}{l_y} \right) \quad (5)$$

$$z_v = \text{ceil} \left(\frac{z - z_{min}}{l_z} \right) \quad (6)$$

For each voxel, the voxel information can be represented using a single bit of data based on the spatial relationship between each voxel unit and the 3D model. The spatial

relationship refers to the intersection or separation between the voxel unit and the 3D model. If the voxel unit intersects with the surface of the 3D model, it is represented as 1; otherwise, it is represented as 0. After this operation, a complete voxelized model of the 3D model can be obtained. Considering the spatial locality, continuity, and effectiveness of the 3D voxel model, and making the model easy to handle and analyze, we chose to map the 3D voxel model to one-dimensional data space using spatial-filling curves. Spatial-filling curves are one-dimensional curves that can “fill” d-dimensional space. The most common and widely used spatial-filling curve mapping methods include Hilbert curves [51], Z curves [52], Gray curves [53], etc. Among them, the Hilbert spatial-filling curve performs the best in data clustering.

The Hilbert curve is a space-filling curve that can fill an entire square plane. It was proposed by mathematician David Hilbert in 1891. Taking the first-order Hilbert space-filling curve as an example, a square space is evenly divided into four subspaces. The center points of these subspaces are denoted as (R1, R2, R3, R4), and then connecting these center points forms the grid curve. In practical applications, it is often necessary to construct an i -th-order Hilbert curve. During construction, the i -th-order Hilbert curve can be divided into grids within the grid, replacing the $(i - 1)$ -th-order curve. Additionally, to ensure the continuity of the curve, a flipping operation is performed on the curve within the grid. The flipping operation is carried out as follows: Firstly, for the curve in R1, rotate it 180° around the line connecting the bottom-left and top-right points of the grid. Then, connect it with the curve in R2, and connect R2 with R3. Next, rotate them 180° around the line connecting the top-left and bottom-right points of the grid, and connect them with the curve in R3. This process yields the i -th-order Hilbert space-filling curve. Figure 2 depicts the first-, second-, third-, and fourth-order Hilbert space-filling curves, respectively.

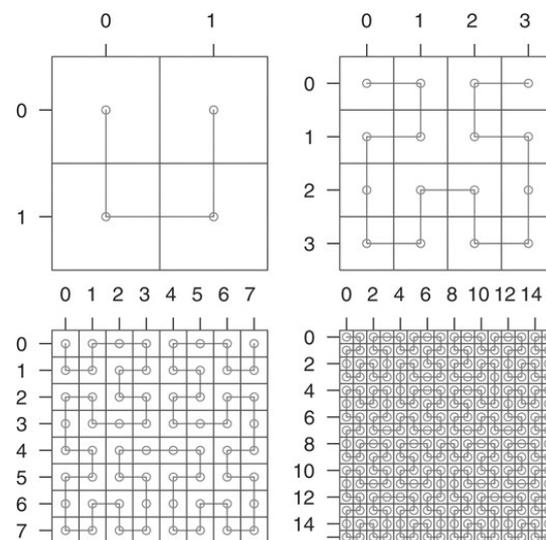


Figure 2. The Hilbert curve.

Similar to the construction of Hilbert space-filling curves in two-dimensional space, it is also possible to construct Hilbert space-filling curves in three-dimensional space. Figure 3 shows a simple three-dimensional Hilbert space-filling curve. If we assume that a three-dimensional Hilbert curve can fill a space R^3 with dimensions $2^n \times 2^n \times 2^n$, then we refer to this curve as an n -th-order three-dimensional curve.

By constructing the Hilbert curve to fill the 3D model space, there is only one traversal through the discrete voxel units of the 3D model. The Hilbert encoding avoids large-scale jumps, resulting in the good clustering performance of the Hilbert space arrangement. Adjacent points on the Hilbert curve are guaranteed to be adjacent in the original space. After scanning the 3D voxel model in the order of the 3D Hilbert curve, an octree data structure is selected to store the voxel data of the 3D model. Fine-grained voxel data are

stored for local models with rich detail, while coarse-grained voxel data are stored for parts of the model with less detail. This approach not only saves the storage space required during the 3D model storage process but also preserves the detailed features of the 3D model to a great extent. During the octree node partitioning process, different levels of precision for voxel models are achieved by adjusting the data threshold in the nodes and the tree depth. Non-empty nodes that do not reach the threshold are further split, increasing the number of approximation operations of the space-filling curve. For an n -order Hilbert curve, as n approaches infinity, the Hilbert curve approximately fills the entire space.

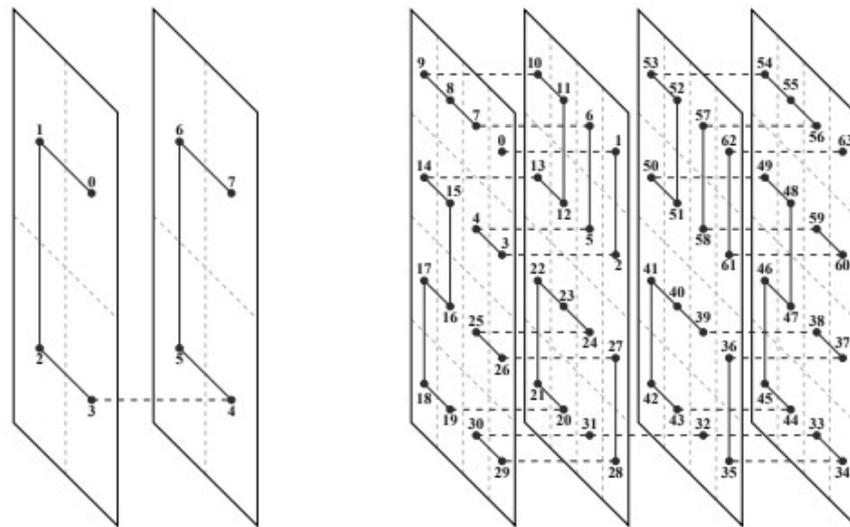


Figure 3. The three-dimensional Hilbert curve.

Figure 4 shows the flowchart of voxelization and octree storage based on Hilbert space curve filling. For all three-dimensional model data, voxelization is achieved using the Hilbert curve method to obtain three-dimensional voxel models, and an octree structure is employed to flexibly store model structural data, saving storage space.

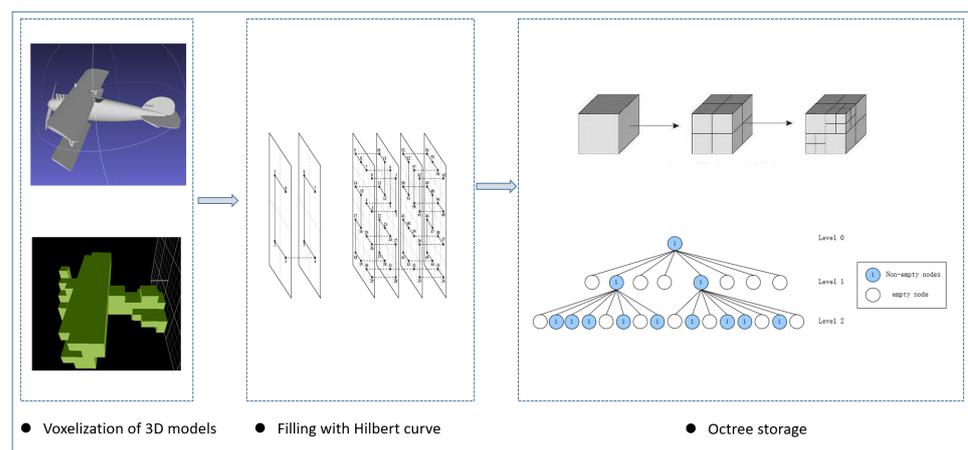


Figure 4. Hilbert space-filling curve-based voxelization and octree storage of 3D models.

3.2. Three-Dimensional Voxel Model Feature Representation Based on Symbolic Operators

After applying the voxelization method represented in Figure 4, a three-dimensional voxel model can be obtained. Storing the voxel model using an octree structure also helps to save storage space. However, when facing a large-scale 3D model retrieval, frequent data reading imposes significant pressure on the similarity calculation. Therefore, by traversing the octree, the 3D voxel model data are mapped to a binary encoding sequence, achieving the initial feature mapping of the 3D model. Figure 5a represents the binary sequence

representation of a voxel granularity of $4 \times 4 \times 4$ under the second-order Hilbert curve, while Figure 5b represents the same model with a voxel granularity of $8 \times 8 \times 8$ under the third-order Hilbert curve.

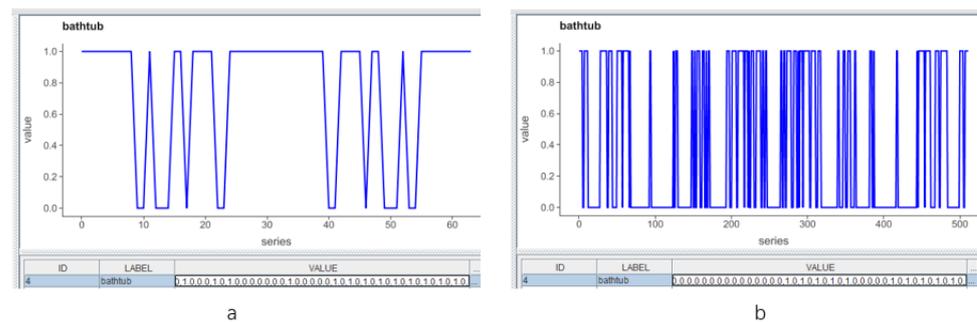


Figure 5. (a) Voxel granularity $4 \times 4 \times 4$, (b) voxel granularity $8 \times 8 \times 8$.

Although the binary sequence can achieve a certain level of representation for the 3D model, as the data volume increases and higher voxel granularity is required, the binary sequence still occupies a large amount of storage space and cannot effectively represent the feature differences between different 3D voxel models, leading to an accurate similarity measurement. Therefore, it is necessary to use the binary sequence representation of the voxel model as a basis for further dimensionality reduction and similarity measurement.

Let $X = (x_1, x_2, \dots, x_n)$ be the original sequence, D be the operator applied to X , and the sequence obtained by applying operator D to X be denoted as

$$XD = (x_{1d}, x_{2d}, \dots, x_{nd}) \quad (7)$$

D is referred to as a sequence operator. If the sequences formed by XD consist solely of symbols, then D is called a symbolic operator. D is a mapping from the function space to the symbol space, through which the symbol sequence XD is obtained. By using sequence operators, the feature mapping from 3D models to sequence data is achieved. As the order of the Hilbert curve increases, a higher-precision representation of 3D voxel models is obtained. However, this approach has high computational complexity and requires a large amount of storage space. Therefore, this paper proposes a method for representing 3D voxel models based on hexadecimal symbolic operators, called VSO (representation of 3D voxel model based on symbolic operators). Since every four adjacent voxels can form sixteen different filling states, their planar representation is shown in Figure 6a. Each state can be mapped to a hexadecimal symbol. In order to better measure the similarity of 3D voxel models and perform 3D model retrieval, the high-dimensional serialized data obtained are mapped through symbolic operators to obtain a specified length of symbol sequence. By calculating the distance between the sequence data, the complex similarity calculation between 3D voxel models is transformed into a similarity calculation between the sequence data. Based on the hexadecimal representation method for 3D voxel models, a hexadecimal symbol sequence of the 3D model can be obtained. Considering that a model should have at least eight voxels to be distinguishable, the sequence data are divided into model blocks, with every two characters representing one model block, as shown in Figure 6b.

Dimensionality reduction can be further applied to the obtained symbol sequences from the hexadecimal representation. Since any nonzero value in a state implies a nonzero symbol representation, the symbol sequence can be transformed from hexadecimal to binary representation by grouping every four bits into a unit and setting the nonzero parts to 1 while leaving the zero parts as 0. This process yields a binary representation derived from the hexadecimal representation. Repeating this procedure allows for obtaining shorter sequences that capture global features. At the same time, the representation of the 3D voxel model becomes coarser, achieving a lower granularity in the voxel model representation.

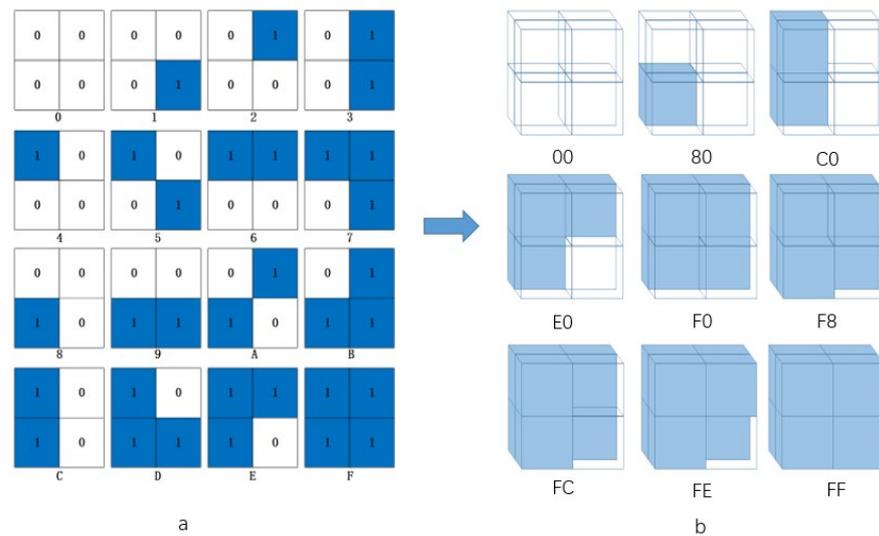


Figure 6. (a) Sixteen filling states of voxels (b) Representation of 3D models based on symbolic operators.

Figure 7 depicts the symbol sequences generated using two different VSO methods for three different models from distinct categories. Since the models have different categories and significant shape variations, the resulting symbol sequences are entirely different. The symbol sequences provide detailed shape descriptions, and it is evident that the three sequences have low similarity, indicating that they do not belong to the same label category. Figure 8 represents the symbol sequences generated from three different models within the same category. Although the details still differ slightly, the sequences exhibit minor differences and have similar trends and variations, suggesting that they belong to the same label category.

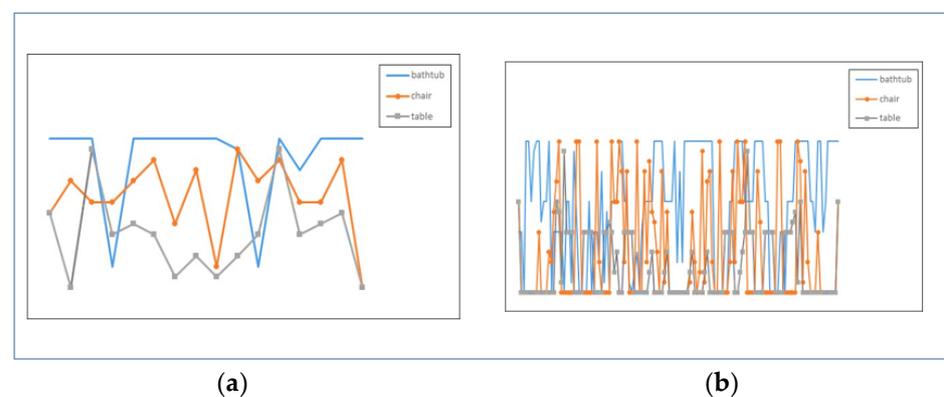


Figure 7. (a) Symbol sequence of length 16, (b) symbol sequence of length 128.

Using the VSO for the feature representation of three-dimensional models enables the expression of 3D models as symbol sequences. That is, representing a three-dimensional model as a sequence composed of symbols, and being able to distinguish different models through sequence features, allows similar three-dimensional models to have similar symbol sequences, while different three-dimensional model symbol sequences are also different. When traversing voxel models using Hilbert curves of different orders, the three-dimensional space can be partitioned more accurately, resulting in voxel models with different granularities.

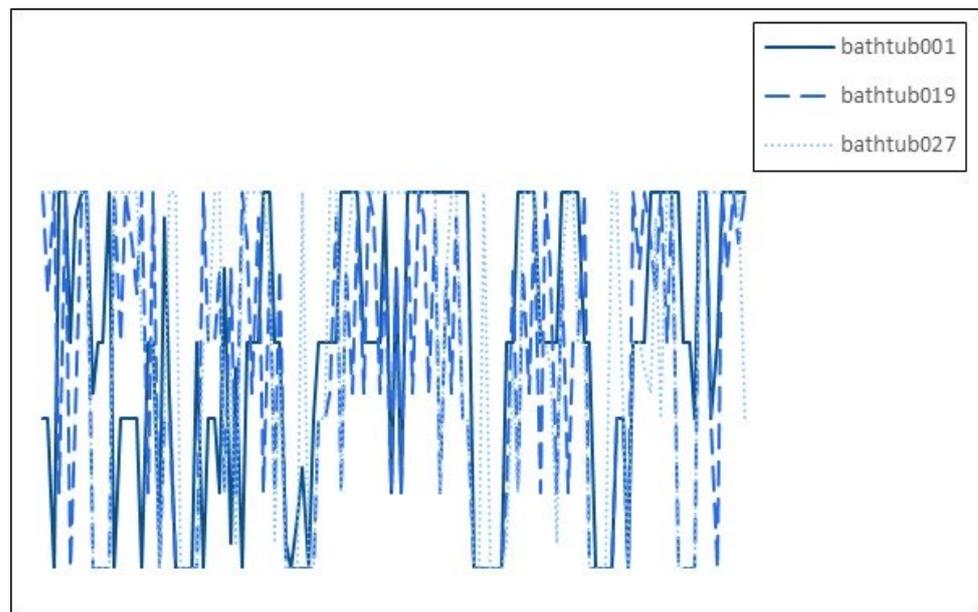


Figure 8. Symbol sequence for the bathtub class model.

3.3. Similarity Measurement Method Based on VSO

As shown in Figure 9, the voxelization of 3D models and their representation using the proposed symbol operator-based method allow for the transformation of 3D models into a series of one-dimensional symbol sequences. The calculation of similarity between two 3D models is then transformed into the computation of similarity between two symbol sequences.

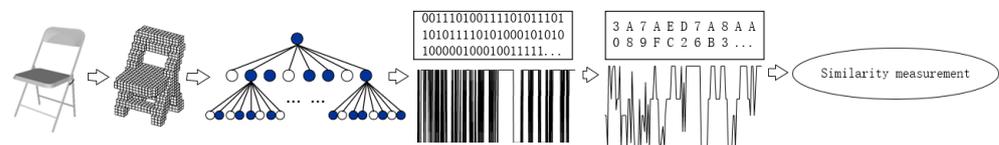


Figure 9. Feature representation processes based on symbolic operators.

When calculating the similarity between two models, the feature representation of the voxel model's symbolic sequence is divided based on adjacent model blocks in 3D space. Each pair of characters represents the status of a model block. When calculating the similarity of each model block, only the corresponding symbolic sequence distances need to be computed. To preserve the structural information of adjacent voxels within model blocks, this approach maps the symbolic sequences to a 16×16 plane coordinate system. Each model block is represented by a point in the corresponding plane. The Euclidean distance between two points in the plane is calculated as the similarity distance measure between the model blocks. The closer the points are in the plane, the more similar the model blocks are. By mapping all symbolic sequences to the plane coordinate system and calculating the distances between each unit, the total sum of distances between all sequences is obtained as the final similarity distance of the 3D voxel model. The process of the similarity measurement for 3D voxel models is as follows:

For each 3D voxel model M , its serialized representation with a length of n is obtained as follows:

$$M = \{m_1, m_2, \dots, m_n\} \quad (8)$$

The segmented sequence S is obtained by further dimensionality reduction representation:

$$S = \text{segMapper} = \{s_1, s_2, \dots, s_{n/8}\} \quad (9)$$

The sequence data are obtained by the sign operator to obtain a sequence of symbols of length w :

$$H = hvaMapper(S) = \{h_1, h_2, \dots, h_w\} \quad (10)$$

Map the characters into a hexadecimal flat grid to obtain the point sequence P for each model block:

$$P = htpMapper(H) = \{p_1, p_2, \dots, p_{w/2}\} \quad (11)$$

$p_i = (p_{ix}, p_{iy})$ PD (Pq, Pc) represents the Euclidean distance between sequences Pq and Pc , used to measure their similarity. The calculation method is as follows:

$$PD(Pq, Pc) = \sqrt[2]{\sum_{i=1}^w \left((htpMapper(h_i^q) - htpMapper(h_i^c))^2 \right)} \quad (12)$$

$$= \sqrt[2]{\sum_{i=1}^n (p_i^q - p_i^c)^2} \quad (13)$$

$$= \sqrt[2]{\sum_{i=1}^n \left((p_{ix}^q - p_{ix}^c)^2 + (p_{iy}^q - p_{iy}^c)^2 \right)} \quad (14)$$

As shown in Figure 10, for a specific local model block in the 3D voxel model, the binary encoding of the model block obtained through the Hilbert curve is represented as (0110 0011) and (1100 0111), respectively. Using VSO representation, the corresponding symbol sequence representations are "63" and "C7". These symbol sequences are then mapped to a 16×16 plane coordinate system, resulting in the coordinates (6,3) and (12,7), respectively. By calculating the Euclidean distance between these two points, we can obtain the similarity distance between the two model blocks. By sequentially calculating the distances between the corresponding points of all segmented model blocks in the 3D voxel model, we can determine the overall similarity between the 3D models.

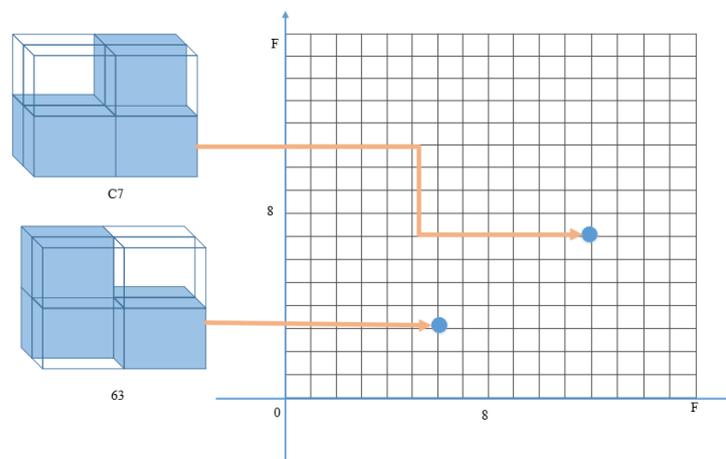


Figure 10. Similarity measurement based on VSO.

After the voxelization of the 3D model based on the Hilbert curves in Section 3.1 and the feature representation of the three-dimensional voxel model based on the symbolic operators in Section 3.2, we obtain a sequence M representing the features of the 3D model. Diminishing sequence M results in S , which undergoes the operation of the symbol operator D to produce the symbol sequence H . Subsequently, mapping each character in H to the plane coordinate system yields the corresponding point sequence P for the model. By calculating the Euclidean distance for each corresponding point in the point sequence, we can determine the similarity between the final 3D models.

3.4. Summary

This section mainly introduces the architecture of 3D model retrieval based on the VSO. Firstly, it elaborates on how to combine Hilbert filling curves with octrees to achieve a voxelization method based on Hilbert filling curves. Then, for voxel models, it adopts a sequence representation method for dimensionality reduction and expression, representing three-dimensional voxel models as a series of symbolic sequences based on symbolic operators. Finally, based on the symbolic sequence representation method, corresponding similarity distance metrics are proposed to calculate the similarity between different 3D models.

4. Experimental Validation

This chapter primarily focuses on the experimental evaluation and result analysis of several research aspects proposed in the previous two chapters, including model voxelization, feature representation methods, and similarity measurement methods. The experimental evaluation is conducted based on experimental data, evaluation metrics methods, and results, with a comparative analysis from aspects such as the classification accuracy of similarity measurement, dimensionality reduction rate, etc.

4.1. Experimental Data

The experimental data used in this study are the ModelNet dataset provided by Princeton University. The ModelNet dataset consists of 127,915 CAD models divided into two subsets: ModelNet10 and ModelNet40. ModelNet10 contains 10 categories with a total of 4899 CAD models, including training and testing sets. The training set consists of 3991 models, while the testing set consists of 908 models. ModelNet40 contains 40 categories with a total of 12,311 3D models, also divided into training and testing sets. The training set consists of 9843 models, and the testing set consists of 2468 models. The models in the database have standardized coordinates, with the main axis oriented vertically, and they have been normalized based on translation and rotation. The dataset includes virtual 3D models from various sources, such as airplanes, and tables.

4.2. Evaluation Metrics

The evaluation metrics used to validate the effectiveness of 3D model retrieval in this paper include the following:

- Accuracy:

Accuracy is a common metric used to evaluate classification models, representing the percentage of correct predictions among all samples. In binary classification problems, all classifications can be divided into positive class (Positive) and negative class (Negative). Assuming that one class is considered as the positive class in the classification results of 3D models, while the other class or several classes are considered as the negative class, where correctly predicting a positive instance as positive is True Positive (TP), correctly predicting a negative instance as negative is True Negative (TN), incorrectly predicting a positive instance as negative is False Positive (FP), and incorrectly predicting a negative instance as positive is False Negative (FN). The formula for accuracy is calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\% \quad (15)$$

- Confusion Matrix:

The confusion matrix compares the relationship between the classification results and the ground truth of each category in matrix form. Each column in the matrix represents a predicted category, each row represents a true category, the data in each column indicate the proportion of models predicted as that category, and the data in each row indicate the proportion of true models in that category. The contents of the confusion matrix are shown in Table 3.

Table 3. Confusion Matrix.

		Predicted Value	
		Positive	Negative
Ground truth	Positive	TP	FN
	Negative	FP	TN

4.3. Experimental Methods

The experimental part of this paper is mainly divided into two parts for evaluation and analysis:

- Three-dimensional model voxelization particle size analysis experiment:

The main objective of this experiment is to validate the effectiveness of the voxelization method proposed in this paper, based on Hilbert curves. Voxelization granularity is an important factor that affects the results and efficiency of 3D voxel model retrieval. The experiment analyzes the classification accuracy and computation time of 3D voxel models under different voxelization granularities.

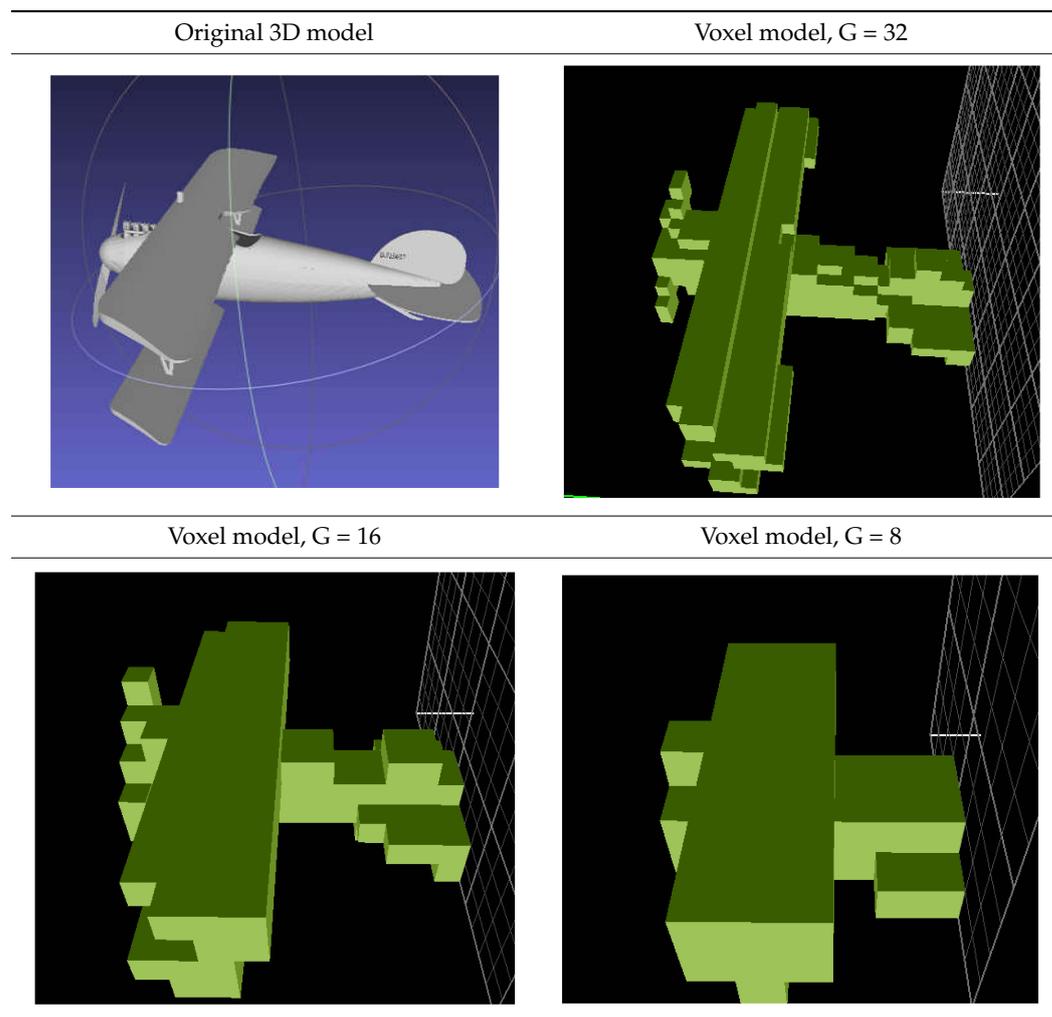
- Three-dimensional model similarity measurement method and feature dimension analysis experiment:

The main purpose of this experiment is to validate the effectiveness of the proposed 3D model feature representation method and similarity measurement in this paper. To reduce the influence of classifiers, the experiment uses the classical nearest neighbor classification method KNN [54] for classification experiments. In order to demonstrate the superiority of the proposed method more intuitively, a comparative analysis with classical 3D model feature representation methods such as the ED, SAX, SPH, LFD, and 3D ShapeNets is conducted. The ED is a classical distance measurement method that always ensures high accuracy. SAX is an efficient symbolic sequence representation method that uses the Piecewise Aggregate Approximation (PAA) method for rapid dimensionality reduction. SPH, LFD, and 3D ShapeNets are commonly used feature representation methods in the field of 3D model retrieval. Additionally, some parameters involved in the experiment can also affect the results, so the experiment includes an analysis and evaluation of the classification accuracy under different parameter settings.

4.4. Analysis of Experimental Results

- Three-dimensional model voxelization particle size analysis experiment:

When the order of the space-filling curve is n , the corresponding voxelization granularity is 2^n . The length of the symbol sequence obtained through VSO feature representation is 2^{n^3} . Voxel data obtained after the voxelization of 3D models can to some extent describe the information of the models. A higher voxelization granularity value results in a more detailed representation of the 3D model. However, pursuing a high granularity voxelization for 3D models alone is not practical. As the voxelization granularity increases, the scale of the voxel model data in the 3D model database also increases, requiring more storage space and increasing the computational time, which becomes an important factor in the query time. Table 4, using an airplane model from the ModelNet40 database as an example, demonstrates the voxel models at different voxelization granularities. Compared to the original 3D model, the voxel models show varying degrees of loss in detail features. However, even at higher voxelization granularities, the voxel models are still able to represent the shape of the airplane and differentiate it from other categories.

Table 4. Three-dimensional voxel models at different granularities.

In the query process of 3D models, both the accuracy of query matching and the query time consumption are equally important. If the accuracy is low and the retrieved results differ significantly from the given data, the goal of returning similar models cannot be achieved. Conversely, if the query time consumption is too long and the efficiency is low, it becomes difficult to apply the method in practical production and daily life. Table 5 compares the classification accuracy and average query time for different voxelization granularities in the ModelNet10 dataset. The calculation involves the feature representation method VSO proposed in this paper and the similarity measurement. The research experiment indicates that as the voxel resolution continues to increase, the fine details of the models become more prominent, resulting in increased differences between different models. The classification accuracy of the dataset also shows a slight improvement. However, the voxel data size increases dramatically, leading to a significant increase in the average query time. In practical application scenarios, if a high accuracy of the retrieval results is not a strict requirement, one can choose a relatively lower voxel resolution in order to achieve efficient queries, instead of solely pursuing high accuracy at the cost of a longer retrieval time.

Table 5. The influence of voxel granularity on model classification and query.

Voxelized Granularity	Classification Accuracy	Average Calculation Time (ms)
$2^2, 4 \times 4 \times 4$	75.3%	0.208
$2^3, 8 \times 8 \times 8$	80.0%	3.398
$2^4, 16 \times 16 \times 16$	84.0%	10.392
$2^5, 32 \times 32 \times 32$	85.9%	118.163

- Three-dimensional model similarity measurement method and feature dimension analysis experiment:

In this section, the feature representation method and similarity measurement based on the symbolic operator are adopted, and the experimental results are compared with other common 3D model feature representation methods and similarity measurement methods. In this experiment, the ModelNet10 and ModelNet40 datasets were used to calculate the feature representation method and similarity measure of the 3D model in the classification accuracy of a certain feature dimension. To compare, the experiment utilized light field descriptors (LFDs), spherical harmonic descriptors (SPHs), and features learned by the 3DShapeNets for the feature representation of 3D models. The LFD employs multiple camera arrays to capture multi-view 2D images for describing 3D models; the SPH uses spherical harmonics to analyze and compute sampled points on the shape of 3D models, describing their geometric and structural features; whereas 3DShapeNets utilizes a convolutional deep belief network to represent the shape of 3D models using the probability distribution of binary variables on a three-dimensional voxel grid. The descriptor vector dimensions are 4700 and 544 for the LFD and SPH, respectively, while the feature vector dimension for 3D ShapeNet is 4000. The experimental results are shown in Table 6.

Table 6. Comparison of the classification accuracy of the model.

Algorithm	Feature Dimensions	Dataset	
		ModelNet 10	ModelNet 40
SPH [29]	544	0.798	0.682
LFD [17]	4700	0.799	0.754
3DshapeNets [35]	4000	0.792	0.720
ED [47]	4096	0.827	0.727
SAX [41]	1024	0.652	0.692
VSO	128	0.840	0.726

By comparing the data, it can be found that the method proposed in this paper has the best classification performance rate on the ModelNet10 dataset, reaching 84%, and the classification accuracy on the ModelNet40 dataset is only 2.8% lower than that of the best LFD. Although the LFD method has the highest classification accuracy on the ModelNet40 dataset, the corresponding feature dimension is also large, the computational complexity of the 3D model classification calculation is higher, and the pressure of feature vector storage is large. To validate the effectiveness of the proposed similarity measurement method, the experiment incorporates the ED and SAX methods. By combining the voxelized 3D models and representation methods, the classification accuracy is computed for the ModelNet10 and ModelNet40 datasets with a voxel resolution of 4. The Euclidean distance (ED) achieves a relatively high accuracy on both the ModelNet10 and ModelNet40 datasets. However, the ED does not effectively reduce the dimensionality of the data. The feature dimension of a model increases by several tens of times compared to the VSO method, which poses greater challenges in large-scale data environments. The SAX method can reduce the dimensionality based on the chosen parameters. When selecting a feature dimension of 1024, SAX, which is based on the Piecewise Aggregate Approximation (PAA) method, only retains the mean values of data segments while ignoring the fluctuation information of the

sequence. This leads to a loss of fine-grained details in the models, resulting in a lower classification accuracy on both the ModelNet10 and ModelNet40 datasets compared to the VSO representation and similarity measurement methods. The experimental results demonstrate that the proposed algorithm exhibits notable advantages in the classification of 3D models.

Figure 11 shows the confusion matrix generated based on the classification results of the ModelNet10 dataset. In the confusion matrix, the rows represent the true class labels, and the columns represent the predicted class labels. The diagonal elements of the matrix represent the classification accuracy of each class in the ModelNet10 dataset. The off-diagonal elements represent the misclassification rates, indicating the probability of incorrect predictions for each class being assigned to other classes.

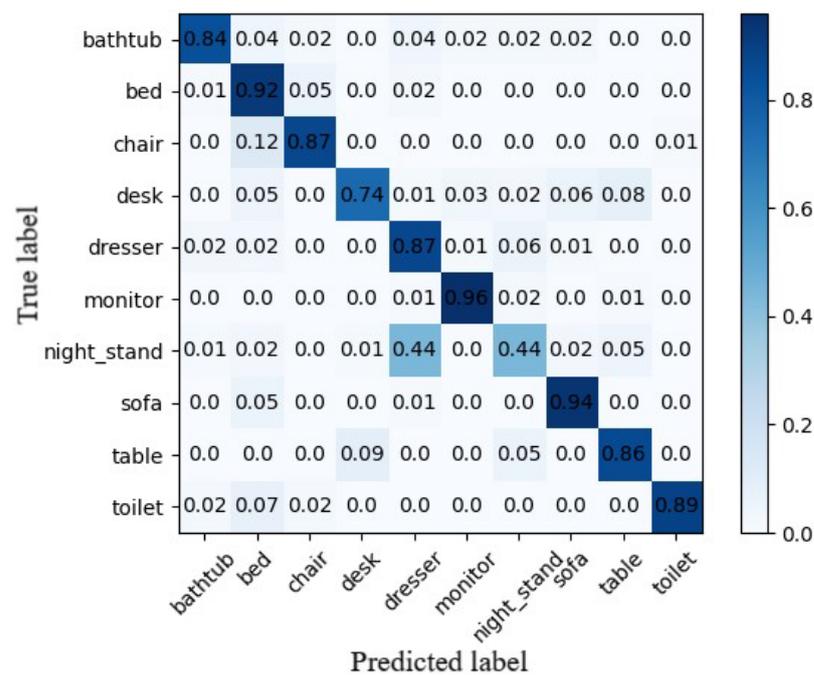


Figure 11. ModelNet10 classification confusion matrix.

Observing the experimental data, it is found that there are some types of data in the ModelNet10 dataset that are easy to confuse and misseparate, among which the night_stand class and dresser class, desk class, and table class are the most obvious. Figure 12 shows the confusing model in the night_stand class and the dresser class.

The “night_stand” and “dresser” classes have similar cube-like shapes, making it challenging to distinguish their details when using voxel representations. The main difference between the two classes lies in the fact that dressers are typically larger in size. However, the VSO representation does not explicitly capture the volume information when generating the serialized models, making it difficult to differentiate the classes based on volume. However, this characteristic of VSO representation can be advantageous for objects within the same class but with different sizes. It allows for judging the similarity of objects based solely on their shapes while disregarding the differences in volume.



Figure 12. Confusing night_stand and dresser models.

The experiment has shown that different feature dimensions have a significant impact on the classification efficiency of 3D models. The VSO representation method utilizes the voxelization of 3D models to obtain voxel data, followed by dimensionality reduction through the filling of spatial curves of different orders. This approach not only preserves the shape information of 3D models but also retains the neighboring structural information. Figure 13 illustrates the feature representation dimensions of the aforementioned algorithm. The feature descriptors of the aforementioned algorithms can essentially be converted into vectors. The vector dimensions of the LFD descriptor and SPH descriptor are 4700 and 540, respectively, while the vector dimension of the 3D shape features learned by 3DShapeNets is 4000. In the experiments, the three-dimensional models were voxelized with a voxel granularity of 4 using the method proposed in this paper. The feature vector dimensions for similarity measurements of three-dimensional models using the ED, SAX, and VSO methods are 4096, 1024, and 128, respectively.

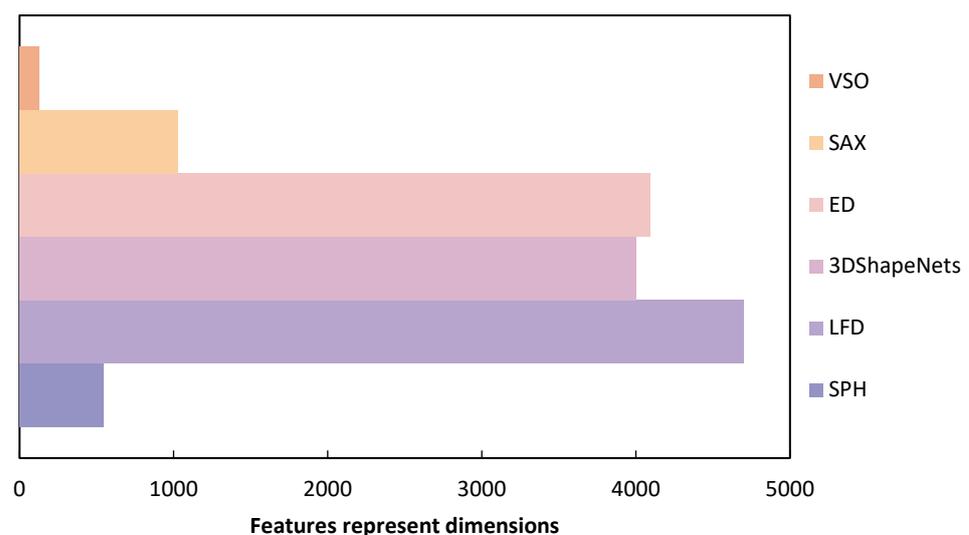


Figure 13. The section represents the characteristic dimensions of the method.

In the classification experiments, the VSO representation method achieves a high classification accuracy with a minimum feature dimension of only 128, which is much smaller than other popular feature representation methods such as 3DShapeNets. Addi-

tionally, it does not require high storage or computational performance from the computer. This characteristic makes the proposed method suitable for storing and processing massive amounts of 3D models.

5. Discussion

This paper conducts comparative analyses using the publicly available dataset ModelNet to validate the effectiveness of the proposed method. Regarding the feature extraction of 3D voxel models, the choice of voxelization granularity simultaneously affects the accuracy of 3D model classification and the time consumed in experiments. An appropriate voxelization granularity can achieve a better classification accuracy in a shorter time. In terms of the similarity measurement, compared with classical 3D model retrieval methods such as SPH, LFD, and 3D ShapeNets, the VSO representation and its similarity measurement method achieve a classification accuracy of 84% on the ModelNet10 dataset, with a feature dimension of only 128, which is superior to other algorithmic results. It is worth noting that compared with other algorithms, the VSO representation method has the highest dimensionality reduction rate, with the feature dimension being less than half of other algorithms, without high requirements for computer storage and computing performance. However, the method proposed in this paper operates on 3D model voxel forms, and when the voxel granularity increases, the voxelization time cost is higher. Additionally, since the feature extraction method VSO only considers the structural information of 3D models and ignores other attribute information, the retrieval accuracy of 3D models has not reached the most ideal state. In the future, we aim to overcome these limitations and achieve the efficient and accurate retrieval of 3D models.

6. Conclusions and Future Works

The main focus of this paper is on the similarity measurement and indexing methods for 3D voxel models. To reduce information loss and avoid the curse of dimensionality, which can lead to decreased retrieval performance, a series of solutions are proposed for 3D model classification and retrieval. Key issues in 3D model retrieval research include how to extract features from 3D models to retain more model information, and how to choose suitable feature descriptors and similarity measurement methods to reduce computational complexity. Addressing these issues, this paper first proposes a voxelization method based on space-filling curves to obtain 3D voxel data that preserve the structural features of 3D models to some extent. Then, it introduces the VSO feature representation method based on symbolic operators to effectively reduce the feature dimensionality of the data. Finally, a similarity measurement method based on symbolic sequences is proposed to achieve efficient retrieval on 3D model databases. Experiments conducted on the publicly available ModelNet dataset demonstrate that the proposed methods can improve the efficiency and accuracy of 3D model similarity calculation. Moreover, they do not require high demands on computer storage and computing performance, making them suitable for storing and processing massive 3D models. Despite the partial success achieved in the aforementioned research, there are still several areas that require further investigation:

1. Feature extraction of 3D voxel models is based on voxel representation, preserving the complete voxel data of 3D models. However, many 3D models have complex structures composed of numerous edges and faces. When the voxel granularity increases, the time cost of voxelization also increases. Therefore, improving the efficiency of voxelization for 3D models is an important area for future research.
2. The VSO representation and similarity measurement methods for 3D voxel models currently focus mainly on shape and structural information. However, real-world 3D models also include rich attribute information such as color, texture, and density. Incorporating other attribute information in addition to shape and structural information is expected to enhance the accuracy of 3D model classification and retrieval and broaden the application scope. Therefore, exploring how to fully utilize vari-

- ous attribute information of models and designing similarity measurement methods applicable to multi-attribute features are worthwhile directions for further research.
3. In the field of geographic information, there are many three-dimensional geographic spatial objects, such as underground geological bodies like rock masses and ore bodies, as well as landform features like mountains and valleys. Utilizing the method proposed in this paper to voxelize these three-dimensional geographic spatial objects and exploring how to use them for feature representation and similarity measurement is one of our future research directions. This research can facilitate the understanding, analysis, and management of the Earth's surface and subsurface spaces, providing more possibilities for the application of geographic information systems.

Author Contributions: Zhenwen He wrote the test codes and performed the experiments. Xianzhen Liu designed the architecture of 3D model retrieval based on VSO. Chunfeng Zhang analyzed the experimental dataset and revised the manuscript. Xianzhen Liu wrote the paper, and all authors reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (2022YFF0801203, 2022YFF0801200) and the National Natural Science Foundation of China (41972306).

Data Availability Statement: The analysis datasets during the current study are available from the corresponding author on reasonable request (zwhe@cug.edu.cn).

Acknowledgments: Thanks for the reviewers' constructive comments and they really help us improve this paper.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Dobroś, K.; Hajto-Bryk, J.; Zarzecka, J. Application of 3D-printed teeth models in teaching dentistry students: A scoping review. *Eur. J. Dent. Educ.* **2023**, *27*, 126–134. [\[CrossRef\]](#)
2. Xu, X.; Rioux, T.P.; Castellani, M.P. Three dimensional models of human thermoregulation: A review. *J. Therm. Biol.* **2023**, *112*, 103491. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Liu, W.; Zang, Y.; Xiong, Z.; Bian, X.; Wen, C.; Lu, X.; Wang, C.; Junior, J.M.; Goncalves, W.N.; Li, J. 3D building model generation from MLS point cloud and 3D mesh using multi-source data fusion. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *116*, 103171. [\[CrossRef\]](#)
4. Li, P. Production of Film and Television Animation Based on Three-Dimensional Models Based on Deep Image Sequences. In Proceedings of the Machine Learning, Image Processing, Network Security and Data Sciences: 4th International Conference, MIND 2022, Virtual Event, 19–20 January 2023; Proceedings, Part II; Springer: Cham, Switzerland, 2023; pp. 131–138.
5. Zhang, C.; Chen, T. Efficient feature extraction for 2D/3D objects in mesh representation. In Proceedings of the 2001 International Conference on Image Processing (Cat. No. 01CH37205), Thessaloniki, Greece, 7–10 October 2001; Volume 3, pp. 935–938. [\[CrossRef\]](#)
6. Serafin, J.; Olson, E.; Grisetti, G. Fast and robust 3D feature extraction from sparse point clouds. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 4105–4112. [\[CrossRef\]](#)
7. Lu, K.; Wang, Q.; Xue, J.; Pan, W. 3D model retrieval and classification by semi-supervised learning with content-based similarity. *Inf. Sci.* **2014**, *281*, 703–713. [\[CrossRef\]](#)
8. Fu, T. A review on time series data mining. *Eng. Appl. Artif. Intell.* **2011**, *24*, 164–181. [\[CrossRef\]](#)
9. Wattenberg, M. A note on space-filling visualizations and space-filling curves. In Proceedings of the IEEE Symposium on Information Visualization, INFOVIS 2005, Minneapolis, MN, USA, 23–25 October 2005; pp. 181–186.
10. Wang, P.S.; Liu, Y.; Guo, Y.X.; Sun, C.Y.; Tong, X. O-cnn: Octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–11. [\[CrossRef\]](#)
11. Jjumba, A.; Dragičević, S. Towards a voxel-based geographic automata for the simulation of geospatial processes. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 206–216. [\[CrossRef\]](#)
12. Piris, G.; Herms, I.; Griera, A.; Colomer, M.; Arnó, G.; Gomez-Rivas, E. 3DHIP-calculator—A new tool to stochastically assess deep geothermal potential using the heat-in-place method from voxel-based 3D geological models. *Energies* **2021**, *14*, 7338. [\[CrossRef\]](#)
13. Fisher-Gewirtzman, D.; Shashkov, A.; Doytsher, Y. Voxel based volumetric visibility analysis of urban environments. *Surv. Rev.* **2013**, *45*, 451–461. [\[CrossRef\]](#)

14. Khan, M.S.; Kim, I.S.; Seo, J. A boundary and voxel-based 3D geological data management system leveraging BIM and GIS. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *118*, 103277. [[CrossRef](#)]
15. Min, P.; Kazhdan, M.; Funkhouser, T. A comparison of text and shape matching for retrieval of online 3D models. In Proceedings of the Research and Advanced Technology for Digital Libraries: 8th European Conference, ECDL 2004, Bath, UK, 12–17 September 2004; Proceedings 8; Springer: Berlin/Heidelberg, Germany, 2004; pp. 209–220.
16. Kassimi, M.A.; Elbeqqali, O. Semantic based 3D model retrieval. In Proceedings of the 2012 International Conference on Multimedia Computing and Systems, Tangiers, Morocco, 10–12 May 2012; pp. 195–199.
17. Tangelder, J.W.H.; Veltkamp, R.C. A survey of content based 3D shape retrieval methods. *Multimed. Tools Appl.* **2008**, *39*, 441–471. [[CrossRef](#)]
18. Pei, Y.; Gu, K. Overview of content and semantic based 3D model retrieval. *J. Comput. Appl.* **2020**, *40*, 1863–1872.
19. Chen, D.Y.; Tian, X.P.; Shen, Y.T.; Ouhyoung, M. On visual similarity based 3D model retrieval. In *Proceedings of the Computer Graphics Forum*; Blackwell Publishing, Inc.: Oxford, UK, 2003; Volume 22, pp. 223–232.
20. Regli, W.C.; Cicirello, V.A. Managing digital libraries for computer-aided design. *Comput.-Aided Des.* **2000**, *32*, 119–132. [[CrossRef](#)]
21. Zaharia, T.; Preteux, F.J. 3D-shape-based retrieval within the MPEG-7 framework. Nonlinear image processing and pattern analysis xii. *SPIE* **2001**, *4304*, 133–145.
22. Biasotti, S.; Marini, S.; Mortara, M.; Patane, G.; Spagnuolo, M.; Falcidieno, B. 3D shape matching through topological structures. In Proceedings of the Discrete Geometry for Computer Imagery: 11th International Conference, DGCI 2003, Naples, Italy, 19–21 November 2003; Proceedings 11; Springer: Berlin/Heidelberg, Germany, 2003; pp. 194–203.
23. Kim, H.; Cha, M.; Mun, D. Shape distribution-based retrieval of 3D CAD models at different levels of detail. *Multimedia Tools Appl.* **2017**, *76*, 15867–15884. [[CrossRef](#)]
24. Li, B.; Godil, A.; Johan, H. Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3D model retrieval. *Multimed. Tools Appl.* **2014**, *72*, 1531–1560. [[CrossRef](#)]
25. Van De Weijer, J.; Schmid, C. Coloring local feature extraction. In Proceedings of the Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Proceedings, Part II 9; Springer: Berlin/Heidelberg, Germany, 2006; pp. 334–348.
26. Imrie, F.; Bradley, A.R.; van der Schaar, M.; Deane, C.M. Deep generative models for 3D linker design. *J. Chem. Inf. Model.* **2020**, *60*, 1983–1995. [[CrossRef](#)]
27. Gezawa, A.S.; Zhang, Y.; Wang, Q.; Yunqi, L. A review on deep learning approaches for 3D data representations in retrieval and classifications. *IEEE Access* **2020**, *8*, 57566–57593. [[CrossRef](#)]
28. Lian, Z.; Godil, A.; Bustos, B.; Daoudi, M.; Hermans, J.; Kawamura, S.; Kurita, Y.; Lavoua, G.; Suetens, P.D. Shape retrieval on non-rigid 3D watertight meshes. In Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR), Llundudno, UK, 10 April 2011.
29. Cheng, H.C.; Lo, C.H.; Chu, C.H.; Kim, Y.S. Shape similarity measurement for 3D mechanical part using D2 shape distribution and negative feature decomposition. *Comput. Ind.* **2011**, *62*, 269–280. [[CrossRef](#)]
30. Kazhdan, M.; Funkhouser, T.; Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3D shape descriptors. In Proceedings of the Symposium on Geometry Processing, Aachen, Germany, 23–25 June 2003; Volume 6, pp. 156–164.
31. Zhang, J.; Yao, Y.; Deng, B. Fast and robust iterative closest point. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3450–3466. [[CrossRef](#)]
32. Hilaga, M.; Shinagawa, Y.; Kohmura, T.; Kunii, T.L. Topology matching for fully automatic similarity estimation of 3D shapes. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; pp. 203–212.
33. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Aligning point cloud views using persistent feature histograms. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3384–3391.
34. Salti, S.; Tombari, F.; Di Stefano, L. SHOT: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **2014**, *125*, 251–264. [[CrossRef](#)]
35. Maturana, D.; Scherer, S. Voxnet: A 3D convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
36. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
37. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view convolutional neural networks for 3D shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
38. Garcia-Garcia, A.; Gomez-Donoso, F.; Garcia-Rodriguez, J.; Orts-Escolano, S.; Cazorla, M.; Azorin-Lopez, J. Pointnet: A 3D convolutional neural network for real-time object class recognition. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 1578–1584.
39. Sun, D.; Qu, L. Survey on Feature Representation and Similarity Measurement of Time Series. *J. Front. Comput. Sci. Technol.* **2021**, *15*, 195–205.

40. Agrawal, R.; Faloutsos, C.; Swami, A. Efficient similarity search in sequence databases. In *Proceedings of the International Conference on Foundations of Data Organization & Algorithms*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 69–84.
41. Chan, K.P.; Fu, A.W.C. Efficient time series matching by wavelets. In *Proceedings of the 15th International Conference on Data Engineering*; IEEE Computer Society: Los Alamitos, CA, USA, 1999; pp. 126–133.
42. Keogh, E.; Chakrabarti, K.; Pazzani, M.; Mehrotra, S. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowl. Inf. Syst.* **2001**, *3*, 263–286. [[CrossRef](#)]
43. Lin, J.; Keogh, E.; Wei, L.; Lonardi, S. Experiencing SAX: A novel symbolic representation of time series. *Data Min. Knowl. Discov.* **2007**, *15*, 107–144. [[CrossRef](#)]
44. Korn, F.; Jagaciish, H.V.; Faloutsos, C. Efficiently supporting ad hoc queries in large datasets of time sequences. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Tucson, AZ, USA, 13–15 May 1997; ACM: New York, NY, USA, 1997; pp. 289–300.
45. Mor, B.; Garhwal, S.; Kumar, A. A systematic review of hidden Markov models and their applications. *Arch. Comput. Methods Eng.* **2021**, *28*, 1429–1448. [[CrossRef](#)]
46. Fokianos, K.; Fried, R.; Kharin, Y.; Voloshko, V. Statistical analysis of multivariate discrete-valued time series. *J. Multivar. Anal.* **2022**, *188*, 104805. [[CrossRef](#)]
47. Chen, H.Y.; Liu, C.H.; Sun, B. Survey on similarity measurement of time series data mining. *Control Decis.* **2017**, *32*, 1–11.
48. Bai, S.; Qi, H.D.; Xiu, N. Constrained best Euclidean distance embedding on a sphere: A matrix optimization approach. *SIAM J. Optim.* **2015**, *25*, 439–467. [[CrossRef](#)]
49. Hsu, C.J.; Huang, K.S.; Yang, C.B.; Guo, Y.P. Flexible dynamic time warping for time series classification. *Procedia Comput. Sci.* **2015**, *51*, 2838–2842. [[CrossRef](#)]
50. Wang, S.; Wen, Y.; Zhao, H. Study on the similarity query based on LCSS over data stream window. In *Proceedings of the IEEE 2015 IEEE 12th International Conference on e-Business Engineering*, Beijing, China, 23–25 October 2015; pp. 68–73.
51. Jagadish, H.V. Analysis of the Hilbert curve for representing two-dimensional space. *Inf. Process. Lett.* **1997**, *62*, 17–22. [[CrossRef](#)]
52. Kilimci, P.; Kalipsiz, O. Indexing of spatiotemporal Data: A comparison between sweep and z-order space filling curves. In *Proceedings of the IEEE International Conference on Information Society (i-Society 2011)*, London, UK, 27–29 June 2011; pp. 450–456.
53. Mokbel, M.F.; Aref, W.G.; Kamel, I. Analysis of multi-dimensional space-filling curves. *Geoinformatica* **2003**, *7*, 179–209. [[CrossRef](#)]
54. Levchenko, O.; Kolev, B.; Yagoubi, D.E.; Akbarinia, R.; Masegla, F.; Palpanas, T.; Shasha, D.; Valduriez, P. BestNeighbor: Efficient evaluation of kNN queries on large time series databases. *Knowl. Inf. Syst.* **2021**, *63*, 349–378. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.