J Vis
https://doi.org/10.1007/s12650-021-00770-2

REGULAR PAPER

Haoran Dai · Yubo Tao · Xiangyang He · Hai Lin

IsoExplorer: an isosurface-driven framework for 3D shape analysis of biomedical volume data

Received: 8 July 2021 / Accepted: 23 July 2021

Abstract The high-resolution scanning devices developed in recent decades provide biomedical volume datasets that support the study of molecular structure and drug design. Isosurface analysis is an important tool in these studies, and the key is to construct suitable description vectors to support subsequent tasks, such as classification and retrieval. Traditional methods based on handcrafted features are insufficient for dealing with complex structures, while deep learning-based approaches have high memory and computation costs when dealing directly with volume data. To address these problems, we propose IsoExplorer, an isosurface-driven framework for 3D shape analysis of biomedical volume data. We first extract isosurfaces from volume data and split them into individual 3D shapes according to their connectivity. Then, we utilize octree-based convolution to design a variational autoencoder model that learns the latent representations of the shape. Finally, these latent representations are used for low-dimensional isosurface representation and shape retrieval. We demonstrate the effectiveness and usefulness of IsoExplorer via isosurface similarity analysis, shape retrieval of real-world data, and comparison with existing methods.

Keywords Isosurface · Shape analysis · Variational autoencoder

1 Introduction

The development of high-precision 3D scanning technology has enabled humans to observe objects at an extremely fine scale over the past decade. For instance, the structure of apoferritin, obtained by cryogenic electron microscopy (cryo-EM) at a 1.25 Å-resolution, provided unprecedented structural details (Yip et al. 2020). As of May 2021, the Electron Microscopy Data Bank (EMDB) (Lawson et al. 2016) provides more than 15,000 electron microscopy data with different resolutions, offering rich resources for molecular structure studies. These achievements are useful for biological data analysis, drug development, and disease detection. Based on the meta-information on the data, researchers can explore and search the dataset (Liu et al. 2019), but this is limited by the completeness of that information, whereas searching based on the sample structure does not have this constraint and can help researchers to retrieve relevant data accurately. Analysis and comprehension of the 3D shapes embedded in the scanned objects (e.g., classification and retrieval) naturally becomes a key task.

Generating isosurfaces from scanning data (i.e., biomedical volume data) for subsequent 3D shape visual analysis are a common workflow, and the key technique is to construct an effective representation of the isosurfaces. Castillo-Barnes et al. (2020) leveraged morphological measurement-based isosurface analysis to find new biomarkers in scanning data that related to Parkinson's Disease. El-Baz et al. (2011) introduced

H. Dai Y. Tao (✉) · X. He H. Lin
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
E-mail: taoyubo@cad.zju.edu.cn

Published online: 19 August 2021

the spherical harmonic approach to analyze 3D surfaces in computed tomography (CT) data for early diagnosis of malignant lung nodules. These handcrafted features work well when the shape structure is relatively simple, however, as the resolution continues to increase and the structure of the scanned object becomes finer, it is increasingly challenging to manually design features that accurately characterize the target structure.

To address the shortcomings of manual feature design, various data-driven approaches inspired by the field of computer vision use 3D convolutional neural networks (3D CNNs) to automatically extract 3D shape features. For instance, Han et al. (2018) proposed FlowNet, which represents a stream surface with a binary volume and then automatically generates its feature vector with an autoencoder model, to assist with stream surface selection. However, since 3D CNN operations have high memory costs (e.g., 11GB of graphics memory in FlowNet can only handle data up to 64^3), most approaches require downsampling or slicing to pre-process the data, which sacrifices the benefits of high-resolution volume data.

In this paper, to address the above issues, we propose IsoExplorer, an isosurface-driven framework that facilitates 3D shape analysis of biomedical volume data. Specifically, we aim to automatically extract probabilistic latent features (i.e., latent representation) for 3D shapes embedded in biomedical volume data. The proposed framework comprises three steps: (1) To reduce the memory and computational cost, we first separate the different isosurfaces into 3D shapes based on the connection relation. (2) We introduce the octree-based CNN (O-CNN) (Wang et al. 2017a) to further improve the efficiency of the convolution operations, and design a variational autoencoder model which has stochastics to learn the latent representation of each 3D shape in the probabilistic latent space. (3) We introduce low-dimensional isosurface representation for isosurface similarity analysis, and locality-sensitive hashing (LSH) to store the latent representations of 3D shapes for fast shape retrieval. To validate the effectiveness and usefulness of IsoExplorer, we evaluate it with real-world cypo-EM, photon microscopy, and CT scanning data, and compare it with other methods.

The main contributions of this paper are as follows.

- We design an isosurface-driven shape extraction approach based on isosurface generation and maximum connected component partitioning, to explore the explicit shapes in biomedical volume data.
- We introduce a variational autoencoder model to learn the latent representation for each shape in biomedical volume data and further utilize it for low-dimensional isosurface representation and shape similarity measurement.
- We validate the effectiveness of IsoExplorer by performing isosurface similarity analysis and similar shape retrieval within one volume/between different biomedical volumes using multiple datasets and comparing with existing methods.

2 Related work

In this section, we draw on prior work on 3D shape analysis and deep learning for volume visualization.

2.1 3D shape analysis

3D shape analysis is widely used for various tasks, e.g., object recognition, shape retrieval, and comparative analysis. The key technique to support these tasks is the shape description method, which produces a feature vector for a given shape while maintaining invariance to translation, scaling, and rotation (Loncaric 1998). There are two main kinds of shapes, i.e., line segments and surfaces, and the methods for generating feature vectors for them can be classified as feature-based and data-driven approaches.

Feature-based approaches mainly leverage the topological and statistical information hidden in 3D shapes to design a series of features to generate feature vectors. For 3D streamlines, Li et al. (2014) designed a spatially sensitive bag-of-features for streamline similarity computation. Wang et al. (2017b) further allowed users to define features of interest to find streamline segments with similar patterns. For 3D surfaces, Shinagawa et al. (1991) proposed a manifold-based method that utilized a Reeb graph to encode a shape. Zaharescu et al. (2009) proposed the 3D scale-invariant feature transform (SIFT) method to generate local features. In addition, Tabia et al. (2014) introduced covariance matrices that can exploit the different features from different modalities. For 3D scalar fields without explicit shapes, Thomas and Natarajan (2011) utilized a contour tree to find symmetric structure. Bruckner and Möller (2010) quantified the

similarity between isosurfaces from an information-theoretic perspective by mutual information. Furthermore, Wang et al. (2015) introduced 3D SIFT to facilitate user-defined pattern matching in 3D multi-field scalar data.

Data-driven approaches benefit from machine learning techniques, and their primary aim is to analyze large amounts of data to extract meaningful 3D shape information. Rostami et al. (2019) conducted a survey on data-driven 3D shape descriptors, and classified them into shallow (e.g., clustering-based and optimization-based) and deep (e.g., CNN-based and autoencoder-based) shapes from an algorithm perspective. Gao et al. (2016) introduced the rotation-invariant mesh difference (RIMD) representation, which encodes vertex positions and local rotations simultaneously, and further leveraged it for optimization-based surface reconstruction. Xie et al. (2015) proposed DeepShape, a discriminative autoencoder model that imposed the Fisher discrimination criterion, to generate high-level shape representations for shape retrieval. Han et al. (2019) proposed a permutation voxelization strategy to convert 3D shapes into 3D voxels and learn their hierarchical local features. Instead of extracting features directly from the 3D shape itself, Bai et al. (2017) considered that users usually observe a 3D shape from different 2D viewpoints, and trained a CNN-based model from projected images of the 3D shape to perform retrieval.

Among these methods, most of the state-of-the-art methods are based on deep learning (Rostami et al. 2019). As it is helpful to learn a task-independent latent representation of a shape for subsequent analysis, we expect that these representations can accurately capture the information embedded in various complex shapes, which is consistent with the goal of the autoencoder model, which minimizes reconstruction errors. Thus, we design a variational autoencoder model, which is more robust to noise in the data and has more structured latent space than the autoencoder model, to learn the latent representation of the shapes extracted from volume data.

2.2 Deep learning for volume visualization

The burgeoning deep learning techniques provide powerful tools for visualization and visual analytics of volume data. Typically, volume data can be visualized by direct volume rendering (DVR) or indirect volume rendering (IVR). The former generates results based on ray-casting (Kruger and Westermann 2003) and user-defined transfer functions, while the latter renders geometric primitives generated by fitting the values to volume data (e.g., the Marching Cube algorithm (Lorensen and Cline 1987) for isosurface generation).

Deep learning techniques have been applied to both DVR and IVR. For DVR, deep learning can exploit the learned features to assist in parameter setting in the visualization. For example, Engel and Ropinski (2021) proposed deep volumetric ambient occlusion (DVAO) that combines global unstructured information and 3D CNN operations to compute volumetric ambient occlusion and thereby enhance the quality of interactive DVR. Berger et al. (2018) leveraged a generative adversarial network (GAN) to learn a view-invariant latent space and encode how transfer functions affected the rendered results to assist users in transfer function design. Yang et al. (2019) proposed a CNN-based neural network model to evaluate the score of a specific rendered image for viewpoint recommendation in DVR. He et al. (2019) designed InSituNet with a GAN architecture to assist in exploring the parameter space in in-situ visualization. For IVR, deep learning can help analyze geometric primitives obtained from volume data in multiple perspectives. For example, Cheng et al. (2018) proposed a CNN-based model to derive characteristic feature vectors for voxels and then utilize them to generate a binarized volume for the Marching Cube algorithm. Han et al. (2018) utilized a 3D CNN-based autoencoder model to learn dense representations of stream surfaces and lines, and then used projection to assist with selection and clustering analysis. Weiss et al. (2019) leveraged a frame-recurrent neural network to introduce super-resolution techniques to IVR, reducing the amount of data samples required to render isosurfaces.

In addition, deep learning can be used for visual analysis between multiple volumes. Porter et al. (2019) proposed an autoencoder model to learn a representation for each volume in a time-varying multivariate dataset and then leveraged them for representative time step selection. Han et al. (2021) designed a GAN to enable exploration of multivariate time-varying data in variable selection and translation analysis. Tkachev et al. (2021) introduced a prediction model for spatiotemporal volume data, which can facilitate irregular process detection and time step selection. These works incorporate deep learning effectively into the visual analytics workflow of volume data. However, most of them can only handle data smaller than 128^3 due to GPU memory limitations, and thus require downsampling or slicing of the data. This leads to underutilization of the information provided in volume data and a potential inability to accurately capture the 3D

shapes embedded in the data. Therefore, we introduce maximum connected component partitioning in isosurface generation and the O-CNN (Wang et al. 2017a) operations in the variational autoencoder model to increase the size of volume data the model could handle within the GPU memory constraints.

3 IsoExplorer

A volume generated by scanning usually contains one or more objects, and each object comprises one or more components. For example, brain data may contain multiple similar neurons, and each neuron can be further subdivided into an axon and cell body. For any given biomedical volume, we aim to mine the hidden shape and construct a suitable representation to encode its shape information. To achieve this, we design IsoExplorer with three steps to assist in the analysis of 3D shapes hidden in volume data. As shown in Fig.1, we first explore various shapes from the volume by isosurface-driven shape extraction. Then, we design a variational autoencoder model to automatically learn how to generate a latent representation for each shape. Due to self-supervised learning, no costly manual labeling is required for this model training process. Finally, the low-dimensional isosurface representation is computed for isosurface similarity analysis and locality-sensitive hashing is introduced to speed up the shape-retrieval process.

3.1 Isosurface-driven shape extraction

Isosurface generation is an important approach widely used for CT or cypo-EM data analysis, which allows for the generation of anatomical structures from volume data. Given a threshold value, the isosurface generation algorithm will extract the points with equal values from volume data and construct a triangular mesh. However, the generated isosurface may contain multiple objects (e.g., multiple neurons), which do not facilitate individual shape analysis and similar shape analysis between different volume data (e.g., comparing morphological differences between neurons with different functions). Moreover, the optimal structures of different objects may be generated by different thresholds. Therefore, we design the first step of

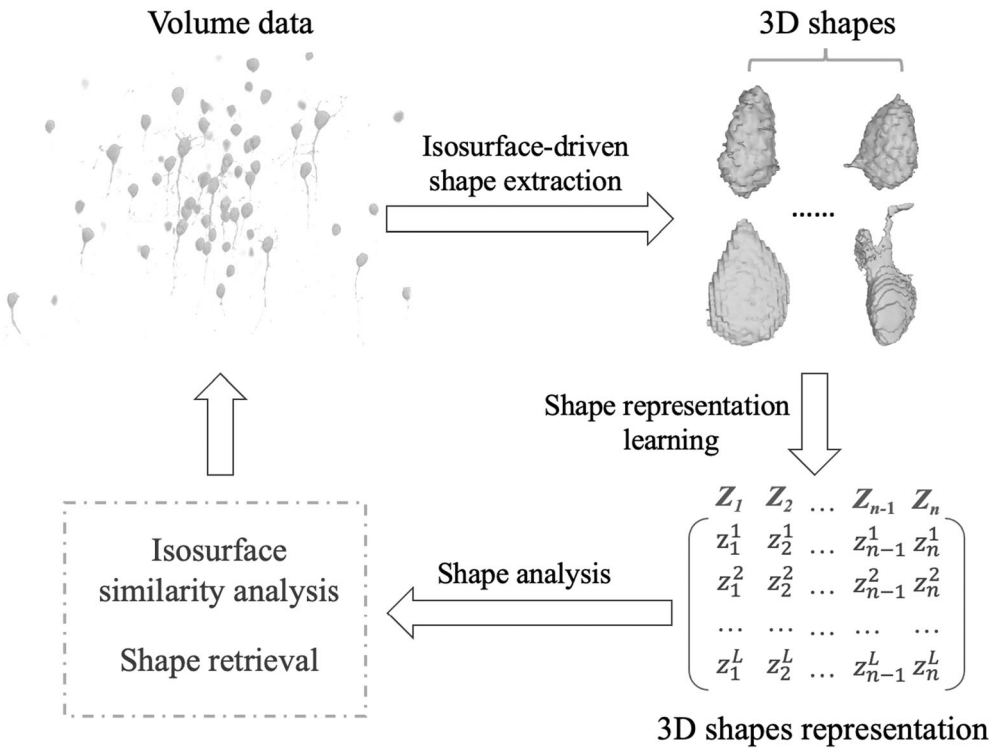

Fig. 1 IsoExplorer workflow consisting of three steps: it first partitions the isosurfaces extracted from the volume data into individual 3D shapes, then creates a latent representation for each shape, and finally leverages the representations to conduct shape analysis

IsoExplorer to extract 3D shapes from volume data by combining isosurface generation and maximum connected component partitioning.

As shown in Fig. 2, given a volume, we first rescale the value of each voxel to [0, 255] and utilize the Marching Cube algorithm (Lorensen and Cline 1987), which is widely used for isosurface generation, to uniformly sample 255 isosurfaces. In addition, in view of the fact that each isosurface may contain more than one object that may contain multiple disconnected components, we could naturally partition each isosurface by its internal connectivity relations. That is, we treat spatially disconnected components of an isosurface as distinct shapes by means of maximum connected component partitioning. For example, the input volume shown in Fig. 2 contains one object, and Isosurface #1 is all internally connected, resulting in a component that allows analysis of the object as a whole, while Isosurface #200 is further divided into four components to facilitate local analysis.

Through this step, we obtain a set of 3D shapes from the volume data, and the space covered by the shapes is usually smaller than the size of the input volume, which mitigates memory consumption. Due to random errors in the scanning process, multiple tiny fragmented shapes may be generated, which affects subsequent analysis. To solve this problem, shapes can be filtered out by setting a threshold for the number of points. In our experiments, depending on the experimental datasets, we filter out shapes that contain less than 1500 points.

3.2 Shape representation learning

3D feature descriptors are the key to shape analysis, they remain invariant to rotations and translations for 3D shapes and insensitive to minor variations. Previously handcrafted feature descriptors usually cannot perform well on multiple datasets and tasks, and in recent years, state-of-the-art results have been achieved by data-driven automatic feature descriptor extraction methods. Xiao et al. (2020) investigated these data-driven approaches and classified them into groups such as voxel-based, image-based, and surface-based. The image-based methods take depth images or a set of images as input and inevitably lose geometric details, while voxel-based methods that take a volumetric grid as input have high computation costs and cannot handle high-resolution data. These methods are not suitable for dealing with 3D shapes obtained from high-resolution volume data. Surface-based methods take points or meshes (discretized forms of 3D shape) as input, which can reduce unnecessary computation while achieving better results. Considering the existence of random noise in the scanning process, the triangular mesh extracted from raw volume may have incorrect connection relations in local regions. Therefore, we turn the 3D shapes into points and express the connection relations between them implicitly in their properties, i.e., the normal vectors of points calculated

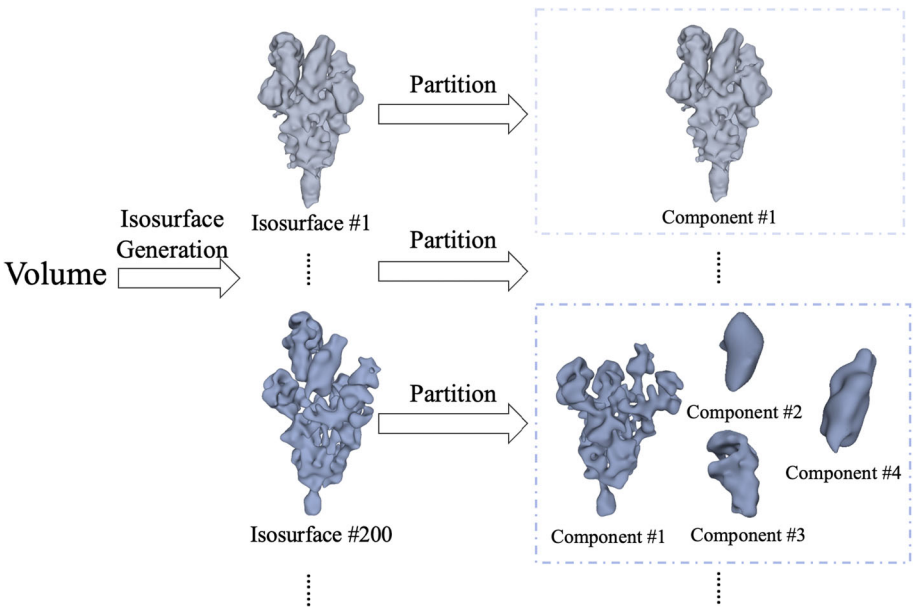

Fig. 2 Isosurface-driven shape extraction. Isosurfaces are generated from a volume, and each isosurface is further partitioned into different components based on the connectivity

from the face normal vectors of surrounding triangles. Concretely, we design the second step of IsoExplorer to automatically learn the latent representation of 3D shapes by using points containing normal vector attributes as the input.

3.2.1 Network architecture

Our design for 3D shape representation learning in IsoExplorer is inspired by the variational autoencoder model (Kingma and Welling 2013). This model can be viewed as a variant of the autoencoder model (Bengio 2009), in that its encoding distribution is regularized during training to ensure that its latent space follows a priori distribution (e.g., Gaussian distribution). While the input in the autoencoder model is encoded to a point in the latent space, the variational autoencoder model encodes the input to a distribution over the latent space, which can handle noise well and prevent overfitting. To enhance the scalability of IsoExplorer, we introduce the O-CNN operation (Wang et al. 2017a) in the network, which improves the efficiency of convolution and reduces memory consumption.

O-CNN operations Traditional 3D CNN operations evolved from CNN operations widely used on images, but cannot handle large volume data well due to the exponential increase in computational cost from pixels to voxels. To make 3D CNN available for high-resolution data, Wang et al. (2017a) proposed O-CNN, which leverages octrees to represent the 3D shapes and designs a series of operations on octrees (e.g., 3D convolution, deconvolution, and pooling). This allows various 3D CNN structures to be efficiently executed on the GPU with limited memory and computation costs.

In O-CNN, the input points are organized by an octree, which divides the input points into eight regions according to their spatial locations. Each node contains a label and a property: the label indicates whether it contains a point or not, and the property is the average normal of its contained points. As the depth of the tree grows, it is divided more finely to describe the model more accurately, which also leads to greater computational and memory consumption. In our experiments, we set the depth of the octree to 6 depending on the experimental datasets.

Network structure As shown in Fig. 3, the variational autoencoder comprises two modules: (1) the encoder takes points organized by an octree as input and encodes it into a distribution on the probabilistic latent space, (2) the decoder takes the latent representation as input and reconstructs the points. We design the convolutional block (Conv Block) and deconvolutional block (Deconv Block) to capture information from the input data. In addition, inspired by He et al. (2016), we incorporate the residual block (Res Block) with skip connection to increase the learning ability of the network and prevent degradation. In all the internal layers, we use batch normalization (BN) (Ioffe and Szegedy 2015) that can reduce internal covariate shift and rectified linear unit (Relu) (Nair and Hinton 2010) as the activation function to avoid gradient vanishing and explosion. The variational autoencoder consists of the Conv block, Deconv block, and Res block.

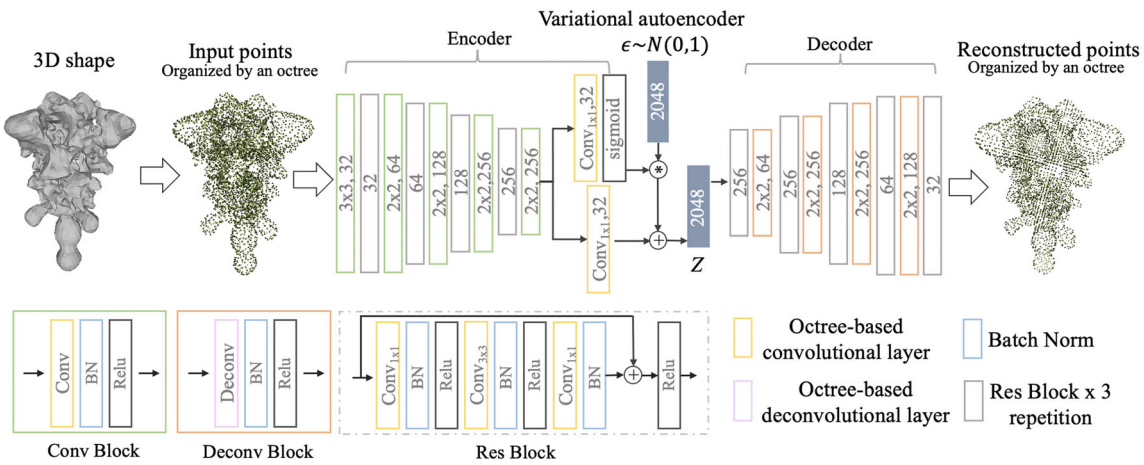

Fig. 3 Architecture of the variational autoencoder model in IsoExplorer. The model takes the points in the 3D shape as input and proceeds through a series of convolution blocks, residual blocks and deconvolution blocks to finally reconstruct the input points

The encoder contains a Conv block with a kernel size of 3, four combinations of a Conv block with a kernel size of 2 and a stride of 2 and a Res block, as well as two O-CNN layers with a kernel size of 1 to output a mean vector and a deviation vector. The mean vector is generated without an activation function, while the deviation vector uses the sigmoid as the activation function. We set the dimensionality L of both vectors to 2048.

With the reparameterization method, we obtain the input needed for the decoder from the mean and deviation vectors of the encoder output. The decoder contains four combinations of a Deconv block, with a kernel size of 2 and a stride of 2, and a Res block, as well as a Res Block to output the reconstructed points organized by an octree.

Specifically, the input 3D shape S is represented by a set of points $\{(x_1, y_1, z_1, nx_1, ny_1, nz_1), \ldots, (x_s, y_s, z_s, nx_s, ny_s, nz_s)\}$, where $(x_i, y_i, z_i, nx_i, ny_i, nz_i)$ stands for the position and normal of the i-th point, and s is the number of points. Then the O-CNN will construct an octree to organize them for the variational autoencoder model. The encoder generates a mean vector $\boldsymbol{M}$ and a deviation vector $\boldsymbol{V}$. By sampling $\boldsymbol{\epsilon}$ from a normal distribution that obeys $N(0, 1)$, we obtain the latent representation $\boldsymbol{Z} = \boldsymbol{M} + \boldsymbol{\epsilon} * \boldsymbol{V}$. Then, the decoder takes this as input and reconstructs the input points organized by an octree.

3.2.2 *Loss functions*

The following two kinds of losses are included in our model.

Reconstruction loss To allow the variational autoencoder to learn correctly how to construct the latent representation of the input, the reconstruction loss is mainly calculated from the position and normal information. For the position information, as the input and reconstructed points are organized by an octree, octree nodes are labeled 1 if they contain points and 0 if they do not. We use softmax loss with two classes at each level of the octree. For the normal information, since the properties of the nodes in the octree are the mean of normals of the contained points, we use regression loss at each level.

*Kullback–Leibler divergence (KLD) loss*Kullback and Leibler (1951). In the variational autoencoder, the encoder is designed to map the input to the latent representation $\boldsymbol{Z}$ and we expect it to obey the standard normal distribution of $N(0, 1)$. To achieve this, we calculate the KLD loss from the mean vector $\boldsymbol{M}$ and the deviation vector $\boldsymbol{V}$ generated by the encoder.

These two types of losses control the model from two aspects. The model is able to generate more accurate reconstruction results when the reconstruction loss has a greater weighting, and to generate a more structured latent space when the KLD loss has a greater weighting. Inspired by Higgins et al. (2017), in our experiment, we set the weight of KLD loss to 0.1.

3.3 3D shape analysis

The learned 3D shape latent representations allow us to easily perform further analysis of shapes. One scenario is users searching the virus database for shapes similar to those obtained from a current sample. In this paper, we focus on isosurface similarity analysis and shape retrieval. Specifically, after the shape representation learning, the 3D shape obtained by extracted from the volume dataset can be represented as a set of latent representations $\{\boldsymbol{Z_1}, \ldots, \boldsymbol{Z_n}\}$, where $\boldsymbol{Z_i}$ is the normalized latent representation of the i-th shape and n is the number of shapes in the volume data.

Isosurface similarity analysis can provide users with an overall understanding of volume data (Tao et al. 2019). For volume data containing multiple complex shapes, directly learning the low-dimensional representation of the entire isosurface requires a strong learning capability of the model, which results in huge computational and memory requirements. Since shapes are obtained by splitting from isosurfaces, the low-dimensional isosurface representation can be naturally obtained by combining the latent representation of decomposed shapes. Because larger shapes should have a greater importance than smaller shapes, we compute the low-dimensional isosurface representation by weighted summation. Then, we adopt cosine similarity, which is commonly used in representation similarity measures, as the similarity between isosurfaces.

Mathematically, the low-dimensional representation $\boldsymbol{I_j}$ of the j-th isosurface can be computed as:

$$\boldsymbol{I_j} = \sum_{i=1}^{N} \frac{\boldsymbol{Z_i^j} * p_i^j}{p_{\max}^j}$$

where $\mathbf{Z}_i^j$ and p_i^j is the latent representation and the number of points of the i-th shapes in the j-th isosurface, and $p_{\max}^j$ is the maximum number of points of all the shapes in the j-th isosurface.

Shape retrieval The key to shape retrieval is to measure the difference between shapes to recommend those that are similar to the retrieval target. We use cosine similarity to calculate the similarity between two shapes. However, shape retrieval in multiple volumes is usually not limited to a single volume dataset, but also involves shape retrieval across volume datasets. As the number of 3D shapes grows from the number of volumes contained in the dataset, it becomes unacceptably computationally expensive to compute similarities for all shapes each time shape retrieval is performed. To speed up the efficiency of shape retrieval and enhance scalability, we further introduce LSH (Leskovec et al. 2020) to help manage the latent representations. It compresses latent representations into signatures and then calculates the hashing values. Similar data will have similar hashing values, which greatly reduces computational effort.

Concretely, for cosine similarity, LSH first divides the space into two regions multiple times, independently, using randomly selected hyperplanes. It then generates the corresponding signatures based on where the latent representation of shapes is located in each division. Then, it hashes signatures into buckets, with similar shapes in the same buckets.

4 Evaluation

In this section, we first introduce the dataset used to evaluate IsoExplorer and the training process of the variational autoencoder, then describe the application of IsoExplorer in isosurface similarity analysis and shape retrieval.

4.1 Dataset and network training

Dataset We use real-world CT scanning, photon microscopy, and cryo-EM datasets to evaluate IsoExplorer. As listed in Table 1, these datasets are CT-Chest, Neurons (Klacansky 2017), SARS-CoV Spike (Kirchdoerfer et al. 2018), MERS-CoV Spike (Park et al. 2019), H-CoV Spike (Park et al. 2019), PD-CoV Spike (Shang et al. 2018), SARS-CoV-2 (Yao et al. 2020), SARS-CoV-2 Spike up (Melero et al. 2020), SARS-CoV-2 Spike down 1 (Yao et al. 2020), and SARS-CoV-2 Spike down 2 (Gobeil et al. 2021). To simulate researchers searching in a shape database, we refer to the shapes extracted from SARS-CoV Spike, MERS-CoV Spike, H-CoV Spike, PD-CoV Spike, and SARS-CoV-2 datasets jointly as CoVs datasets. The last three datasets (SARS-CoV-2 Spike up, SARS-CoV-2 Spike down 1, and SARS-CoV-2 Spike down 2) are used to mimic the new scanning data obtained by searchers in the CoVs dataset. Thus, we generate 10 shapes from each of the last three datasets for evaluation based on sampling around the threshold values recommended by the authors.

Network training The variational autoencoder is implemented with TensorFlow, and the training process is performed on the CT-Chest, Neurons, and CoVs datasets. We divide the first two datasets into training and testing sets by a ratio of 7:3, while for the CoVs dataset, we first divide the sub-datasets it contains by the same ratio, and then combine them into training and testing sets. We set the size of each batch to 16 and use the stochastic gradient descent method. To avoid overfitting, for each input shape, we perform a random rotation for data augmentation. In addition, we initialize the learning rate to 0.1 and apply a weight decay of

Table 1 Dataset overview

Dataset	Dimension	#Shapes
CT-Chest	384 × 384 × 240	3164
Neurons (Klacansky 2017)	1024 × 1024 × 314	8948
SARS-CoV Spike (Kirchdoerfer et al. 2018)	360 × 360 × 360	1171
MERS-CoV Spike (Park et al. 2019)	400 × 400 × 400	5159
H-CoV Spike (Park et al. 2019)	280 × 280 × 280	3149
PD-CoV Spike (Shang et al. 2018)	256 × 256 × 256	2868
SARS-CoV-2 (Yao et al. 2020)	512 × 512 × 512	3071
SARS-CoV-2 Spike up (Melero et al. 2020)	432 × 432 × 432	10
SARS-CoV-2 Spike down 1 (Yao et al. 2020)	256 × 256 × 256	10
SARS-CoV-2 Spike down 2 (Gobeil et al. 2021)	300 × 300 × 300	10

0.005 to avoid overfitting. The network is trained on a NVIDIA 2080 Ti GPU with 50 epochs, each taking about 15 min.

The reconstruction accuracies of the CT-Chest, Neurons and CoVs datasets on the training set are 0.87, 0.93 and 0.88, respectively, demonstrating that the variational autoencoder has the capability to learn a suitable representation for shapes, while their reconstruction accuracies on the testing set are 0.85, 0.91 and 0.87, respectively, which indicates that the variational autoencoder is not overfitting. This shows that the proposed model is able to capture the position and normal information of the input shape well and generate appropriate latent representations for them, which is the basis for the subsequent analysis.

4.2 Application I: Isosurface similarity analysis

Similarity analysis of isosurfaces can help users uncover clustering patterns between isosurfaces and select appropriate threshold values. We demonstrate the application of IsoExplorer for isosurface analysis with the CT-Chest dataset. The dataset was generated by CT scanning of the human chest, as shown in Fig. 4a, which illustrates that the human chest is principally composed of the skin, lungs, ribs, scapulae, and vertebrae.

By computing the similarity between pairs of different isosurfaces, we obtain a similarity matrix. In the heat map shown in Fig. 4b, the darker color represents the higher degree of similarity, and one can see that there are four main clusters represented in all the isosurfaces. In conjunction with the image rendered by the ray-casting algorithm, it is evident that the color of the skin and the lungs are similar, in red. This corresponds to the cluster formed by the isosurface, with a threshold less than 100, and the cluster could be further divided into two highly similar internal sub-clusters with a threshold of 45, which corresponds to the skin and lungs. Similarly, the ribs, spine, and scapulae included in the CT-Chest dataset are captured by other clusters, whose correspondence is displayed in Fig. 4. These findings indicate that the low-dimensional isosurface representation computed from the latent representations of shapes correctly encodes the information it contains, and could provide users with a general overview of the data and aid in the design of transfer functions.

4.3 Application II: Shape retrieval

Shape retrieval enables users to find similar shapes to the target of interest in volume data, which mainly involves retrieval between the same or different volume data.

4.3.1 Shape retrieval within the same volume data

Shape retrieval within an individual data is effective for finding shapes similar to the desired ones within the current dataset, and the search process incidentally provides analysis of morphology and occurrence

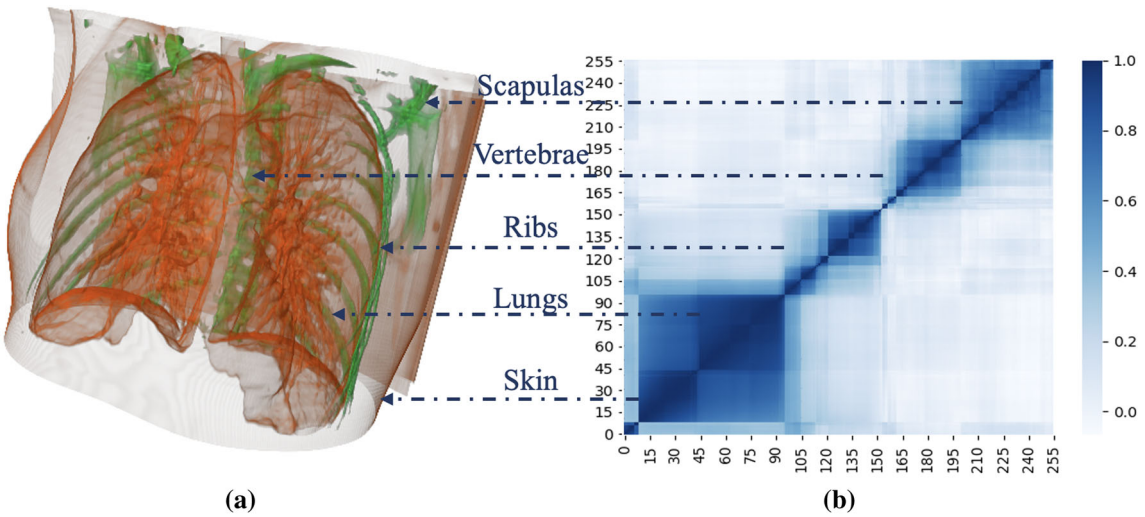

Fig. 4 Isosurface similarity analysis of the CT-Chest dataset. **a** The raw data visualized by the ray-casting algorithm. **b** Isosurface similarity matrix visualized by the heat map

patterns. In this case, we perform shape retrieval within the Neurons dataset, which contains a series of marmoset primary visual cortical pyramidal neurons labeled by green fluorescent protein, imaged under photon microscopy.

In Fig. 5a, the shape marked in red is the shape of interest (i.e., the target shape), and we list the three shapes that are most similar to it in descending order of similarity. All the retrieval results are similar to the target shape and are scattered on the isosurfaces with isovalues of 100, 136, and 176. This is a common occurrence in the analysis of isosurfaces in volume data, i.e., each shape contained in the volume data may have its own unique optimal isosurface threshold. Furthermore, Fig. 5b visualizes the entire dataset and plots the position of these shapes. This reflects the characteristic of this dataset that neurons are all composed of neuronal bodies and axons, which are mostly similar, and different shapes of neurons are distinguished by their body size and axon length. Using shape retrieval in the same volume data, researchers can automatically find results similar to the target shape from different isosurfaces, instead of manually traversing the isosurfaces.

4.3.2 Shape retrieval between volume data

Shape retrieval between volume data can help users discover similar shapes embedded in various volume data to gain insights. We use the CoVs dataset that contains data on various coronaviruses as a candidate set, and then use the shapes generated from the SARS-CoV-2 Spike up and two SARS-CoV-2 Spike down

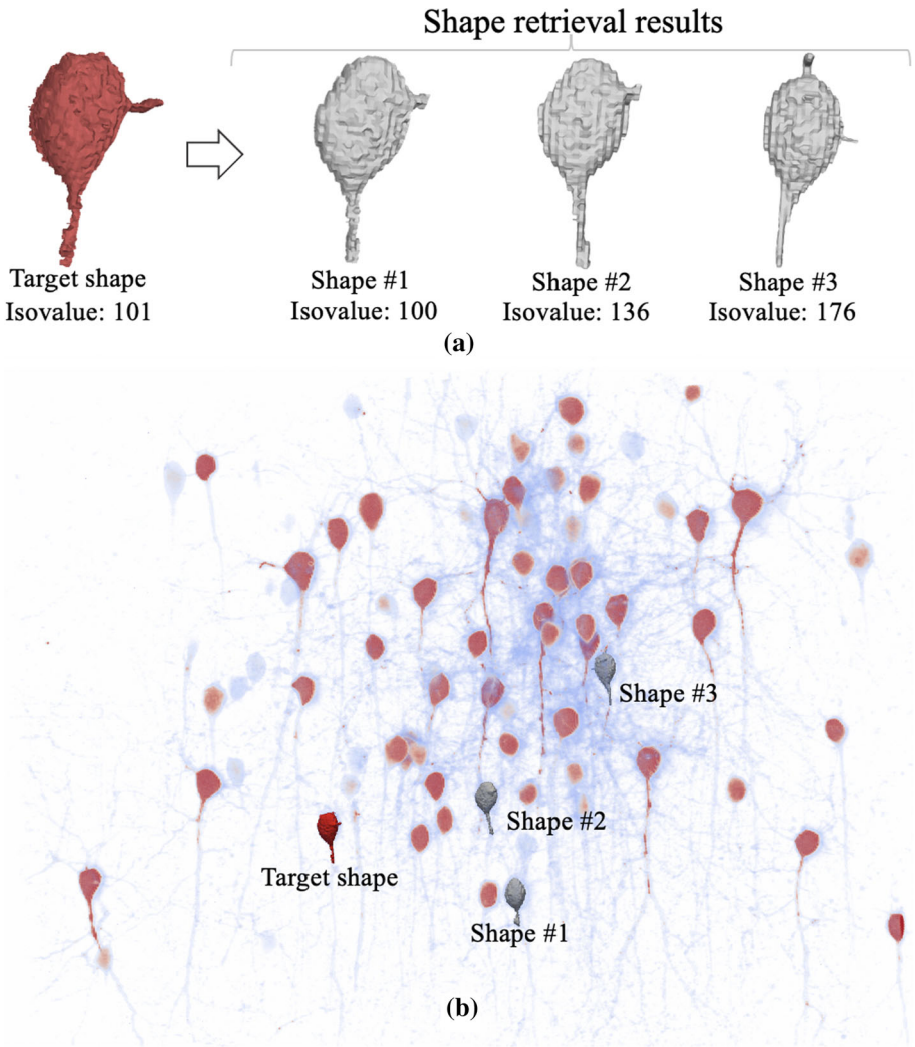

Fig. 5 Shape retrieval in the Neurons dataset. **a** The target shape with the retrieval results and **b** their positions in the dataset

datasets as targets to retrieve similar shapes from the candidate set. All datasets were obtained by cryo-EM scanning. SARS-CoV-2 is a novel coronavirus that resulted in a global pandemic beginning in 2020. As shown in Fig. 6a, its molecular architecture consists of four main components, namely spikes, a membrane, an envelope and nucleocapsid proteins (Yao et al. 2020). The spikes play an important role in membrane fusion, and in the prefusion state, they can be divided into up and down conformations depending on the orientation. Note that the target shapes extracted from the SARS-CoV-2 Spike up and two SARS-CoV-2 Spike down datasets does not appear in the training or testing process of the model.

For each of these three datasets, we generate a series of target shapes by sampling 10 isosurfaces around the recommended values, and Fig. 6b presents the retrieval results generated by our method and by FlowNet (Han et al. 2018) for the target shapes generated with the recommended values. FlowNet represents the input shape as a binary volume, and subsequently uses a 3D CNN-based autoencoder to construct a representation vector of dimension 1024. Due to limited GPU memory, we downsample the input shape and generate a volume with a size of 51^3 as the input to FlowNet. By comparing the Top-1 search results, we find that both IsoExplorer and FlowNet could correctly find similar target shapes from the CoVs dataset. However, when comparing the Top-3 search results, FlowNet seems to confuse the two conformations of the spike of SARS-CoV-2 as well as the spikes of other coronaviruses. This is mainly due to the need for downsampling, which possibly ignores the local differences that occur in volume data and limits the representational capabilities of the FlowNet. In addition, we perform a quantitative evaluation based on the labels of the retrieved results and the target shape. In these three datasets, the averaged Top-5 accuracy of IsoExplorer and FlowNet are 84% and 57%, respectively. Compared to traditional knowledge-driven 3D descriptors, which focus on specific tasks (e.g., contour tree for symmetry analysis Thomas and Natarajan 2011, 3D SIFT for pattern matching Wang et al. 2015), data-driven 3D descriptors are more generic for different tasks and require no or little knowledge about data (Rostami et al. 2019). During retrieval between different volume data, users can establish connections between shapes scanned at different resolutions and search the shape database for target shapes as keywords to find similar shapes.

In addition, during shape retrieval, the growth in the number of shapes will increase the retrieval time because of the increase in the size and number of volume data. In contrast to linear search, LSH has a

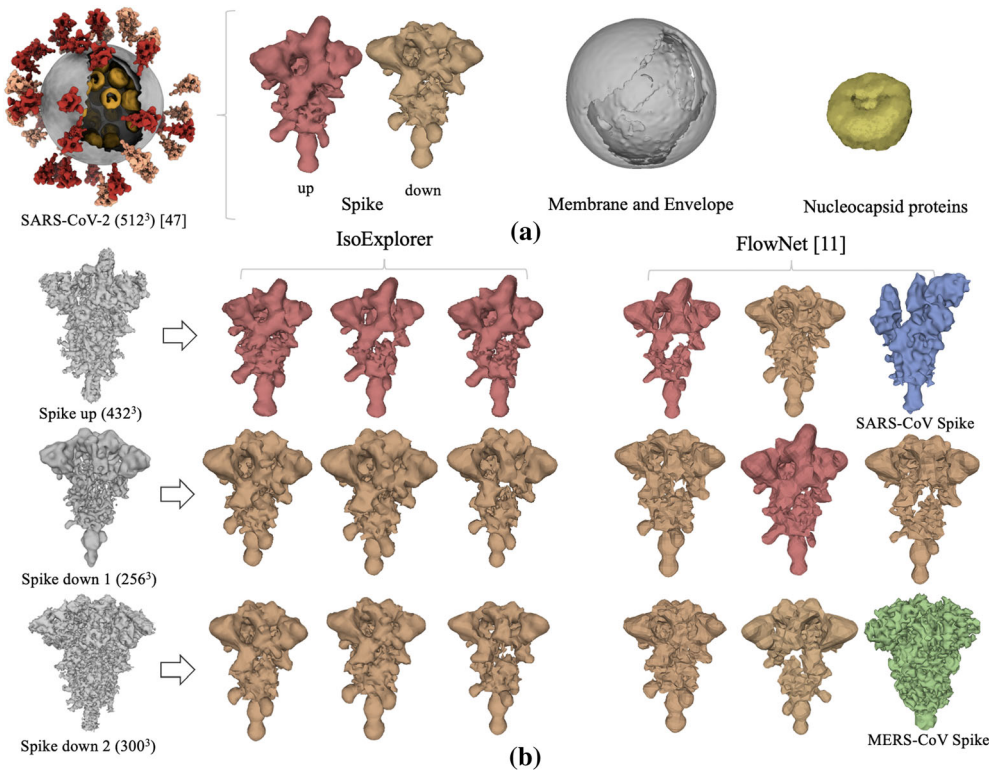

Fig. 6 Shape retrieval the CoVs dataset. **a** The molecular architecture of SARA-CoV-2. **b** The target shapes and corresponding retrieval results of IsoExplorer and FlowNet

Table 2 Comparison of per query time for linear search and LSH

#Retrieved shapes	LSH time (sec)		Linear search time (sec)	
	CoVs	Neurons	CoVs	Neurons
1	0.013	0.0001	0.624	0.352
5	0.013	0.0001	0.613	0.360
10	0.013	0.0001	0.611	0.357
20	0.014	0.0001	0.613	0.352
40	0.014	0.0001	0.613	0.348
80	0.014	0.0001	0.612	0.353
160	0.014	0.0001	0.601	0.350

construction step before initial use, which takes 1.73s and 0.22s on the Neurons and CoVs datasets, respectively. Table 2 illustrates the average time needed for 100 queries using LSH and linear search for different numbers of retrieved shapes in the Neurons and CoVs datasets. We can find that the construction time for LSH is only comparable to the time needed for a few linear searches, and after construction is completed, LSH takes very little time compared to a linear search that requires scanning all the data each time. This permits fast retrieval in a database containing numerous shapes.

4.4 Discussion

Through isosurface-driven shape extraction, we obtain various shapes hidden in volume datasets, which are then captured by a variational autoencoder and encoded into latent representation for subsequent shape analysis. We validate the effectiveness and usefulness of IsoExplorer by performing isosurface similarity analysis and shape retrieval on a real dataset. With isosurface similarity analysis, users can obtain an overview of volume data. With shape retrieval, users can construct connections between different isosurfaces of the same resolution, or different volume data with different resolutions to find similar shapes. The scalability, generalizability, and limitations of IsoExplorer are discussed below.

Scalability CNNs on volumes generally have a large number of parameters due to the dense convolution operation. Thus, previous CNN-based methods cannot take the original volumes as the input due to limited GPU memory. They have to downsample the high-resolution volumes into low-resolution volumes or crop parts of the volumes for the training dataset. For example, FlowNet can only process volumes smaller than 64^3. Benefiting from the maximum connected component partitioning and the sparse convolution operation of O-CNN, the proposed IsoExplorer can directly handle the original volumes to generate the representations for connected components of each isosurface. As a result, the scalability of IsoExplorer on volumes is significantly enhanced, depending on the vertex number of connected components. In our experiments, the largest connected component contains 1,354k points in $384 \times 384 \times 240$ and consumes about 10.5GB of GPU memory at a batch size of 4. In addition, geometry simplification techniques can be used to reduce the number of points in connected components to further improve the scalability of IsoExplorer. From the data scale perspective, the number of shapes used in our evaluation ranges from 3k to 15k, and the variational autoencoder does not overfit under random rotation as data augmentation. For data containing few shapes, less layers of variational autoencoder may be needed to reduce its complexity to prevent overfitting. Meanwhile, we plan to further explore capturing more data from EMDB for training to simulate more realistic research scenarios in the future.

Generalizability IsoExplorer based on self-supervised learning does not require additional labeling and has been demonstrated on biomedical data via isosurface similarity analysis and shape retrieval. Domain experts are usually interested in the boundaries of features in biomedical data, such as bones and cell walls. These boundaries are represented well by the isosurfaces, which can be reconstructed from biomedical data. For volumes in other domains, if the isosurface has a clear semantic meaning and a relatively complete isosurface can be reconstructed from the volume, IsoExplorer can be also used to learn semantic representations of connected components for feature analysis. In addition, for analytical tasks that are sensitive to the relative position between individual components, it is also possible to learn low-dimensional representations of isosurfaces without information loss by using the entire isosurface as input to the variational autoencoder model. Note that this may significantly increase the size of the input and the required GPU memory.

Limitations In the shape extraction stage, we apply uniform sampling to generate isosurfaces, which may not have adequately accounted for certain intervals in the dataset. For data with known parameters, prior

knowledge can be incorporated during sampling of the isosurface to increase the number of shapes of interest. Also, numerous tiny shapes may be generated in the noisy volume data, which increases the difficulty of extracting the complete isosurfaces. In our implementation, we filter out shapes containing less than 1500 points, and this is a hyperparameter that may cause shapes of interest to domain experts to be incorrectly screened out. Thus, preprocessing operations (e.g., smoothing) are necessary when analyzing noisy volumes. In addition, in the shape representation learning stage, we encode their connectivity relations implicitly in the node normal vectors at input, rather than using the connectivity relations directly, an approach which may introduce information loss and thus desensitize the model to tiny local changes. This could be corrected by using a mesh-based model to consider the connectivity relationship.

5 Conclusion and future work

In this paper, we present IsoExplorer to help analyzing features of biomedical volume data by their shapes. It partitions the isosurfaces generated from volume data into different shapes according to the connection relationship and then encodes their geometric information with a variational autoencoder model for subsequent analysis. IsoExplorer uses LSH to store the latent representations of shapes, providing users with a quick way to retrieve them. We verify the effectiveness and usefulness of IsoExplorer by isosurface similarity analysis and similar shape retrieval on real-world datasets and by comparison with other methods.

For future work, we plan to address the limitations of the current IsoExplorer and expand its application in volume data shape analysis.

Acknowledgements This work was supported by the National Key Research & Development Program of China (2017YFB0202203), National Natural Science Foundation of China (61890954 and 61972343), and Key Research & Development Program of Zhejiang Province (2021C03032).

References

Bai S, Bai X, Zhou Z, Zhang Z, Tian Q, Latecki LJ (2017) Gift: towards scalable 3d shape retrieval. IEEE Trans Multim 19(6):1257–1271

Bengio Y (2009) Learning deep architectures for ai. Found Trends Mach Learn 2(1):1–127

Berger M, Li J, Levine JA (2018) A generative model for volume rendering. IEEE Trans Vis Comput Graph 25(4):1636–1650

Bruckner S, Möller T (2010) Isosurface similarity maps. Comput Graph Forum 29(3):773–782

Castillo-Barnes D, Martinez-Murcia FJ, Ortiz A, Salas-Gonzalez D, RamÍrez J, Górriz JM (2020) Morphological characterization of functional brain imaging by isosurface analysis in parkinson's disease. Int J Neural Syst 30(9):2050044

Cheng HC, Cardone A, Jain S, Krokos E, Narayan K, Subramaniam S, Varshney A (2018) Deep-learning-assisted volume visualization. IEEE Trans Vis Comput Graph 25(2):1378–1391

El-Baz A, Nitzken M, Khalifa F, Elnakib A, Gimel'farb G, Falk R, El-Ghar MA (2011) 3d shape analysis for early diagnosis of malignant lung nodules. In: Information processing in medical imaging, pp 772–783

Engel D, Ropinski T (2021) Deep volumetric ambient occlusion. IEEE Trans Vis Comput Graph 27(2):1268–1278

Gao L, Lai YK, Liang D, Chen SY, Xia S (2016) Efficient and flexible deformation representation for data-driven surface modeling. ACM Trans Graph 35(5):1–17

Gobeil S, Janowska K, McDowell S, Mansouri K, Parks R, Stalls V, Kopp MF, Manne K, Saunders KO, Edwards RJ et al (2021) Effect of natural mutations of sars-cov-2 on spike structure, conformation and antigenicity. bioRxiv

Han J, Tao J, Wang C (2018) Flownet: a deep learning framework for clustering and selection of streamlines and stream surfaces. IEEE Trans Vis Comput Graph 26(4):1732–1744

Han Z, Liu Z, Han J, Vong CM, Bu S, Chen CLP (2019) Unsupervised learning of 3-d local features from raw voxels based on a novel permutation voxelization strategy. IEEE Trans Cybern 49(2):481–494

Han J, Zheng H, Xing Y, Chen DZ, Wang C (2021) V2v: a deep learning approach to variable-to-variable selection and translation for multivariate time-varying data. IEEE Trans Vis Comput Graph 27(2):1290–1300

He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: Proceedings of the European conference on computer vision (ECCV), pp 630–645

He W, Wang J, Guo H, Wang KC, Shen HW, Raj M, Nashed YS, Peterka T (2019) Insitunet: deep image synthesis for parameter space exploration of ensemble simulations. IEEE Trans Vis Comput Graph 26(1):23–33

Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-vae: learning basic visual concepts with a constrained variational framework. In: International conference on learning representations

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456

Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114

Kirchdoerfer RN, Wang N, Pallesen J, Wrapp D, Turner HL, Cottrell CA, Corbett KS, Graham BS, McLellan JS, Ward AB (2018) Stabilized coronavirus spikes are resistant to conformational changes induced by receptor recognition or proteolysis. Sci Rep 8(1):1–11

Klacansky P (2017) Open scientific visualization datasets. https://klacansky.com/open-scivis-datasets/

Kruger J, Westermann R (2003) Acceleration techniques for gpu-based volume rendering. In: IEEE visualization, pp 287–292
Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
Lawson CL, Patwardhan A, Baker ML, Hryc C, Garcia ES, Hudson BP, Lagerstedt I, Ludtke SJ, Pintilie G, Sala R et al (2016) Emdatabank unified data resource for 3dem. Nucleic Acids Res 44(D1):D396–D403
Leskovec J, Rajaraman A, Ullman JD (2020) Mining of massive data sets. Cambridge University Press, Cambridge
Li Y, Wang C, Shene CK (2014) Streamline similarity analysis using bag-of-features. In: Visualization and data analysis, vol 9017, p 90170N
Liu J, Gao Y, Shan G, Chi X (2019) Vasem: visual analytics system for electron microscopy data bank. J Vis 22(6):1145–1159
Loncaric S (1998) A survey of shape analysis techniques. Patt Recogn 31(8):983–1001
Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3d surface construction algorithm. ACM Siggraph Comput Graph 21(4):163–169
Melero R, Sorzano COS, Foster B, Vilas JL, Martínez M, Marabini R, Ramírez-Aportela E, Sanchez-Garcia R, Herreros D, Del Caño L et al (2020) Continuous flexibility analysis of sars-cov-2 spike prefusion structures. IUCrJ 7(6)
Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: International conference on machine learning, pp 807–814
Park YJ, Walls AC, Wang Z, Sauer MM, Li W, Tortorici MA, Bosch BJ, DiMaio F, Veesler D (2019) Structures of mers-cov spike glycoprotein in complex with sialoside attachment receptors. Nat Struct Mol Biol 26(12):1151–1157
Porter WP, Xing Y, von Ohlen BR, Han J, Wang C (2019) A deep learning approach to selecting representative time steps for time-varying multivariate data. In: IEEE visualization conference (VIS), pp 1–5
Rostami R, Bashiri FS, Rostami B, Yu Z (2019) A survey on data-driven 3d shape descriptors. Comput Graph Forum 38(1):356–393
Shang J, Zheng Y, Yang Y, Liu C, Geng Q, Tai W, Du L, Zhou Y, Zhang W, Li F (2018) Cryo-electron microscopy structure of porcine deltacoronavirus spike protein in the prefusion state. J Virol 92(4)
Shinagawa Y, Kunii TL, Kergosien YL (1991) Surface coding based on morse theory. IEEE Comput Graph Appl 11(5):66–78
Tabia H, Laga H, Picard D, Gosselin PH (2014) Covariance descriptors for 3d shape matching and retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
Tao J, Imre M, Wang C, Chawla NV, Guo H, Sever G, Kim SH (2019) Exploring time-varying multivariate volume data using matrix of isosurface similarity maps. IEEE Trans Vis Comput Graph 25(1)
Thomas DM, Natarajan V (2011) Symmetry in scalar field topology. IEEE Trans Vis Comput Graph 17(12):2035–2044
Tkachev G, Frey S, Ertl T (2021) Local prediction models for spatiotemporal volume visualization. IEEE Trans Vis Comput Graph 27(7):3091–3108
Wang PS, Liu Y, Guo YX, Sun CY, Tong X (2017) O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Trans Graph 36(4):1–11
Wang Z, Esturo JM, Seidel HP, Weinkauf T (2017) Stream line-based pattern search in flows. Comput Graph Forum 36(8):7–18
Wang Z, Seidel HP, Weinkauf T (2015) Multi-field pattern matching based on sparse feature sampling. IEEE Trans Vis Comput Graph 22(1):807–816
Weiss S, Chu M, Thuerey N, Westermann R (2019) Volumetric isosurface rendering with deep learning-based super-resolution. arXiv preprint arXiv:1906.06520
Xiao YP, Lai YK, Zhang FL, Li C, Gao L (2020) A survey on deep geometry learning: from a representation perspective. Comput Vis Media 6(2):113–133
Xie J, Fang Y, Zhu F, Wong E (2015) Deepshape: deep learned shape descriptor for 3d shape matching and retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1275–1283
Yang C, Li Y, Liu C, Yuan X (2019) Deep learning-based viewpoint recommendation in volume visualization. J Vis 22(5):991–1003
Yao H, Song Y, Chen Y, Wu N, Xu J, Sun C, Zhang J, Weng T, Zhang Z, Wu Z, Cheng L, Shi D, Lu X, Lei J, Crispin M, Shi Y, Li L, Li S (2020) Molecular architecture of the sars-cov-2 virus. Cell **183**(3), 730–738.e13. URL https://www.sciencedirect.com/science/article/pii/S0092867420311594
Yip KM, Fischer N, Paknia E, Chari A, Stark H (2020) Atomic-resolution protein structure determination by cryo-em. Nature 587(7832):157–161
Zaharescu A, Boyer E, Varanasi K, Horaud R (2009) Surface feature detection and description with applications to mesh matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 373–380