Computational Visual Media
https://doi.org/10.1007/s41095-020-0174-8 Vol. 6, No. 2, June 2020, 113–133
Review Article

A survey on deep geometry learning: From a representation perspective

Yun-Peng Xiao[1], Yu-Kun Lai[2], Fang-Lue Zhang[3], Chunpeng Li[1], Lin Gao[1] (✉)

Abstract Researchers have achieved great success in dealing with 2D images using deep learning. In recent years, 3D computer vision and geometry deep learning have gained ever more attention. Many advanced techniques for 3D shapes have been proposed for different applications. Unlike 2D images, which can be uniformly represented by a regular grid of pixels, 3D shapes have various representations, such as depth images, multi-view images, voxels, point clouds, meshes, implicit surfaces, etc. The performance achieved in different applications largely depends on the representation used, and there is no unique representation that works well for all applications. Therefore, in this survey, we review recent developments in deep learning for 3D geometry from a representation perspective, summarizing the advantages and disadvantages of different representations for different applications. We also present existing datasets in these representations and further discuss future research directions.

Keywords 3D shape representation; geometry learning; neural networks; computer graphics

1 Introduction

1.1 Background

Recent improvements in methods for acquisition and rendering of 3D models have resulted in consolidated repositories on the Internet containing huge numbers of 3D shapes. With the increased availability of 3D models, we have been seeing an explosion in the demands of processing, generation, and visualization of 3D models in a variety of disciplines, such as medicine, architecture, and entertainment. Techniques for matching, identification, and manipulation of 3D shapes have become fundamental building blocks in modern computer vision and computer graphics systems. Due to the complexity and irregularity of 3D shape data, effectively representing 3D shapes remains a challenging problem. Thus, there have been extensive research efforts concentrating on how to deal with and generate 3D shapes in different representations.

In early research on 3D shape representations, 3D objects were normally modeled with a global approach, such as constructive solid geometry and deformed superquadrics. Those approaches have several drawbacks when utilized for tasks like recognition and retrieval. Firstly, when representing imperfect 3D shapes, including those with noise and incompleteness, which are common in practice, such representations may have a negative influence on matching performance. Secondly, the high-dimensionality heavily burdens the computation and tends to make models overfit. Hence, more sophisticated methods are designed to extract representations of 3D shapes in a more concise, yet discriminative and informative form.

Several related surveys have been published [1–3], which focus on different aspects of deep learning for 3D geometry. Moreover, with rapid development of 3D shape representations and related techniques for deep learning, it is essential to further summarize up-to-date research. In this survey, we mainly review deep learning methods on 3D shape representations

1 Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. E-mail: Y.-P. Xiao, xiaoypgk@gmail.com; C. Li, cpli@ict.ac.cn; L. Gao, gaolin@ict.ac.cn (✉).
2 School of Computer Science and Informatics, Cardiff University, Wales, UK. E-mail: LaiY4@cardiff.ac.uk.
3 School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. E-mail: fanglue.zhang@ecs.vuw.ac.nz.
Manuscript received: 2020-02-16; accepted: 2020-04-17

and discuss their advantages and disadvantages in different application scenarios. We now give a brief summary of different 3D shape representation categories.

1.2 Depth and multi-view images

Depth and multi-view images can be used to represent 3D models over a 2D field; the regular structure of images makes for efficient processing. Depending on whether depth is included, 3D shapes can be represented by RGB (color) or RGB-D (color and depth) images viewed from different viewpoints. Because of the influx of available depth data due to the popularity of 2.5D sensors, such as Microsoft Kinect, Intel RealSense, etc., multi-view RGB-D images are widely used to represent real-world 3D shapes. Large numbers of image-based models are available in this representation, but it is inevitable that such representations lose some geometric detail.

1.3 Voxels

A voxel is a 3D extension of the concept of pixel. Like pixels in 2D, the voxel-based representation also has a regular structure in 3D space. Architectures of various neural networks which have proved useful in the 2D image field [4, 5] can be easily extended to voxel form. Nevertheless, adding one dimension means an exponential increase in data size. As resolution increases, the memory required and computational costs increase dramatically, which restricts the representation to low resolutions when representing 3D shapes.

1.4 Surfaces

Surface-based representations describe 3D shapes by encoding their surfaces, which can also be regarded as 2-manifolds. Point clouds and meshes are both discretized forms of 3D shape surfaces. Point clouds use a set of sampled 3D point coordinates to represent the surface. They can easily be generated by scanners but are difficult to process due to their lack of order and connectivity information. Researchers use order invariant operators such as the max pooling operator in deep neural networks [6, 7] to mitigate the lack of order. Meshes can depict higher quality 3D shapes with less memory and computational cost compared to point clouds and voxels. A mesh contains a vertex set and an edge set. Due to its graphical nature, researchers have made attempts to build graph-based convolutional neural networks for coping with meshes. Some other methods regard meshes as the discretization of 2-manifolds. Moreover, meshes are more suitable for 3D shape deformation. One can deform a mesh model by transforming vertices while simultaneously retaining the connectivity.

1.5 Implicit surfaces

Implicit surface representation exploits implicit field functions, such as occupancy functions [8] and signed distance functions [9], to describe the surface of 3D shapes. The implicit functions learned by deep neural networks define the spatial relationship between points and surfaces. They provide a description with infinite resolution for 3D shapes with reasonable memory consumption, and are capable of representing shapes with changing topology. Nevertheless, implicit representations cannot reflect the geometric features of 3D shapes directly, and usually need to be transformed to explicit representations such as meshes. Most methods apply iso-surfacing, such as marching cubes [10], which is an expensive operation.

1.6 Structured representation

One way to cope with complex 3D shapes is to decompose them into structure and geometric details, leading to structured representations. Recently, increasing numbers of methods regard a 3D shape as a collection of parts and organize them linearly or hierarchically. The structure of 3D shapes is processed by *recurrent neural networks* (*RNNs*) [11], *recursive neural networks* (*RvNNs*) [12], or other network architectures. Each part of the shape can be processed by unstructured models. The structured representation focuses on the relations (such as symmetry, supporting, being supported, etc.) between different parts within a 3D shape, which provides a better descriptive capability than alternative representations.

1.7 Deformation-based representation

As well as rigid man-made 3D shapes such as chairs and tables, there are also a large number of non-rigid (e.g., articulated) 3D shapes such as human bodies, which also play an important role in computer animation, augmented reality, etc. Deformation-based representation is used mainly to describe intrinsic deformation properties while ignoring extrinsic transformation properties. Many methods use rotation-invariant local features for

describing shape deformation to reduce distortion while retaining geometric details.

1.8 Geometry learning

Recently, deep learning has achieved superior performance to classical methods in many fields, including 3D shape analysis, reconstruction, etc. A variety of architectures of deep networks have been designed to process or generate 3D shape representations, which we refer to as *geometry learning*. In the following sections, we focus on the most recent deep learning based methods for representing and processing 3D shapes in different forms. Based on how the representation is encoded and stored, our survey is organized around the following structure: Section 2 reviews image-based shape representation methods. Sections 3 and 4 introduce voxel- and surface-based representations respectively. Section 5 further introduces implicit surface representations. Sections 6 and 7 review structure- and deformation-based description methods. We then summarize typical datasets in Section 8 and typical applications for shape analysis and reconstruction in Section 9, before concluding the paper in Section 10. Figure 1 provides a timeline of representative deep learning methods based on various 3D shape representations.

2 Image-based representations

2D images are projections of 3D entities. Although the geometric information carried by one image is incomplete, a plausible 3D shape can be inferred from a set of images with different perspectives. The extra channel of depth in RGB-D data further enhances the capacity of image-based representations to encode geometric cues. Benefiting from the image-like structure, research using deep neural networks for 3D shape inference from images started earlier than alternative representations that explicitly depict the surface or geometry of 3D shapes.

Socher et al. [33] proposed a convolutional and recursive neural network for 3D object recognition, which copes with RGB and depth images using single convolutional layers separately and merges the features with a recursive network. Eigen et al. [16] first proposed reconstructing a depth map from a single RGB image and designed a new scale invariant loss for the training stage. Gupta et al. [34] encoded the depth map into three channels including disparity, height, and angle. Other deep learning methods based on RGB-D images designed for 3D object detection [35, 36] outperform previous methods.

Images from different viewpoints can provide complementary cues to infer 3D objects. Thanks to the development of 2D deep learning models, learning methods based on multi-view image representation perform better for 3D shape recognition than those based on other 3D representations. Su et al. [14] proposed *MVCNN* (multi-view convolutional neural network) for 3D object recognition. It processes the images for different views separately in the first part of the CNN, then aggregates the features extracted from different views by view-pooling layers, and finally sends the merged features to the remainder of the CNN. Qi et al. [37] proposed adding a multi-resolution strategy to MVCNN for higher classification accuracy.

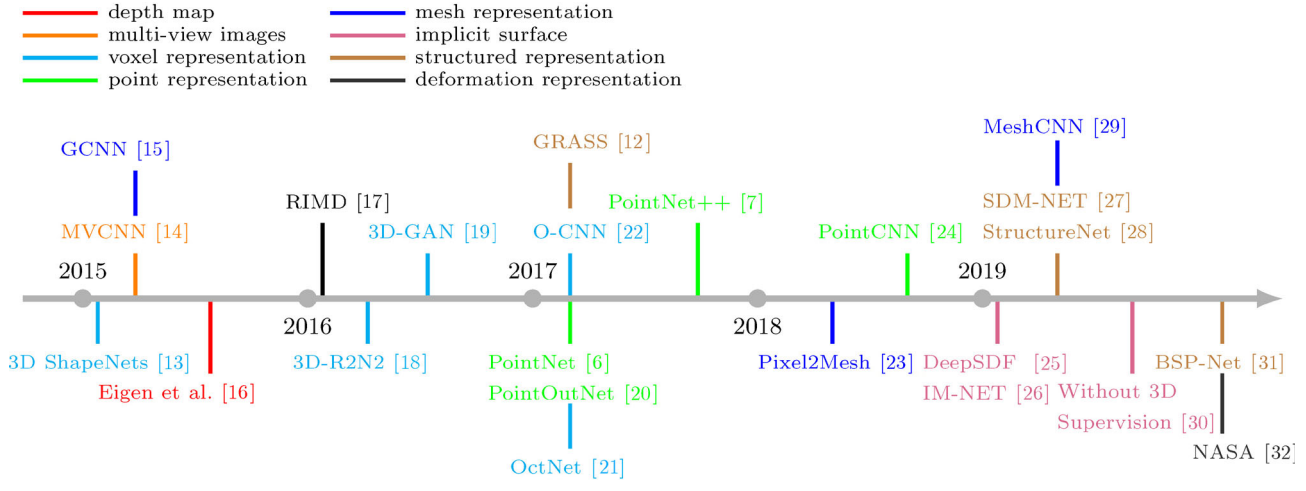


Fig. 1 The timeline of deep learning based methods for various 3D shape representations.

3 Voxel-based representations

3.1 Dense voxel representation

The voxel-based representation is traditionally a dense representation, which describes 3D shape data by a volumetric grid in 3D space. Each voxel in a cuboid grid records occupancy status (i.e., occupied or unoccupied).

One of the earliest methods to apply deep neural networks to volumetric representations, *3D ShapeNets*, was proposed by Wu et al. [13] in 2015. They assigned three different states to the voxels in the volumetric representation produced by 2.5D depth maps: observed, unobserved, and free. 3D ShapeNets extended the deep belief network (DBN) [38] from pixel data to voxel data and replaced fully connected layers in the DBN with convolutional layers. The model takes the aforementioned volumetric representation as input, and outputs category labels and predicted 3D shape by iterative computations. Concurrently, Maturana et al. proposed processing a volumetric representation with 3D convolutional neural networks (3D CNNs) [39] and designed *VoxNet* [40] for object recognition. VoxNet defines several volumetric layers, including an input layer, convolutional layers, pooling layers, and fully connected layers. Although these layers simply extend traditional 2D CNNs [4] to 3D, VoxNet is easy to implement and train, and gets promising performance as the first attempt at volumetric convolution. In addition, to ensure that VoxNet is invariant to orientation, Maturana et al. augmented the input data by rotating each shape into n instances with different orientations during training, and added a pooling operation after the output layer to group all predictions from the n instances during testing.

In addition to the development of deep belief networks and convolutional neural networks for shape analysis based on volumetric representation, two most successful generative models, namely auto-encoders and generative adversarial networks (GANs) [41] have also been extended to support this representation. Inspired by denoising auto-encoders (DAEs) [42, 43], Sharma et al. [44] proposed an autoencoder model *VConv-DAE* to cope with voxels. It is one of the earliest unsupervised learning approaches for voxel-based shape analysis. Without object labels for training, VConv-DAE chooses mean square loss or cross entropy loss as the reconstruction loss function. Girdhar et al. [45] also proposed the *TL-embedding network*, which combines an auto-encoder for generating a voxel-based representation with a convolutional neural network for predicting the embedding from 2D images.

Choy et al. [18] proposed *3D-R2N2* which takes single or multiple images as input and reconstructs objects within an occupancy grid. 3D-R2N2 regards input images as a sequence; its 3D recurrent neural network is based on LSTM (long short-term memory) [46] or GRU (gated recurrent units) [47]. The architecture consists of three parts: an image encoder to extract features from 2D images, 3D-LSTM to predict hidden states as coarse representations of final 3D models, and a decoder to increase the resolution and generate target shapes.

Wu et al. [19] designed a generative model called *3D-GAN* that applies a generative adversarial network (GAN) [41] to voxel data. 3D-GAN learns to synthesize a 3D object from a sampled latent space vector z with probability distribution $P(z)$. Moreover, Ref. [19] also proposed *3D-VAE-GAN* inspired by *VAE-GAN* [48] for the object reconstruction task. 3D-VAE-GAN puts the encoder before 3D-GAN to infer the latent vector z from input 2D images, and shares the decoder with the generator of 3D-GAN.

After early attempts to use volumetric representations with deep learning, researchers began to optimize the architecture of volumetric networks for better performance and more applications. The motivation is that a naive extension from traditional 2D networks often does not perform better than image-based CNNs such as MVCNN [14]. The main challenges affecting performance include overfitting, orientation, data sparsity, and low resolution.

Qi et al. [37] proposed two new network structures aiming to improve the performance of volumetric CNNs. One introduces an extra task, predicting class labels for subvolumes to prevent overfitting, and another utilizes elongated kernels to compress the 3D information into 2D in order to use 2D CNNs directly. Both use mlpconv layers [49] to replace traditional convolutional layers. Ref. [37] also augments the input data using different orientations and elevations to encourage the network to obtain more local features in different poses so that the results are less influenced by

orientation changes. To further mitigate the impact of orientation on recognition accuracy, instead of using data augmentation like Refs. [37, 40], Ref. [50] proposed a new model called *ORION* which extends VoxNet [40] and uses a fully connected layer to predict the object class label and orientation label simultaneously.

3.2 Sparse voxel representation (octree)

Voxel-based representations often lead to high computational cost because of the exponential increase in computations from pixels to voxels. Most methods cannot cope with or generate high-resolution models within a reasonable time. For instance, the *TL-embedding network* [45] was designed for a 20^3 voxel grid; *3DShapeNets* [13] and *VConv-DAE* [44] were designed for a 24^3 voxel grid with 3 voxels padding in each direction; *VoxNet* [40], *3D-R2N2* [18], and *ORION* [50] were designed for a 32^3 voxel grid; *3D-GAN* was designed to generate a 64^3 occupancy grid as a 3D shape representation. As the voxel resolution increases, the occupied voxels become sparser in the 3D space, which leads to more unnecessary computation. To address this problem, Li et al. [51] designed a novel method called *FPNN* to cope with data sparsity.

Some methods instead encode the voxel grid using a sparse, adaptive data structure, the octree [52] to reduce the dimensionality of the input data. Häne et al. [53] proposed hierarchical surface prediction (HSP) which can generate a voxel grid in the form of an octree from coarse to fine. Häne et al. observed that only the voxels near the object surface need to be predicted at high resolution, allowing the proposed HSP to avoid unnecessary calculation for affordable generation of a high resolution voxel grid. Each node in the octree is defined as a voxel block with a fixed number (16^3 in the paper) of voxels of different sizes, and each voxel block is classified as occupied, boundary, or free. The decoder of the model takes a feature vector as input, and predicts feature blocks that correspond to voxel blocks hierarchically. The HSP defines that the octree has 5 layers and each voxel block contains 16^3 voxels, so HSP can generate up to a grid of up to 256^3 voxels. Tatarchenko et al. [54] proposed a decoder called *OGN* for generating high resolution volumetric representations. Nodes in the octree are separated into three categories: empty, full, or mixed. The octree representing a 3D model and the feature map of the octree are stored in the form of hashing tables indexed by spatial position and octree level. In order to process feature maps represented as hash tables, Tatarchenko et al. designed a convolutional layer named *OGN-Conv*, which converts the convolutional operation into matrix multiplication. Ref. [54] generates different resolution octree cells in each decoder layer by convolutional operations on feature maps, and then decides whether to propagate the features to the next layer according to the label (propagating features if "boundary" and skipping feature propagation if "mixed").

Besides decoder model design for synthesizing voxel grids, shape analysis methods have also been designed using octrees. However, It is difficult to use conventional octree structure [52] in deep networks. Many researchers have tried to resolve the problem by designing new structures for octrees, and special operations such as convolution, pooling, and unpooling on octrees. Riegler et al. [21] proposed *OctNet*. Its octree representation has a more regular structure than a traditional octree, which places a shallow octree in cells of a regular 3D grid. Each shallow octree can have up to 3 levels and is encoded in 73 bits. Each bit determines if the corresponding cell needs to be split. Wang et al. [22] also proposed an octree-based convolutional neural network called *O-CNN*, where the model also removes pointers like a shallow octree [21] and stores the octree data and structure using a series of vectors, including shuffle key vectors, labels, and input signals.

Instead of representing voxels, octree structure can also be utilized to represent 3D surfaces with planar patches. Wang et al. [55] proposed *adaptive O-CNN*, based on a patch-guided adaptive octree, which divides a 3D surface into a set of planar patches restricted by bounding boxes corresponding to octants. They also provided an encoder and a decoder for the octree defined by this paper.

4 Surface-based representations

4.1 Point-based representation

4.1.1 Initial work

The typical point-based representation is also referred to as a point cloud or point set. It can be raw data generated by a 3D scanning device. Because

of its unordered and irregular structure, this kind of representation is relatively difficult to cope with using traditional deep learning methods. Therefore, most researchers avoided directly using point clouds in the early stages of deep learning-based geometry research. One of the first models to generate point clouds by deep learning came out in 2017 [20]. The authors designed a neural network to learn a point sampler based on a 3D point distribution. The network takes a single image and a random vector as input, and outputs an $N \times 3$ matrix representing a predicted point set (x, y, z coordinates for N points). *Chamfer distance* (*CD*) and *earth mover's distance* (*EMD*) [56] were used as loss functions to train the networks.

4.1.2 PointNet

At almost the same time, Charles et al. [6] proposed *PointNet* for shape analysis, which was the first successful deep network architecture to directly process point clouds without unnecessary rendering. Its pipeline is illustrated in Fig. 2. Taking account of three properties of point sets mentioned in Ref. [6], PointNet has three components in its network, including using max-pooling layers as symmetry functions for dealing with lack of ordering, concatenating global and local features for point interaction, and jointly aligning the network for transformation invariance. Based on PointNet, Qi et al. further improved this model in *PointNet++* [7], overcoming the problem that PointNet cannot capture and deal well with local features induced by the metric. In comparison to PointNet, PointNet++ introduces a hierarchical structure, allowing it to capture features at different scales, improving its ability to extract 3D shape features. As PointNet and PointNet++ showed state-of-the-art performance for shape classification and semantic segmentation, more and more deep learning models were proposed based on point-based representations.

4.1.3 CNNs for point clouds

Some research works focus on applying CNNs to analysis of irregular and unordered point clouds. Li et al. [24] proposed *PointCNN* for point clouds and designed the $\mathcal{X}$-transformation to weight and permute the input point features, guaranteeing equivariance for different point orders. Each feature matrix must be multiplied by the $\mathcal{X}$-transformation matrix before passing through the convolutional operator. This process is called the $\mathcal{X}$-Conv operator, which is the key element of *PointCNN*. Wang et al. [57] proposed *DGCNN*, a dynamic graph CNN architecture for point cloud classification and segmentation. Instead of processing point features like PointNet [6], *DGCNN* first connects neighboring points in spatial or semantic space to generate a graph, and then captures local geometric features by applying the EdgeConv operator to it. Moreover, unlike other graph CNNs which process a fixed input graph, *DGCNN* changes the graph to obtain new nearest neighbors in feature space in different layers, which is beneficial in providing larger and sparser receptive fields.

4.1.4 Other point cloud processing techniques using NNs

Klokov et al. [58] proposed the *k-d-network* to process point clouds based on the form of k-d-trees.

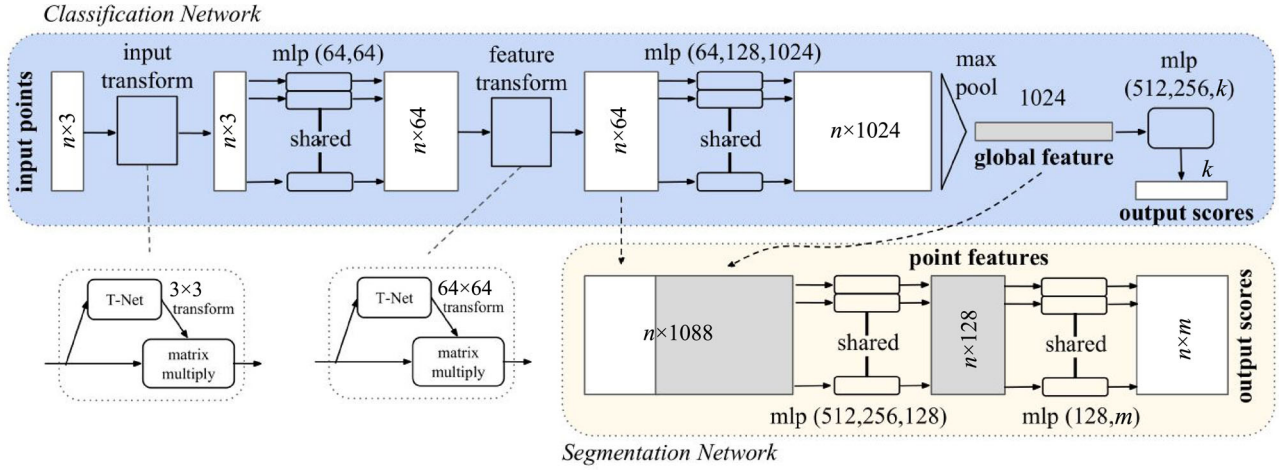


Fig. 2 Pipeline of *PointNet*. Reproduced with permission from Ref. [6], © IEEE 2017.

Yang et al. [59] proposed *FoldingNet*, an end-to-end auto-encoder for further compressing a point-based representation with unsupervised learning. Because point clouds can be transformed into a 2D grid by folding operations, FoldingNet integrates folding operations in their encoder–decoder to recover input 3D shapes. Mehr et al. [60] further proposed *DiscoNet* for 3D model editing by combining multiple autoencoders specifically trained for different types of 3D shapes. The autoencoders use pre-learned mean geometry of 3D training shapes as their templates. Meng et al. [61] proposed *VV-Net* (voxel VAE net) for point segmentation; it represents a point cloud by a structured voxel representation. Instead of using a Boolean value to represent occupancy of each voxel as in a normal volumetric representation, it uses a latent code computed by an RBF-VAE, a variational autoencoder based on radial basis function (RBF) interpolation of points, to describe point distribution within a voxel. This representation is used to extract intrinsic symmetry of point clouds using a group equivariant CNN, and the output is combined with PointNet [6] for better segmentation performance.

4.1.5 Observations

Although point-based representation can be more easily obtained from 3D scanners than other 3D representations, this raw form of 3D shape is typically unsuitable for 3D shape analysis, due to noise and data sparsity. Therefore, unlike other representations, it is essential for methods using point-based representation to incorporate an upsampling module to obtain fine-grained point clouds: see *PU-NET* [62], *MPU* [63], *PU-GAN* [64], etc. Additionally, point cloud registration is also an essential preprocessing step to fuse points from multiple scans: it aims to calculate rigid transformation parameters to align the point clouds. Wang et al. [65] proposed *deep closest point* (*DCP*), which extends the traditional iterative closest point (*ICP*) method [66], using a deep learning method to obtain the transformation parameters. Recently, Guo et al. [3] presented a survey focusing on deep learning models for point clouds, which provides more details in this field.

4.2 Mesh-based representations

Unlike point-based representations, mesh-based representations provide connectivity between neighboring points, so are more suitable for describing local regions on surfaces. As a typical type of representation in non-Euclidean space, mesh-based representations can be processed by deep learning models both in spatial and spectral domains [1].

4.2.1 Parametric representations for meshes

Directly applying CNNs to irregular data structures like meshes is non-trivial. A handful of approaches have emerged that map 3D shape surfaces to 2D domains such as 2D geometry images which can also be regarded as another 3D shape representation, and then apply traditional 2D CNNs to them [67, 68]. Based on geometry images, Sinha et al. [69] proposed *SurfNet* for shape generation using a deep residual network. Similarly, Shi et al. [70] projected 3D models into cylinder panoramic images, which are then processed by CNNs. Other methods convert mesh models into spherical signals, using a convolutional operator in the spherical domain for shape analysis. To address high-resolution signals on 3D meshes, in particular texture information, Huang et al. [71] proposed *TextureNet* to extract features, using a 4-rotationally symmetric (4-RoSy) field to parameterize surfaces. In the following, we review deep learning models according to how meshes are directly treated as input, and introduce generative models working on meshes.

4.2.2 Graphs

The mesh-based representation is constructed from sets of vertices and edges, and can be seen as a graph. Some models have been proposed based on the graph spectral theorem. They generalize CNNs on graphs [72–76] by eigen-decomposition of Laplacian matrices, generalizing convolutional operators to the spectral domain of graphs. Verma et al. [77] proposed another graph-based CNN, *FeaStNet*, which computes the receptive fields of the convolution operator dynamically. Specifically, it determines assignment of neighborhood vertices using features obtained from networks. Hanocka et al. [29] also designed operators for convolution, pooling, and unpooling for triangle meshes, and proposed *MeshCNN*. Unlike other graph-based methods, it focuses on processing features stored in edges, using a convolution operator applied to the edges with a fixed number of neighbors and a pooling operator based on edge collapse. MeshCNN extracts 3D shape features with respect to specific tasks, and learns to preserve important features and ignore unimportant ones.

4.2.3 2-Manifolds

The mesh-based representation can be viewed as a discretization of a 2-manifold. Several works have been designed using 2-manifolds with a series of refined CNN operators adapted to such non-Euclidean spaces. These methods define their own local patches and kernel functions when generalizing CNN models. Masci et al. [15] proposed *geodesic convolutional neural networks* (*GCNNs*) for manifolds, which extract and discretize local geodesic patches and apply convolutional filters to these patches in polar coordinates. The convolution operator works in the spatial domain and their geodesic CNN is quite similar to conventional CNNs applied in Euclidean space. *Localized spectral CNNs* [78] proposed by Boscaini et al. apply *windowed Fourier transforms* in non-Euclidean space. *Anisotropic convolutional neural networks* (*ACNNs*) [79] use an anisotropic heat kernel to replace the isotropic patch operator in GCNN [15], giving another solution to avoid ambiguity. Xu et al. [80] proposed *directionally convolutional networks* (*DCNs*), which define local patches based on faces of the mesh representation. They also designed a two-stream network for 3D shape segmentation, which takes local face normals and the global face distance histogram as training input. Moti et al. [81] proposed *MoNet* which replaces the weight functions in Refs. [15, 79] with Gaussian kernels with learnable parameters. Fey et al. [82] proposed *SplineCNN* which uses a convolutional operator based on B-splines. Pan et al. [83] designed a surface CNN for irregular 3D surfaces; it preserves the standard CNN property of translation equivariance by using parallel translation frames and group convolutional operations. Qiao et al. [84] proposed the *Laplacian pooling network* (*LaplacianNet*) for 3D mesh analysis. It considers both spectral and spatial information from the mesh, and contains 3 parts: preprocessing features as network input, mesh pooling blocks to split the surface and cluster patches for feature extraction, and a correlation network to aggregate global information.

4.2.4 Generative models

There are also many generative models for mesh-based representation. Wang et al. [23] proposed *Pixel2Mesh* for reconstructing 3D shapes from single images; it generates the target triangular mesh by deforming an ellipsoidal template. As shown in Fig. 3, the Pixel2Mesh network is implemented based on a *graph-based convolutional networks* (*GCNs*) [1] and generates the target mesh from coarse to fine by an unpooling operation. Wen et al. [85] advanced Pixel2Mesh and proposed *Pixel2Mesh++*, which extends single image 3D shape reconstruction to 3D shape reconstruction from multi-view images. To do so, Pixel2Mesh++ introduces a *multi-view deformation network* (*MDN*) to the original *Pixel2Mesh*; it incorporates cross-view information in the process of mesh generation. Groueix et al. [86] proposed *AtlasNet*, which generates 3D surfaces from multiple patches. AtlasNet learns to convert 2D square patches into 2-manifolds to cover the surface of 3D shapes using an MLP (multi-layer perceptron). Ben-Hamu et al. [87] proposed a multi-chart generative model for 3D shape generation. It uses a multi-chart structure as input; the network architecture is based on standard image GAN [41]. The transformation between 3D surface and multi-chart structure is based on Ref. [68]. However, methods based on deforming a template mesh into the target shape cannot express the complex topology of some 3D shapes. Pan et al. [88] proposed a new single-view reconstruction method which combines a deformation network and a topology modification network to model meshes with complex topology. In the topology modification network, faces with high distortion are removed. Tang et al. [89] proposed generating complex topology meshes using a skeleton-bridged learning method, as a skeleton can well preserve topology information. Instead of generating triangular meshes, Nash et al. [90] proposed *PolyGen* to generate a polygon mesh representation. Inspired by neural autoregressive models in other fields like natural language processing, they regarded mesh generation as a sequential process, and designed a transformer-based network [91], including a vertex model and a face model. The vertex model generates a sequence of vertex positions and the face model generates variable-length vertex sequences conditioned on input vertices.

5 Implicit representations

In addition to explicit representations such as point clouds and meshes, implicit representations have increased in popularity in recent studies. A major reason is that implicit representations are

not limited to fixed topology or resolution. An increasing number of deep models define their own implicit representations and build on them for various methods of shape analysis and generation.

5.1 Occupancy and indicator functions

Occupancy and indicator functions are one way to represent 3D shapes implicitly. An *occupancy network* was proposed by Mescheder et al. [8] to learn a continuous occupancy function as a new 3D shape representation for neural networks. The occupancy function reflects 3D point status with respect to the 3D shape's surface, where 1 means inside the surface and 0 otherwise. Researchers regarded this problem as a binary classification task and designed an occupancy network which inputs 3D point position and 3D shape observation and outputs the probability of occupancy. The generated implicit field is then processed by a multi-resolution isosurface extraction method *MISE* and marching cubes algorithm [10] to obtain a mesh. Moreover, researchers have introduced encoder networks to obtain latent embeddings. Similarly, Chen et al. [26] designed *IM-NET* as a decoder for learning generative models, which also takes an implicit function in the form of an indicator function.

5.2 Signed distance functions

Signed distance functions (SDFs) are another form of implicit representation. They map a 3D point to a real value instead of a probability, the value indicating the spatial relation and distance to the 3D surface. Let $\mathrm{SDF}(x)$ be the signed distance value of a given 3D point $x \in \mathbb{R}^3$. Then $\mathrm{SDF}(x) > 0$ if point x is outside the 3D shape, $\mathrm{SDF}(x) < 0$ if point x is inside the shape, and $\mathrm{SDF}(x) = 0$ if point x is on the surface. The absolute value of $\mathrm{SDF}(x)$ gives the distance between point x and the surface. Park et al. [25] proposed *DeepSDF* and introduced an *auto-decoder-based DeepSDF* as a new 3D shape representation. Xu et al. [9] also proposed *deep implicit surface networks* (*DISNs*) for single-view 3D reconstruction based on SDFs. Thanks to the advantages of SDFs, DISN was the first to reconstruct 3D shapes with flexible topology and thin structure in the single-view reconstruction task, which is difficult for other 3D representations.

5.3 Function sets

Occupancy functions and signed distance functions represent the 3D shape surface by a single function learned by a deep neural network. Genova et al. [92, 93] proposed representing an entire 3D shape by combining a set of shape elements. In Ref. [92], they proposed *structured implicit functions* (*SIFs*); each element is represented by a *scaled axis-aligned anisotropic 3D Gaussian*, and the sum of these shape elements represents the whole 3D shape. The Gaussians' parameters are learned by the CNN. Ref. [93] improved the SIF and proposed *deep structured implicit functions* (*DSIFs*) which added deep neural networks as *deep implicit functions* (*DIFs*) to provide local geometry details. To summarize, *DSIF* exploits *SIF* to depict coarse information for each shape element, and applies *DIF* for local shape details.

5.4 Approach without 3D supervision

The above implicit representation models need to sample 3D points in a 3D shape bounding box as ground truth and train the model supervised with 3D information. However, 3D ground truth may not be readily available in some situations. Liu et al. [30] proposed a framework which learns implicit representations without explicit 3D supervision. The model uses a field probing algorithm to bridge the gap between 3D shape and 2D images, using a silhouette loss to constrain 3D shape outline, and geometry

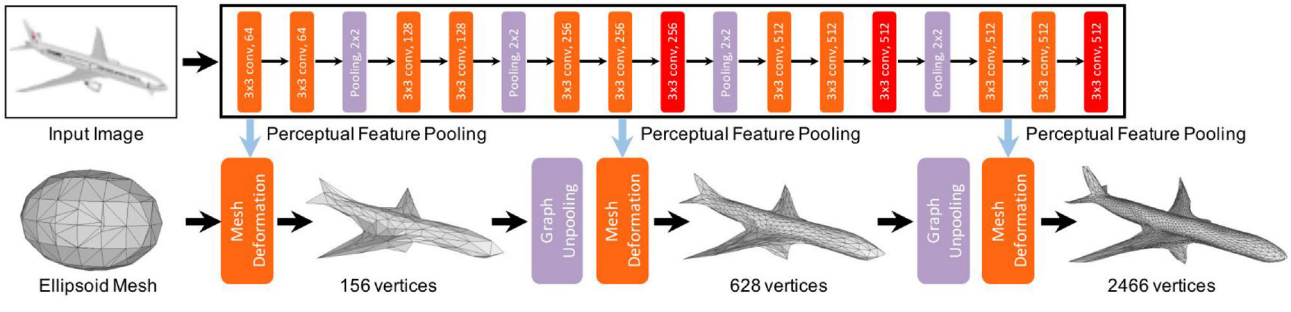


Fig. 3 Pipeline of *Pixel2Mesh*. Reproduced with permission from Ref. [23], © Springer Nature Switzerland AG 2018.

regularization to constrain the surface to be plausible.

6 Structure-based representations

Recently, more and more researchers have realized the importance of integrating structural information into deep learning models. Primitive representations are a typical kind of structure-based representation which explicitly depict 3D shape structure: they represent a 3D shape using several primitives such as oriented 3D boxes, using a compact parameter set. Instead of providing a description of geometric details, the primitive representation concentrates on the overall structure of a 3D shape. More importantly, obtaining a primitive representation encourages a method to generate more detailed and plausible 3D shapes.

6.1 Linear organization

Observing that humans often regard 3D shapes as a collection of parts, Zou et al. [11] proposed *3D-PRNN*, which applies LSTM in a primitive generator to generate primitives sequentially. The resulting primitive representations show great efficiency for depicting simple and regular 3D shapes. Wu et al. [94] further proposed an RCNN-based method called *PQ-NET* which also regards 3D shape parts as a sequence. The difference is that PQ-NET encodes geometry features in the network. Gao et al. [27] proposed a deep generative model named *SDM-NET* (structured deformable mesh-net). They designed a two-level VAE, containing a PartVAE for part geometry and an SP-VAE (structured parts VAE) for both structure and geometry features. Each shape part is encoded in a well designed form, which records both structure information (symmetry, supporting, and supported) and geometry features.

6.2 Hierarchical organization

Li et al. [12] proposed *GRASS* (generative recursive autoencoders for shape structures), one of the first attempts to encode 3D shape structure using a neural network. They describe shape structure in a hierarchical binary tree, in which child nodes are merged into the parent node by either adjacency or symmetry relations. Leaves in this structure tree represent oriented bounding boxes (OBBs) and geometry features for each part, while intermediate nodes represent both the geometric features of child nodes and relations between child nodes. Inspired by recursive neural networks (RvNNs) [33, 95], GRASS also recursively merges the codes representing the OBBs into a root code which depicts the whole shape structure. The architecture of GRASS has three parts: an RvNN autoencoder for encoding a 3D shape into a fixed length code, a GAN for learning the distribution of root codes and generating plausible structures, and another autoencoder (inspired by Ref. [45]) for synthesizing the geometry of each part. Furthermore, to synthesize fine-grained geometry in voxels, *structure-aware recursive features* (*SARFs*)

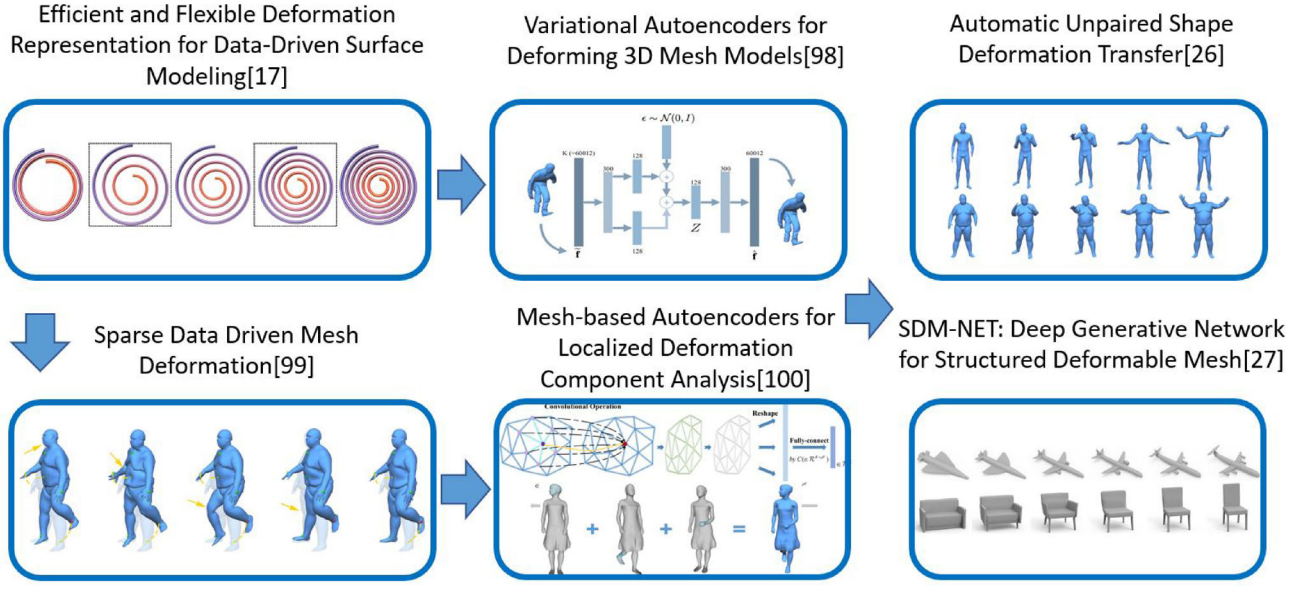


Fig. 4 Deformation-based shape representation used by the geometry learning group, ICT, CAS.

are used, which contain both the geometric features of each part and global and local OBB layout.

However, GRASS [12] uses a binary tree to organize the part structure, which leads to ambiguity; binary trees are unsuitable for large scale datasets. To address the problem, Mo et al. [28] proposed *StructureNet* which organizes the hierarchical structure in the form of graphs.

The *BSP-Net* (binary space partitioning-Net) proposed by Chen et al. [31] was the first method to depict sharp geometric features. It constructs a 3D shape from convex components organized in a BSP-tree [31]. The BSP-net includes three layers, for hyperplane extraction, hyperplane grouping, and shape assembly. The convex components can also be seen as a new form of primitive which can represent geometric details of 3D shapes rather than general structures.

6.3 Structure and geometry

Researchers have tried to encode 3D shape structure and geometric features separately [12] or jointly [96]. Wang et al. [97] proposed a *global-to-local* (*G2L*) generative model to generate man-made 3D shapes from coarse to fine. To address the problem that GANs cannot generate geometric details well [19], *G2L* first applies a GAN to generate a coarse voxel grid with semantic labels that represents shape structure at the global level, and then puts voxels separated by semantic labels into an autoencoder called the *part refiner* (*PR*) to optimize geometric details part by part at the local level. Wu et al. [96] proposed *SAGNet* for detailed 3D shape generation; it encodes structure and geometry jointly using a GRU [47] architecture in order to find relationships between them. SAGNet shows better performance for modeling tenon-mortise joints than other structure-based learning methods.

7 Deformation-based representations

Deformable 3D models play an important role in computer animation. However, most methods mentioned above focus on rigid 3D models, and pay less attention to deformation of non-rigid models. Unlike other representations, deformation-based representations parameterize the deformation information and achieve better performance for non-rigid 3D shapes such as articulated models.

7.1 Mesh-based approaches

A mesh can be seen as a graph, which is convenient when manipulating the vertex positions while maintaining the connectivity between vertices. Therefore, a great number of methods choose meshes to represent deformable 3D shapes. Based on this property, some mesh-based generation methods generate target shapes by deforming a mesh template [23, 27, 85, 88], and these methods can also be regarded as deformation-based methods. The graph structure makes it easy to store deformation information as vertex features, which can be seen as a deformation representation. Gao et al. [17] designed an efficient, rotation-invariant deformation representation called *rotation-invariant mesh difference* (*RIMD*), which achieves high performance for shape reconstruction, deformation, and registration. Based on Ref. [17], Tan et al. [98] proposed *Mesh VAE* for deformable shape analysis and synthesis. It takes *RIMD* as the feature inputs of VAE and uses fully connected layers for the encoder and decoder. Further, Gao et al. [99] designed an *as-consistent-as-possible* (*ACAP*) *representation* to constrain the rotation angle and rotation axes between adjacent vertices in the deformable mesh, to which graph convolution is easily applied. Tan et al. [100] proposed *SparseAE* based on the ACAP representation [99]. It applies graph convolutional operators [101] with ACAP [99] to analyse mesh deformations. Gao et al. [102] proposed *VC-GAN* (VAE CycleGAN) for unpaired mesh deformation transfer. It is the first automatic approach for unpaired mesh deformation transfer. It takes the ACAP representation as input, and encodes the representation into latent space by a VAE, and then transfers deformation between source and target in the latent space domain with cycle consistency and visual similarity consistency. Gao et al. [27] first viewed the geometric details shown in Fig. 5 as the deformations. Based on previous techniques [98–100, 102], geometric details can be encoded and generated. The structure in Ref. [27] is also analyzed to determine stable support in Ref. [103]. Yuan et al. [104] applied a newly designed pooling operation based on mesh simplification and graph convolution to the VAE architecture, which also takes ACAP representation as input to the network. Tan et al. [105] used ACAP representation for simulating thin-shell deformable

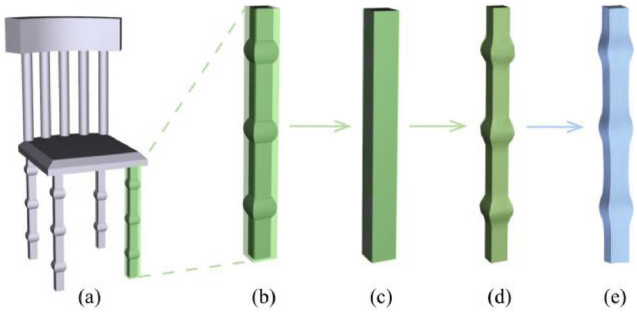


Fig. 5 Representing a chair leg by deforming the bounding box using *SDM-NET*. Reproduced with permission from Ref. [27], © ACM 2019.

materials, applying a graph-based CNN to embed high-dimensional features into low-dimensional features. In addition to considering a single deformable mesh, mesh sequences play an important role in computer animation. The deformation-based representation ACAP [99] is suitable for representing a mesh sequence. The deformation-based representation and related works are illustrated in Fig. 4.

7.2 Implicit surface-based approaches

With the development of implicit surface representations, Jeruzalski et al. [32] proposed a method to represent articulated deformable shapes by pose parameters, called *neural articulated shape approximation* (*NASA*). Pose parameters record the transformation of bones defined in models. They compared three different network architectures, including an unstructured model (U), a piecewise rigid model (R), and a piecewise deformable model (D) in the training dataset and test dataset, which opens another direction to represent deformable 3D shapes.

8 Datasets

With the development of 3D scanners, 3D models are easier to obtain, and more and more 3D shape datasets have been proposed with different 3D representations. The larger datasets with more details bring more challenges for existing techniques, further promoting the development of deep learning on different 3D representations.

The datasets can be divided into several types according to different representations and different applications. Choosing the appropriate type benefits performance and generalization for learning based models.

8.1 RGB-D images

RGB-D image datasets can be collected by depth sensors like *Microsoft Kinect*. Most RGB-D image datasets can be regarded as a video sequence. The *NYU Depth* [106, 107] indoor scene RGB-D image dataset was first provided as a benchmark for the segmentation problem. Version 1 [106] has 64 categories while the version 2 [107] has 464 categories. The *KITTI* [108] dataset provides outdoor scene images aimed mainly at autonomous driving, and contains 5 categories including road, city, residential, campus, and person. Depth maps for the images can be calculated using the development kit provided with the KITTI dataset. This dataset also contains 3D object annotations for applications such as object detection. *ScanNet* [109] is a large annotated RGB-D video dataset which includes 2.5M views with 3D camera pose of 1513 scenes, surface reconstructions, and semantic segmentations. Another dataset, *Human10* [110], is sampled from 10 human action sequences.

8.2 Man-made 3D objects

ModelNet [13] is a famous CAD model dataset for 3D shape analysis, including 127,915 3D CAD models in 662 categories. Two subsets, ModelNet10 and ModelNet40, include 10 and 40 categories from the whole dataset; in each subset, the 3D models are aligned manually. *ShapeNet* [111] provides a larger dataset, containing more than 3 million models in more than 4k categories. It also contains two smaller subsets: ShapeNetCore and ShapeNetSem. ShapeNet [111] provides rich annotations for 3D objects in the dataset, including category labels, part labels, symmetry information, etc. *ObjectNet3D* [112] is a large-scale dataset for 3D object recognition from 2D images. It includes 201,888 3D objects in 90,127 images and 44,147 different 3D shapes. The dataset is annotated with 3D pose parameters which align 3D objects with 2D images. *SUNCG* [113] includes full 3D models of rooms, and is suitable for 3D scene analysis and scene completion tasks. Its 3D models are represented by dense voxel grids with object annotations. The whole dataset includes 49,884 valid floors with 404,058 rooms and 5,697,217 object instances. *PartNet* [114] provides a more detailed CAD model dataset with fine-grained, hierarchical

part annotations, bringing more challenges, and resources for 3D object applications such as semantic segmentation, shape editing, and shape generation. 3D-Future [115] provides a large-scale furniture dataset, which includes over 20,000 scenes in over 5000 rooms with over 10,000 3D instances. Each 3D shape is of high quality; this dataset currently has the best texture information.

8.3 Non-rigid models

TOSCA [116] is a high-resolution 3D non-rigid model dataset containing 80 objects in 9 categories, in mesh representation. Objects in the same category have the same connectivity. *FAUST* [117] is a dataset of 3D human body scans of 10 different people in a variety of poses; ground truth correspondences are also provided. Because FAUST was proposed for real-world shape registration, the scans are noisy and incomplete, but the corresponding ground truth is water-tight and aligned. *AMASS* [118] provides a large and varied human motion dataset, gathering previous mocap datasets in a consistent framework and parameterization. It contains 344 subjects, 11,265 motions, and more than 40 hours of recordings.

9 Shape analysis and reconstruction

The shape representations discussed above are fundamental for shape analysis and shape reconstruction. In this section, we summarize representative works in these two directions respectively and compare their performance.

9.1 Shape analysis

Shape analysis methods usually extract latent codes from different 3D shape representations using different network architectures. The latent codes are then used for specific applications like shape classification, shape retrieval, shape segmentation, etc. Different representations are usually suited to different applications. We now review the performance of different representations in different models and discuss suitable representations for specific applications.

9.1.1 Shape classification and retrieval

Shape classification and retrieval are basic problems of shape analysis. Both rely on feature vectors extracted from the analysis networks. For shape classification, the datasets ModelNet10 and ModelNet40 [13] are widely used as benchmarks and Table 2 shows the accuracy of some different methods on ModelNet10 and ModelNet40. For shape retrieval, given a 3D shape as a query, the target is to find the most similar shape(s) in the dataset that match the query. Retrieval methods usually learn to find a compact code to represent the object in a latent space, and seek the closest object based on Euclidean distance, Mahalanobis distance, or some other distance metric. Unlike the classification task, shape retrieval has a

Table 1 3D model datasets

Source	Type	Dataset	Year	Categories	Items	Description
Real-world	RGB-D Images	NYU Depth v1 [106]	2011	64	—	Indoor Scene
Real-world	RGB-D Images	NYU Depth v2 [107]	2012	464	407024	Indoor Scene
Real-world	RGB-D Images	KITTI [108]	2013	5	—	Outdoor Scene
Real-world	RGB-D Images	ScanNet [109]	2017	1513	2.5M	Indoor Scene Video
Real-world	RGB-D Images	Human10 [110]	2018	10	9746	Human Action
Synthetic	3D CAD Models	ModelNet [13]	2015	662	127915	Mesh Representation
Synthetic	3D CAD Models	ModelNet10 [13]	2015	10	4899	—
Synthetic	3D CAD Models	ModelNet40 [13]	2015	40	12311	—
Synthetic	3D CAD Models	ShapeNet [111]	2015	4K	3M	Richly Annotated
Synthetic	3D CAD Models	ShapeNetCore [111]	2015	55	51300	—
Synthetic	3D CAD Models	ShapeNetSem [111]	2015	270	12000	—
Synthetic	Images and 3D Models	ObjectNet3D [112]	2016	100	44161	2D Aligned with 3D
Synthetic	3D CAD Models	SUNCG [113]	2017	—	49884	Full Room Scenes
Synthetic	3D CAD Models	PartNet [114]	2019	24	26671	573585 Part Instances
Synthetic	3D CAD Models	3D-FUTURE [115]	2020	—	10K	Texture Information
Synthetic	Non-Rigid Models	TOSCA [116]	2008	9	80	—
Real-world	Non-Rigid Models	FAUST [117]	2014	10	300	Human Bodies
Synthetic	Non-Rigid Models	AMASS [118]	2019	344	11265	Human Motions

Table 2 Accuracy of shape classification methods on ModelNet10 and ModelNet40 datasets

Form	Model	Accuracy(%)	
		MN10	MN40
Voxel	3DShapeNet [13]	83.54	77.32
Voxel	VoxNet [40]	92	83
Voxel	3D-GAN [19]	91.0	83.3
Voxel	Qi et al. [37]	—	86
Voxel	ORION [50]	93.8	—
Point	PointNet [6]	—	89.2
Multi-view	MVCNN [14]	—	90.1
Point	Kd-net [58]	93.3	90.6
Multi-view	Qi et al. [37]	—	91.4
Point	PointNet++ [7]	—	91.9
Point	Point2Sequence [119]	95.3	92.6

number of evaluation measures, including precision, recall, mAP (mean average precision), etc.

9.1.2 Shape segmentation

Shape segmentation aims to discriminate the parts of a 3D shape. This task plays an important role in understanding 3D shapes. Mean intersection-over-union (mIOU) is often used as the evaluation metric for shape segmentation. Most researchers choose to use point-based representation for the segmentation task [6, 7, 24, 58, 61].

9.1.3 Symmetry detection

Symmetry is important in 3D shapes, and can be further used in many other applications such as shape alignment, registration, completion, etc. Gao et al. [120] designed the first unsupervised deep learning method, *PRS-Net* (planar reflective symmetry net), to detect planar reflective symmetry in 3D shapes, using a new symmetry distance loss and a regularization loss, as illustrated in Fig. 6. It proved robust in the presence of noisy and incomplete input, and more efficient than traditional methods. As symmetry is largely determined by overall shape, *PRS-Net* is based on a 3D voxel CNN and has high performance at low resolution.

9.2 Shape reconstruction

Learning based generative models have been proposed for different representations, which is also an important field in geometry learning. Reconstruction applications include single-view shape reconstruction, shape generation, shape editing, etc. The generation methods can be summarized on the basis of representation. For voxel-based representations, learning based models try to predict the occupancy probability of each voxel. For point-based representations, learning based models either sample 3D points in space or fold the 2D grid into a target 3D object. For mesh-based representations, most generation methods choose to deform a mesh template into the final mesh. A recent study shows that more and more methods choose to use a structured representation and generate 3D shapes in a coarse-to-fine way.

10 Summary

This survey has reviewed deep learning methods based on different 3D object representations. We first overviewed different 3D representation learning models. The tendency in geometry learning can be summarized to be to reduce computation and memory demands, and to increase detail and structure. Then, we introduced 3D datasets widely used in research. These datasets provide rich resources and support evaluation of data-driven learning methods. Finally,

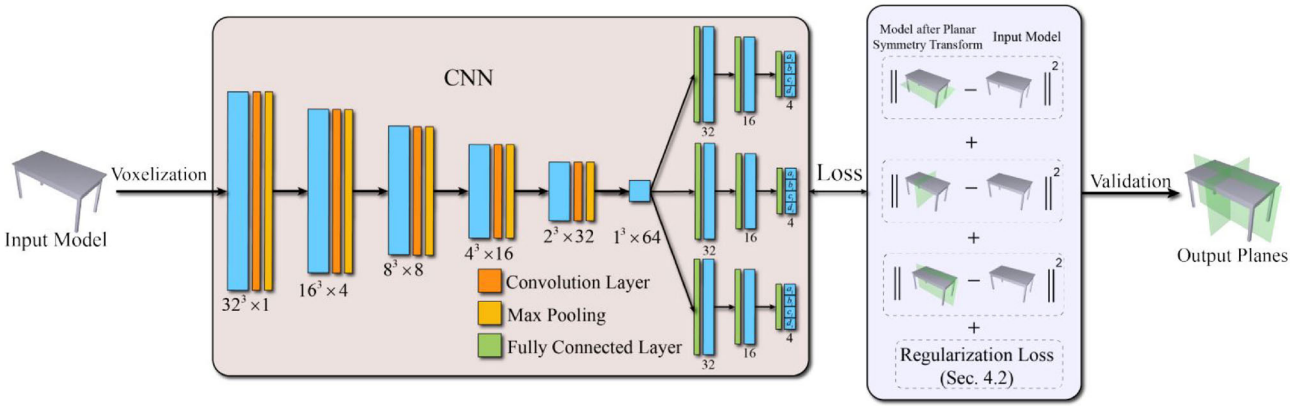


Fig. 6 Pipeline of PRS-Net. Reproduced with permission from Ref. [120].

we discuss 3D shape applications based on different 3D representations, including shape analysis and shape reconstruction. Different representations suit different applications; it is important to choose suitable 3D representations for specific tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61828204, 61872440), Beijing Municipal Natural Science Foundation (L182016), Youth Innovation Promotion Association CAS, CCF-Tencent Open Fund, Royal Society-Newton Advanced Fellowship (NAF\R2\192151), and the Royal Society (IES\R1\180126).

References

[1] Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine* Vol. 34, No. 4, 18–42, 2017.

[2] Ahmed, E.; Saint, A.; Shabayek, A. E. R.; Cherenkova, K.; Das, R.; Gusev, G.; Aouada, D.; Ottersten, B. Deep learning advances on different 3D data representations: A survey. *arXiv preprint* arXiv:1808.01462, 1, 2018.

[3] Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3D point clouds: A survey. *arXiv preprint* arXiv:1912.12033, 2019.

[4] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, 1097–1105, 2012.

[5] LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In: Proceedings of the IEEE International Symposium on Circuits and Systems, 253–256, 2010.

[6] Charles, R. Q.; Hao, S.; Mo, K. C.; Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 652–660, 2017.

[7] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the Advances in Neural Information Processing Systems, 5099–5108, 2017.

[8] Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; Geiger, A. Occupancy networks: Learning 3D reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4460–4470, 2019.

[9] Xu, Q.; Wang, W.; Ceylan, D.; Mech, R.; Neumann, U. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In: Proceedings of the Advances in Neural Information Processing Systems, 490–500, 2019.

[10] Lorensen, W. E.; Cline, H. E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics* Vol. 21, No. 4, 163–169, 1987.

[11] Zou, C. H.; Yumer, E.; Yang, J. M.; Ceylan, D.; Hoiem, D. 3D-PRNN: Generating shape primitives with recurrent neural networks. In: Proceedings of the IEE International Conference on Computer Vision, 900–909, 2017.

[12] Li, J.; Xu, K.; Chaudhuri, S.; Yumer, E.; Zhang, H.; Guibas, L. GRASS: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 52, 2017.

[13] Wu, Z. R.; Song, S. R.; Khosla, A.; Yu, F.; Zhang, L. G.; Tang, X. O.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1912–1920, 2015.

[14] 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, 945–953, 2015.

[15] Masci, J.; Boscaini, D.; Bronstein, M. M.; Vandergheynst, P. Geodesic convolutional neural networks on Riemannian manifolds. In: Proceedings of the IEEE International Conference on Computer Vision Workshop, 37–45, 2015.

[16] Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the Advances in Neural Information Processing Systems, 2366–2374, 2014.

[17] Gao, L.; Lai, Y.-K.; Liang, D.; Chen, S.-Y.; Xia, S. Efficient and flexible deformation representation for data-driven surface modeling. *ACM Transactions on Graphics* Vol. 35, No. 5, Article No. 158, 2016.

[18] Choy, C. B.; Xu, D. F.; Gwak, J.; Chen, K.; Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9912*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 628–644, 2016.

[19] Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3D generativeadversarial modeling. In: Proceedings of the Advances in Neural Information Processing Systems, 82–90, 2016.

[20] Fan, H. Q.; Su, H.; Guibas, L. A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 605–613, 2017.

[21] Riegler, G.; Ulusoy, A. O.; Geiger, A. OctNet: Learning deep 3D representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3577–3586, 2017.

[22] 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 Transactions on Graphics* Vol. 36, No. 4, Article No. 72, 2017.

[23] Wang, N. Y.; Zhang, Y. D.; Li, Z. W.; Fu, Y. W.; Liu, W.; Jiang, Y. G. Pixel2Mesh: Generating 3D mesh models from single RGB images. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11215*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 55–71, 2018.

[24] Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution on xtransformed points. In: Proceedings of the Advances in Neural Information Processing Systems, 820–830, 2018.

[25] Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[26] Chen, Z. Q.; Zhang, H. Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5939–5948, 2019.

[27] Gao, L.; Yang, J.; Wu, T.; Yuan, Y.-J.; Fu, H.; Lai, Y.-K.; Zhang, H. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 243, 2019.

[28] Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N. J.; Guibas, L. J. StructureNet: Hierarchical graph networks for 3D shape generation. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 242, 2019.

[29] Hanocka, R.; Hertz, A.; Fish, N.; Giryes, R.; Fleishman, S.; Cohen-Or, D. MeshCNN: A network with an edge. *ACM Transactions on Graphics* Vol. 38, No. 4, Article No. 90, 2019.

[30] Liu, S.; Saito, S.; Chen, W.; Li, H. Learning to infer implicit surfaces without 3D supervision. In: Proceedings of the Advances in Neural Information Processing Systems, 8293–8304, 2019.

[31] Chen, Z.; Tagliasacchi, A.; Zhang, H. BSP-Net: Generating compact meshes via binary space partitioning. *arXiv preprint* arXiv:1911.06971, 2019.

[32] Jeruzalski, T.; Deng, B.; Norouzi, M.; Lewis, J.; Hinton, G.; Tagliasacchi, A. NASA: Neural articulated shape approximation. *arXiv preprint* arXiv:1912.03207, 2019.

[33] Socher, R.; Huval, B.; Bath, B.; Manning, C. D.; Ng, A. Y. Convolutional-recursive deep learning for 3D object classification. In: Proceedings of the Advances in Neural Information Processing Systems, 656–664, 2012.

[34] Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8695*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer, Cham, 345–360, 2014.

[35] Gupta, S.; Arbelaez, P.; Girshick, R.; Malik, J. Aligning 3D models to RGB-D images of cluttered scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4731–4740, 2015.

[36] Song, S. R.; Xiao, J. X. Deep sliding shapes for amodal 3D object detection in RGB-D images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 808–816, 2016.

[37] Qi, C. R.; Su, H.; NieBner, M.; Dai, A.; Yan, M. Y.; Guibas, L. J. Volumetric and multi-view CNNs for object classification on 3D data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5648–5656, 2016.

[38] Hinton, G. E.; Osindero, S.; Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation* Vol. 18, No. 7, 1527–1554, 2006.

[39] Maturana, D.; Scherer, S. 3D convolutional neural networks for landing zone detection from LiDAR. In: Proceedings of the IEEE International Conference on Robotics and Automation, 3471–3478, 2015.

[40] Maturana, D.; Scherer, S. VoxNet: A 3D convolutional neural network for real-time object recognition. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 922–928, 2015.

[41] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In: Proceedings of the

Advances in Neural Information Processing Systems, 2672–2680, 2014.

[42] Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P. A. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning, 1096–1103, 2008.

[43] Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* Vol. 11, 3371–3408, 2010.

[44] Sharma, A.; Grau, O.; Fritz, M. VConv-DAE: Deep volumetric shape learning without object labels. In: *Computer Vision – ECCV 2016 Workshops. Lecture Notes in Computer Science, Vol. 9915*. Hua, G.; Jégou, H. Eds. Springer Cham, 236–250, 2016.

[45] Girdhar, R.; Fouhey, D. F.; Rodriguez, M.; Gupta, A. Learning a predictable and generative vector representation for objects. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 484–499, 2016.

[46] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* Vol. 9, No. 8, 1735–1780, 1997.

[47] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint* arXiv:1406.1078, 2014.

[48] Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint* arXiv:1512.09300, 2015.

[49] Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv preprint* arXiv:1312.4400, 2013.

[50] Sedaghat, N.; Zolfaghari, M.; Amiri, E.; Brox, T. Orientation-boosted voxel nets for 3D object recognition. In: Proceedings of the British Machine Vision Conference, 2017.

[51] Li, Y.; Pirk, S.; Su, H.; Qi, C. R.; Guibas, L. J. FPNN: Field probing neural networks for 3D data. In: Proceedings of the Advances in Neural Information Processing Systems, 307–315, 2016.

[52] Meagher, D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing* Vol. 19, No. 2, 129–147, 1982.

[53] Hane, C.; Tulsiani, S.; Malik, J. Hierarchical surface prediction for 3D object reconstruction. In: Proceedings of the International Conference on 3D Vision, 412–420, 2017.

[54] Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In: Proceedings of the IEEE International Conference on Computer Vision, 2088–2096, 2017.

[55] Wang, P.-S.; Sun, C.-Y.; Liu, Y.; Tong, X. Adaptive O-CNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 217, 2018.

[56] Rubner, Y.; Tomasi, C.; Guibas L. J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* Vol. 40, No. 2, 99–121, 2000.

[57] Wang, Y.; Sun, Y. B.; Liu, Z. W.; Sarma, S. E.; Bronstein, M. M.; Solomon, J. M. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics* Vol. 38, No. 5, Article No. 146, 2019.

[58] Klokov, R.; Lempitsky, V. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision, 863–872, 2017.

[59] Yang, Y. Q.; Feng, C.; Shen, Y. R.; Tian, D. FoldingNet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 206–215, 2018.

[60] Mehr, E.; Jourdan, A.; Thome, N.; Cord, M.; Guitteny, V. DiscoNet: Shapes learning on disconnected manifolds for 3D editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 3474–3483, 2019.

[61] Meng, H. Y.; Gao, L.; Lai, Y. K.; Manocha, D. VV-net: Voxel VAE net with group convolutions for point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 8500–8508, 2019.

[62] Yu, L. Q.; Li, X. Z.; Fu, C. W.; Cohen-Or, D.; Heng, P. A. PU-Net: Point cloud upsampling network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2790–2799, 2018.

[63] Wang, Y. F.; Wu, S. H.; Huang, H.; Cohen-Or, D.; Sorkine-Hornung, O. Patch-based progressive 3D point set upsampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5958–5967, 2019.

[64] Li, R. H.; Li, X. Z.; Fu, C.W.; Cohen-Or, D.; Heng, P.A. PU-GAN: A point cloud upsampling adversarial network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7203–7212, 2019.

[65] Wang, Y.; Solomon, J. Deep closest point: Learning representations for point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 3523–3532, 2019.

[66] Besl, P. J.; McKay, N. D. Method for registration of 3-D shapes. In: Proceedings of the SPIE 1611, Sensor Fusion IV: Control Paradigms and Data Structures, 586–606, 1992.

[67] Sinha, A.; Bai, J.; Ramani, K. Deep learning 3D shape surfaces using geometry images. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 223–240, 2016.

[68] Maron, H.; Galun, M.; Aigerman, N.; Trope, M.; Dym, N.; Yumer, E.; Kim, V. G.; Lipman, Y. Convolutional neural networks on surfaces via seamless toric covers. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 71, 2017.

[69] Sinha, A.; Unmesh, A.; Huang, Q. X.; Ramani, K. SurfNet: Generating 3D shape surfaces using deep residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6040–6049, 2017.

[70] Shi, B. G.; Bai, S.; Zhou, Z. C.; Bai, X. DeepPano: Deep panoramic representation for 3-D shape recognition. *IEEE Signal Processing Letters* Vol. 22, No. 12, 2339–2343, 2015.

[71] Huang, J. W.; Zhang, H. T.; Yi, L.; Funkhouser, T.; NieBner, M.; Guibas, L. J. TextureNet: Consistent local parametrizations for learning from high-resolution signals on meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4440–4449, 2019.

[72] Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv preprint* arXiv:1312.6203, 2013.

[73] Henaff, M.; Bruna, J.; LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv preprint* arXiv:1506.05163, 2015.

[74] Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the Advances in Neural Information Processing Systems, 3844–3852, 2016.

[75] Kipf, T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint* arXiv:1609.02907, 2016.

[76] Atwood, J.; Towsley, D. Diffusionconvolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, 1993–2001, 2016.

[77] Verma, N.; Boyer, E.; Verbeek, J. FeaStNet: Feature-steered graph convolutions for 3D shape analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2598–2606, 2018.

[78] Boscaini, D.; Masci, J.; Melzi, S.; Bronstein, M. M.; Castellani, U.; Vandergheynst, P. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum* Vol. 34, No. 5, 13–23, 2015.

[79] Boscaini, D.; Masci, J.; Rodolà, E.; Bronstein, M. Learning shape correspondence with anisotropic convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, 3189–3197, 2016.

[80] Xu, H. T.; Dong, M.; Zhong, Z. C. Directionally convolutional networks for 3D shape segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, 2698–2707, 2017.

[81] Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5115–5124, 2017.

[82] Fey, M.; Lenssen, J. E.; Weichert, F.; Müller, H. SplineCNN: Fast geometric deep learning with continuous B-spline kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 869–877, 2018.

[83] Pan, H.; Liu, S.; Liu, Y.; Tong, X. Convolutional neural networks on 3D surfaces using parallel frames. *arXiv preprint* arXiv:1808.04952, 2018.

[84] Qiao, Y.-L.; Gao, L.; Yang, J.; Rosin, P. L.; Lai, Y.-K.; Chen, X. LaplacianNet: Learning on 3D meshes with Laplacian encoding and pooling. *arXiv preprint* arXiv:1910.14063, 2019.

[85] Wen, C.; Zhang, Y. D.; Li, Z. W.; Fu, Y. W. Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1042–1051, 2019.

[86] Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; Aubry, M. A papier-Mache approach to learning 3D surface generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 216–224, 2018.

[87] Ben-Hamu, H.; Maron, H.; Kezurer, I.; Avineri, G.; Lipman, Y. Multi-chart generative surface modeling. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 215, 2019.

[88] Pan, J. Y.; Han, X. G.; Chen, W. K.; Tang, J. P.; Jia, K. Deep mesh reconstruction from single RGB images via topology modification networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9964–9973, 2019.

[89] Tang, J. P.; Han, X. G.; Pan, J. Y.; Jia, K.; Tong, X. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4541–4550, 2019.

[90] Nash, C.; Ganin, Y.; Eslami, S.; Battaglia P. W. PolyGen: An autoregressive generative model of 3D meshes. *arXiv preprint* arXiv:2002.10880, 2020.

[91] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems, 5998–6008, 2017.

[92] Genova, K.; Cole, F.; Vlasic, D.; Sarna, A.; Freeman, W.; Funkhouser, T. Learning shape templates with structured implicit functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7154–7164, 2019.

[93] Genova, K.; Cole, F.; Sud, A.; Sarna, A.; Funkhouser, T. Deep structured implicit functions. *arXiv preprint* arXiv:1912.06126, 2019.

[94] Wu, R.; Zhuang, Y.; Xu, K.; Zhang, H.; Chen, B. PQ-NET: A generative part seq2seq network for 3D shapes. *arXiv preprint* arXiv:1911.10949, 2019.

[95] Socher, R.; Lin, C. C.; Manning, C.; Ng, A. Y. Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th International Conference on Machine Learning, 129–136, 2011.

[96] Wu, Z.; Wang, X.; Lin, D.; Lischinski, D.; Cohen-Or, D.; Huang, H. SAGNet: Structure-aware generative network for 3D shape modeling. *ACM Transactions on Graphics* Vol. 38, No. 4, Article No. 91, 2019.

[97] Wang, H.; Schor, N.; Hu, R.; Huang, H.; Cohen-Or, D.; Huang, H. Global-tolocal generative model for 3D shapes. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 214, 2018.

[98] Tan, Q. Y.; Gao, L.; Lai, Y. K.; Xia, S. H. Variational autoencoders for deforming 3D mesh models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5841–5850, 2018.

[99] Gao, L.; Lai, Y.K.; Yang, J.; Zhang, L.-X.; Xia, S. H.; Kobbelt, L. Sparse data driven mesh deformation. *IEEE Transactions on Visualization and Computer Graphics* DOI: 10.1109/TVCG.2019.2941200, 2019.

[100] Tan, Q.; Gao, L.; Lai, Y.-K.; Yang, J.; Xia, S. Mesh-based autoencoders for localized deformation component analysis. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018.

[101] Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In: Proceedings of the Advances in Neural Information Processing Systems, 2224–2232, 2015.

[102] Gao, L.; Yang, J.; Qiao, Y.-L.; Lai, Y.-K.; Rosin, P. L.; Xu, W.; Xia, S. Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 237, 2018.

[103] Huang, S. S.; Fu, H. B.; Wei, L. Y.; Hu, S. M. Support substructures: Support-induced part-level structural representation. *IEEE Transactions on Visualization and Computer Graphics* Vol. 22, No. 8, 2024–2036, 2016.

[104] Yuan, Y.-J.; Lai, Y.-K.; Yang, J.; Fu, H.; Gao, L. Mesh variational autoencoders with edge contraction pooling. *arXiv preprint* arXiv:1908.02507, 2019.

[105] Tan, Q. Y.; Pan, Z. R.; Gao, L.; Manocha, D. Realtime simulation of thin-shell deformable materials using CNN-based mesh embedding. *IEEE Robotics and Automation Letters* Vol. 5, No. 2, 2325–2332, 2020.

[106] Silberman, N.; Fergus, R. Indoor scene segmentation using a structured light sensor. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2011.

[107] Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from RGBD images. In: *Computer Vision – ECCV 2012. Lecture Notes in Computer Science, Vol. 7576*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 746–760, 2012.

[108] Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* Vol. 32, No. 11, 1231–1237, 2013.

[109] Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; Niessner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5828–5839, 2017.

[110] Cao, Y. P.; Liu, Z. N.; Kuang, Z. F.; Kobbelt, L.; Hu, S. M. Learning to reconstruct high-quality 3D shapes with cascaded fully convolutional networks In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11213*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 626–643, 2018.

[111] Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H. et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint* arXiv:1512.03012, 2015.

[112] Xiang, Y.; Kim, W.; Chen, W.; Ji, J. W.; Choy, C.; Su, H.; Mottaghi, R.; Guibas, L.; Savarese, S. ObjectNet3D: A large scale database for 3D object recognition. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9912*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 160–176, 2016.

[113] Song, S. R.; Yu, F.; Zeng, A.; Chang, A. X.; Savva, M.; Funkhouser, T. Semantic scene completion from a single depth image. In: Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[114] Mo, K. C.; Zhu, S. L.; Chang, A. X.; Yi, L.; Tripathi, S.; Guibas, L. J.; Su, H. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 909–918, 2019.

[115] Fu, H.; Jia, R.; Gao, L.; Gong, M.; Zhao, B.; Maybank, S.; Tao, D. 3D-FUTURE: 3D FUrniture shape with TextURE. 2020. Available at https://tianchi.aliyun.com/specials/promotion/alibaba-3d-future.

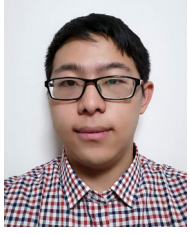
[116] Bronstein, A. M.; Bronstein, M. M.; Kimmel, R. *Numerical Geometry of Non-Rigid Shapes*. Springer Science & Business Media, 2008.

[117] Bogo, F.; Romero, J.; Loper, M.; Black, M. J. FAUST: Dataset and evaluation for 3D mesh registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3794–3801, 2014.

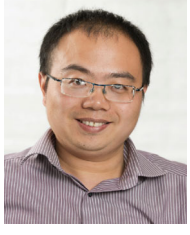
[118] Mahmood, N.; Ghorbani, N.; Troje, N. F.; Pons-Moll, G.; Black, M. AMASS: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5442–5451, 2019.

[119] Liu, X. H.; Han, Z. Z.; Liu, Y.S.; Zwicker, M. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 33, 8778–8785, 2019.

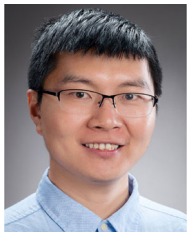
[120] Gao, L.; Zhang, L. X.; Meng, H. Y.; Ren, Y. H.; Lai, Y. K.; Kobbelt, L. PRS-Net: Planar reflective symmetry detection net for 3D models. *arXiv preprint* arXiv:1910.06511, 2019.



Yun-Peng Xiao received his bachelor degree in computer science from Nankai University. He is currently a master student in the Institute of Computing Technology, the Chinese Academy of Sciences. His research interests include computer graphics and geometric processing.



Yu-Kun Lai received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Reader in the School of Computer Science & Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial boards of *Computer Graphics Forum* and *The Visual Computer*.



Fang-Lue Zhang is currently a lecturer with Victoria University of Wellington, New Zealand. He received his bachelor degree from Zhejiang University, Hangzhou, in 2009, and doctoral degree from Tsinghua University, Beijing, in 2015. His research interests include image and video editing, computer vision, and computer graphics. He is a member of IEEE and ACM. He received a Victoria Early-Career Research Excellence Award in 2019.

Chunpeng Li received his Ph.D. degree in 2008 and now is an associate professor at the Institute of Computing Technology, the Chinese Academy of Sciences. His main research interests are in virtual reality, human–computer interaction, and computer graphics.

Lin Gao received his bachelor degree in mathematics from Sichuan University and Ph.D. degree in computer science from Tsinghua University. He is currently an associate professor at the Institute of Computing Technology, the Chinese Academy of Sciences. His research interests include computer graphics and geometric processing. He received a Newton Advanced Fellowship award from the Royal Society in 2019.