

# SkeletonNet: A Topology-Preserving Solution for Learning Mesh Reconstruction of Object Surfaces from RGB Images

Jiapeng Tang, Xiaoguang Han, Mingkui Tan, Xin Tong, Kui Jia

**Abstract**—This paper focuses on the challenging task of learning 3D object surface reconstructions from RGB images. Existing methods achieve varying degrees of success by using different surface representations. However, they all have their own drawbacks, and cannot properly reconstruct the surface shapes of complex topologies, arguably due to a lack of constraints on the topological structures in their learning frameworks. To this end, we propose to learn and use the topology-preserved, skeletal shape representation to assist the downstream task of object surface reconstruction from RGB images. Technically, we propose the novel *SkeletonNet* design that learns a volumetric representation of a skeleton via a bridged learning of a skeletal point set, where we use parallel decoders each responsible for the learning of points on 1D skeletal curves and 2D skeletal sheets, as well as an efficient module of globally guided subvolume synthesis for a refined, high-resolution skeletal volume; we present a differentiable *Point2Voxel* layer to make *SkeletonNet* end-to-end and trainable. With the learned skeletal volumes, we propose two models, the Skeleton-Based Graph Convolutional Neural Network (*SkeGCNN*) and the Skeleton-Regularized Deep Implicit Surface Network (*SkeDISN*), which respectively build upon and improve over the existing frameworks of explicit mesh deformation and implicit field learning for the downstream surface reconstruction task. We conduct thorough experiments that verify the efficacy of our proposed *SkeletonNet*. *SkeGCNN* and *SkeDISN* outperform existing methods as well, and they have their own merits when measured by different metrics. Additional results in generalized task settings further demonstrate the usefulness of our proposed methods. We have made both our implementation code and the ShapeNet-Skeleton dataset publicly available at <https://github.com/tangjiapeng/SkeletonNet>.

**Index Terms**—Surface reconstruction learning from RGB images, skeleton, mesh deformation, implicit surface field

## 1 INTRODUCTION

RECONSTRUCTING surface geometries of objects or scenes from RGB images is a constituent field of computer vision research. Studies on this field of 3D vision date at least back to David Marr [1]. Classical approaches of multi-view geometry [2] have also been developed since then, and they serve as the foundational technologies in many applications, e.g., virtual or augmented reality, robotic navigation, and autonomous driving. While 3D surface reconstruction from multi-view images is traditionally solved by local correspondence matching [3], [4], [5], followed by a global optimization [6], [7], [8], it has been recently shown that, by leveraging the great modeling capacities of deep networks, the 3D surface shapes of generic objects can be learned and reconstructed from as few as a single image [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. In spite of these preliminary successes, this inverse problem is in fact quite difficult to learn due to

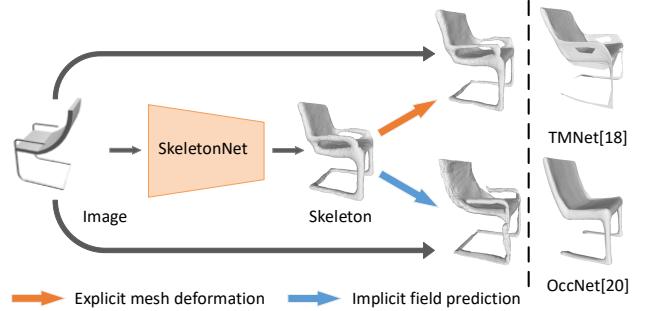


Fig. 1: Given an input RGB image of an object instance, we aim to recover a surface mesh of the object and expect its topology to be correct. To tackle this challenging task, we propose *SkeletonNet*, an end-to-end model that is able to efficiently generate a high-quality skeletal shape representation whose topology is the same as that of the underlying surface. We use the generated skeleton respectively in the state-of-the-art frameworks of explicit and implicit mesh recoveries from RGB images, and obtain improved results.

- *J. Tang and K. Jia are with the School of Electronic and Information Engineering, South China University of Technology, and with Pazhou Lab, Guangzhou, China. E-mail: msjptang@mail.scut.edu.cn, kuijia@scut.edu.cn.*
- *X. Han is with The Chinese University of Hong Kong, Shenzhen and Shenzhen Research Institute of Big Data, Shenzhen, China. E-mail: hanxiaoguang@cuhk.edu.cn.*
- *M. Tan is with the School of Software Engineering, South China University of Technology, Guangzhou, China. E-mail: mingkuitan@scut.edu.cn.*
- *X. Tong is with Microsoft Research Asia, Beijing, China. E-mail: xtong@microsoft.com.*
- *The first two authors contributed equally to this work. K. Jia is the corresponding author.*

the arbitrary shapes of object instances and their possibly complex topologies.

Existing methods of deep learning surface reconstruction are based on different shape representations, and they range from explicit ones using point set or mesh to implicit ones using volume or continuous field. Volume-based methods

[9], [10], [24] exploit the establishment of Convolutional Neural Networks (CNNs) [25], [26], [27], [28], and simply extend CNNs as 3D versions to generate volumetric representations of surface shapes. These methods suffer from high computation and memory costs, which prohibit them from generating high-quality shapes represented by high-resolution volumes. This limitation is to some extent addressed by octree-based methods [11], [29], [30], [31], [32], [33], by designing hierarchical models based on the efficient data structures of the octree. The recent methods of deep implicit field learning [20], [21], [22], [23] address this issue more elegantly by learning Multi-Layer Perception (MLP)-based field functions; since the learned functions represent implicit but continuous fields of object surfaces, high-quality surface meshes can be obtained simply by querying enough points in the 3D space, followed by a final marching cubes operation [34]. For an explicit surface reconstruction, a deep point set regression is used in [14], [35], [36], [37] that directly regresses the coordinates of surface points. However, the discrete surface representation of point set is of lower efficiency and to improve this efficiency, point set regression is replaced in [12], [13], [15] with the deformation of vertices in the given and initial meshes; as such, more efficient, continuous surface meshes are thus obtained. However, these explicit mesh reconstruction methods are limited by the bypassed challenge of learning the true vertex connections that determine the topological structures of the surface to be reconstructed and that may not be consistent with those given in the initial meshes.

Among the various challenges faced by deep learning surface reconstruction from RGB images, common to existing methods is their less efficacy to reconstruct those of complex topologies. Indeed, explicit methods have the difficulty of learning the true vertex connections, while implicit methods have no constraints on what topological structures they would learn. To address this challenge, we connect with the *skeleton*, an interior shape representation of object surface (cf. Sec. 3 for the technical definition), which has the nice properties of preserving the topology of an object surface, while being of lower complexity to learn, and we expect that such a representation would be beneficial to both the implicit and explicit recoveries of surface meshes in a topology-aware manner. Technically, we propose an end-to-end, trainable system, termed *SkeletonNet*, to learn skeletal shape representations from RGB images. SkeletonNet outputs skeletal volumes via the bridged learning of skeletal point sets. The skeleton is originally represented as a set of points sampled from 1D *skeletal curves* or 2D *skeletal sheets* [38], [39]; we correspondingly present a parallel design of *CurSkeDecoder* and *SurSkeDecoder* in the SkeletonNet module of skeletal point set learning. An efficient module of globally guided subvolume synthesis is further stacked in SkeletonNet in order to obtain a high-resolution skeletal volume. And this volumetric representation refines the skeleton by establishing an improved connectivity among skeletal points. Connecting the two modules above is a differentiable *Point2Voxel* layer, which makes SkeletonNet an end-to-end model, as illustrated in Fig. 2.

In this work, we are motivated to use this topology-preserved, skeletal shape representation to assist the downstream task of object surface recovery from RGB images.

We study how to use the volumetric skeleton output of SkeletonNet either as a bridge to explicitly recover a surface mesh, or as a constraint to regularize the learning of an implicit surface field. For the former setting, we present a model of Skeleton-Based Graph CNN (SkeGCNN) that first extracts an initial mesh from the obtained skeletal volume, and then learns the weights of the GCNN to deform the vertices of the initial mesh, such that the initial mesh is inflated to fit the object surface. For the latter setting, we extend the recent state-of-the-art method of Deep Implicit Surface Network (DISN) [23], and we present a Skeleton-Regularized DISN (SkeDISN) model that extracts multi-scale, local features from the skeletal volume to regularize the learning of a binary 3D field of object occupancy. We conduct thorough experiments that evaluate our proposed SkeletonNet for both its efficacy at producing high-quality skeletal representations and its usefulness in the downstream tasks of explicitly and implicitly learning mesh recoveries from RGB images. To enable empirical studies, we contribute a *ShapeNet-Skeleton* dataset, which has prepared ground-truth skeletal point sets and skeletal volumes for object instances in ShapeNet [40]. Careful ablations and comparisons with various alternatives verify the importance of the components of our proposed SkeletonNet. Experiments of explicit and implicit mesh recoveries show that the use of a skeleton as a topology-preserved intermediate feature improves mesh recoveries on most object instances. It is also interesting to observe that SkeGCNN and SkeDISN outperform the respective existing methods of explicit and implicit mesh recoveries from RGB images, and they have their own merits when measured by different quantitative metrics or compared visually.

## 1.1 Contributions

Existing methods of deep learning mesh recovery from RGB images are less effective for object instances that have complex surface topologies. Our previous work [16] offers a preliminary attempt at using skeletons to address this limitation, where given an input RGB image, a separated, three-stage approach is used to successively predict a skeletal point set, a skeletal volume, and a final surface mesh. In the present paper, we aim for a systematic study on the usefulness of skeletal shape representations on the mesh recovery of object surfaces. To this end, we propose the end-to-end, trainable model of SkeletonNet for learning to produce skeletal shape representations. We further design models that use skeletons in the state-of-the-art frameworks of both explicit and implicit mesh recoveries from RGB images. Results of these models confirm the usefulness of skeletal representations produced by SkeletonNet. We summarize our technical contributions as follows.

- We propose the novel design of *SkeletonNet* to learn skeletal shape representations from RGB images. SkeletonNet stacks two modules that can efficiently produce high-quality skeletal volumes via a bridged learning of skeletal point sets. We connect the two modules by a differentiable Point2Voxel layer that makes SkeletonNet end-to-end and trainable.
- To study the usefulness of the obtained skeletal volumes, we propose two models, the Skeleton-

Based Graph CNN (SkeGCNN) and the Skeleton-Regularized Deep Implicit Surface Network (SkeDISN), in the respective state-of-the-art frameworks of explicit and implicit mesh recoveries from RGB images. The former model learns to recover a surface mesh by inflating an initial mesh extracted from the skeletal volume, and the latter model regularizes the learning of a binary occupancy field using multi-scale, local features of the skeletal volume.

- We conduct thorough experiments that confirm both the efficacy of SkeletonNet at producing high-quality skeletal representations, as well as its usefulness for downstream surface recovery tasks. By comparing SkeGCNN and SkeDISN with existing methods, the respective advantages of explicit and implicit methods under different measures manifest themselves as well. Experiments under generalized settings, including deep learning mesh recovery from multi-view images, further demonstrate the usefulness of our proposed methods.

## 2 RELATED WORKS

In this section, we review existing methods of deep learning surface reconstruction from RGB images. We focus on the mesh representation of object surfaces, while other representations, e.g., point sets or volumes, are briefly discussed as well. We organize this review into two lines of methods that can either explicitly learn a deformation of mesh vertices or learn an implicit surface representation from which the surface mesh can be extracted. We finally discuss existing research where the interior shape representation of the skeleton is used for various downstream tasks.

**Explicit Surface Learning** Point set is a discrete and simple, and arguably the most popular representation of object surface. Deep learning surface reconstruction starts from recovery of the surface point set. For example, Fan et al. [14] propose learning a deep regression model for generating coordinates of points residing on the surface of an object observed in a single-view image, where the Chamfer distance (CD) or the Earth Mover’s distance (EMD) between point sets are used to build the training objective. To improve the fidelity of the generated point sets, adversarial training is adopted by Jiang et al. [41] and Achlioptas et al. [42] to constrain the distribution of the learned latent space. Our proposed SkeletonNet is also concerned with an intermediate prediction of skeletal point set, for which we build upon existing works [12] but introduce novel designs as well, in order to better generate skeletal points whose neighboring relations form 1D skeletal curves and 2D skeletal sheets.

Meshes are efficient and continuous surface representations that connect isolated points, i.e. mesh vertices, with mesh edges and faces. Learning to reconstruct a surface mesh directly is challenged by the learning of connecting relations among mesh vertices. To alleviate this issue, existing methods [13], [15], [18] convert this challenge to learning the vertex deformation of initial meshes. For example, AtlasNet [13] proposes to learn deep networks to deform multiple, initial meshes and expects the deformed ones to cover the surface altogether; as such, it fails to produce a closed

surface mesh. Pix2Mesh [15] is able to obtain a closed mesh by learning to deform vertices of a single, initial mesh; however, given the fixed topology of the initial mesh, it cannot learn to generate surface meshes whose topologies are unknown. Although some methods [43], [44], [45] search for initial mesh structures from a set of reference models, they are still limited when it comes to reconstruct the surface meshes of complex topologies. More recently, Pan et al. [18] present the topology modification network (TMNet) to learn and update mesh connectivity when deforming the initial mesh, where topology modification is achieved by a mechanism of face pruning based on estimated face-to-surface distances. This is further improved by Nie et al. [46] with an adaptive thresholding strategy. However, learning the relations of vertex connections dynamically during the deformation process is a rather challenging task, causing these approaches to be less effective for the recovery of thin structures (cf. the examples in Fig. 1).

The present work acknowledges these challenges faced by existing methods. Instead of addressing these challenges directly, we propose to decouple these learning difficulties as challenges concerned with the recovery of topological structures and challenges concerned with the recovery of surface geometries. Technically, we propose SkeletonNet and expect that the produced skeletal representations can serve as topology-preserved, initial meshes to be subsequently inflated as surface meshes.

**Implicit Surface Learning** An object surface can also be represented implicitly as a discrete volume or a continuous field, from which a surface mesh can be extracted, e.g., via marching cubes [34]. Learning to reconstruct volumetric shapes gains earlier popularity due to the regularity of the data structures involved [9], [10], [24], [47]. These methods simply extend the CNNs for 2D image modeling to their 3D counterparts. For example, the work of [9] combines 3D convolutions and long short-term memory (LSTM) units to achieve volumetric grid reconstructions from single- or multi-view RGB images. A 3D auto-encoder is trained in [10] whose decoder is used to build the mapping from a 2D image to a 3D occupancy grid. Wu et al. [24] adopt generative adversarial networks [48] to learn a better volumetric shape reconstruction. Due to the high computation cost of 3D convolution, these methods are only able to produce low-resolution volumes. The hierarchical structure of an octree would alleviate this issue to some extent [11], [29], [30], [31], [32], [33]; however, it seems that such an octree-based data structure makes the training of compatibly designed deep models less effective, resulting in shape reconstructions that often have artifacts.

It has been recently discovered that various representations of implicit surface fields are quite effective for object surface modeling and reconstruction, with the signed distance function (SDF) [22], [49], [50] and the object occupancy field [20], [21] as prominent examples. For example, the works of [19], [20], [21], [51] learn a 3D field of object occupancy, where an object’s surface is considered as a continuous decision boundary that separates the 3D space as either inside or outside the object. The methods of [22], [23] learn a function of SDF that specifies not only whether 3D points are inside or outside the object but also how far they are from the

object surface. These methods typically use MLPs as the models of the implicit field function, and learn latent shape encodings of the input RGB images via CNNs. Owning to the continuity of implicit fields, surface reconstructions from these methods are often smooth and can ideally achieve infinite resolution. However, due to the lack of explicit constraints on what topological structure an implicit field can learn, these methods are not guaranteed to correctly recover the surface topology of an object in an input image. Our proposed use of the topology-preserved skeleton as a structural regularization is indeed intended to address this limitation.

**The Skeleton as an Interior Shape Representation** The medial axis transform (MAT) [52] is an intrinsic shape representation that includes the medial axis together with the associated radius function of the maximally inscribed spheres, which can be used to reconstruct the original shape. The MAT has direct access to both a shape’s boundary and its interior, enabling various downstream applications regarding shape abstraction, classification and segmentation. Here, we refer the reader to [53], which presents a systematic discussion on the applications of MAT. To drive these applications, many methods [54], [55], [56] are designed to compute the MAT for downstream applications such as shape analysis, modeling, and editing. For example, Q-MAT [57] and Q-MAT+ [58] define novel quadratic errors to conduct MAT mesh simplification so that they can compute geometrically accurate, structurally simple, and compact meshes, which are employed either as supervision to constrain the MAT learning from point clouds [59] or as input to extract structural features for 3D object recognition [60]. But in reality, clean 3D mesh models from scanners are not always approachable, which leads to great difficulties in computing accurate MAT meshes. The skeletal shape representation, mostly as a compact point cloud, is considered as an approximation for medial axis transformation [52]. The works of [38], [61], [62], [63] explore meso-skeleton extraction techniques from point sets. A meso-skeleton consists of a mixture of curves and surface sheets adapted to the local 3D geometry, so it is appropriate for representing the topological structures of general 3D shapes. Wu et al. [39] apply the meso-skeleton to surface reconstruction from imperfect point clouds. They extract a meso-skeleton to capture contextual geometric information, which guides the completion of missing regions on the surface. Recently, P2PNet [64] introduces a learning-based approach to transform surfaces into meso-skeletons.

In our work, our motivation is to use the topology-preserved, skeletal shape representation to assist the downstream task of object surface recovery from RGB images. To the best of our knowledge, we are the first to explore skeleton inference from RGB images. To achieve this, we first extract meso-skeletons for supervision based on the method of [39] and then design an end-to-end, trainable network architecture for skeleton learning.

### 3 OVERVIEW

Given an RGB image  $\mathbf{I}$  of an object as input, our goal is to learn a deep model from training data, such that the object surface can be recovered as a triangle mesh  $\mathcal{M} = \{\mathcal{T}, \mathcal{E}, \mathcal{F}\}$ , where  $\mathcal{T}$  denotes the set of mesh vertices,  $\mathcal{E}$  for the edges connecting adjacent vertices, and  $\mathcal{F}$  for the faces defined by three adjacent vertices and their connected edges. The object surface to be recovered could have high algebraic and/or topological complexities, which pose great challenges to existing methods [9], [11], [12], [13], [14], [15], [20], [21].

To address the challenges, our key idea in this work is to leverage the *meso-skeleton*<sup>1</sup> of object surface either as a bridge or as a constraint, such that a better surface recovery could be subsequently achieved. Similar to object surface, a skeleton is usually represented as a point set (cf. examples in Fig. 7). Given a surface shape, skeletal points can be obtained by sinking the sampled surface vertices along the reverse directions of their normals. These points are considered as sampled ones from *skeletal curves* and *skeletal sheets*, which together define the meso-skeleton of object surface that has the nice properties of topology preservation and lower learning complexity. Our motivation to use skeleton to assist surface recovery is in fact based on these properties.

**Learning Skeletons from RGB Images** Our system of surface recovery starts from learning a skeleton from the input  $\mathbf{I}$ . While the skeleton is originally represented as a point set  $\mathcal{K} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^{n_K}$ , we propose a further learning to refine  $\mathcal{K}$  as a skeletal volume  $\mathbf{V} \in \mathbb{R}^{r \times r \times r}$ , as shown in Fig. 2. Learning  $\mathbf{V}$  via 3D CNNs is beneficial to establishing the connectivity among skeletal points, from which a subsequent meshing can also be achieved directly via marching cubes [34]. To learn  $\mathcal{K}$ , we present a novel, parallel design of *CurSkeDecoder* and *SurSkeDecoder* stacked on top of a shared CNN based image encoder; the CurSkeDecoder and SurSkeDecoder are designed to be respectively responsible for regression of points on skeletal curves and skeletal sheets. Given  $\mathcal{K}$ , we present an efficient scheme of globally guided subvolume synthesis to learn  $\mathbf{V}$ , which is essential to produce a high-resolution skeletal volume (e.g.,  $|\mathbf{V}| = 256^3$ ). Connecting the above two modules is a differentiable *Point2Voxel* layer, which makes an end-to-end model of our proposed SkeletonNet, as illustrated in Fig. 2. We present specifics of SkeletonNet in Sec. 4.

**Skeleton-guided Surface Learning** As an interior representation of object surface, the skeleton  $\mathbf{V}$  (and  $\mathcal{K}$ ) preserves the surface topology, and is potentially helpful for learning to recover  $\mathcal{M}$ . In this work, we study how to learn  $\mathcal{M}$  either *explicitly* from the skeletal volume  $\mathbf{V}$ , or *implicitly* by using features of  $\mathbf{V}$  to regularize the learning of a surface field. For the intended explicit surface learning, we present a design of graph convolutional network that produces  $\mathcal{M}$  by learning vertex deformation of an initial mesh  $\mathcal{M}_V$  extracted from  $\mathbf{V}$ ; Fig. 3 gives the illustration. For learning of an implicit surface field, we improve upon recent methods [23] by augmenting their image-level features with multi-scale features from the skeletal volume  $\mathbf{V}$ , such that a probability field of occupancy [20], [21] can be better learned, from which  $\mathcal{M}$  can be recovered via marching cubes [34]; Fig.

1. The skeletal shape representation is a kind of approximation for the medial axis transform (MAT) [52]. While the MAT of a 2D shape is a 1D skeleton, for a 3D model, the MAT is generally composed of 2D surface sheets. The skeleton composed of skeletal curves and skeletal sheets (i.e., medial axes) is generally called meso-skeleton [38], shortened as skeleton.

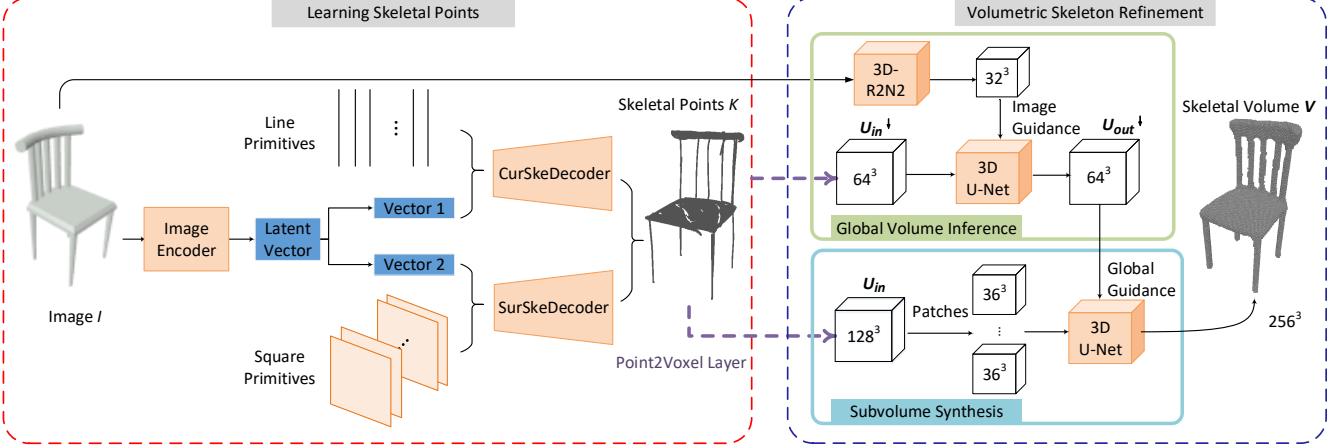


Fig. 2: The pipeline of our proposed *SkeletonNet*. The module in the red dotted box (left) stacks the parallel *CurSkeDecoder* and *SurSkeDecoder* on top of a CNN based image encoder; given an input image  $\mathbf{I}$ , the module is trained to regress a skeletal point set  $\mathcal{K}$ . The module in the blue dotted box (right) consists of two parallel streams of 3D CNNs; the top one is for a global, low-resolution volume synthesis, which guides the synthesis of a high-resolution skeletal volume  $\mathbf{V}$  via a sliding subvolume fashion in the bottom stream. Connecting the two modules is a differentiable *Point2Voxel* layer that converts  $\mathcal{K}$  as the initial volume and makes the proposed *SkeletonNet* an end-to-end trainable network.

4 gives the illustration. We conduct thorough experiments that confirm the efficacy of our proposed *SkeletonNet* for learning surface recovery from RGB images.

## 4 THE PROPOSED SKELETONNET

We present in this section specifics of our proposed *SkeletonNet*. Given an input  $\mathbf{I}$ , *SkeletonNet* is trained in an end-to-end fashion to produce a high-resolution skeletal volume  $\mathbf{V} \in \mathbb{R}^{r \times r \times r}$  via an intermediate representation of skeletal point set  $\mathcal{K} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^{n_{\mathcal{K}}}$ . While both  $\mathcal{K}$  and  $\mathbf{V}$  are skeleton representations of the underlying surface for the object contained in  $\mathbf{I}$ , they have different properties: the point set representation is original and compact, and the volumetric representation is more regular and connected. Our way of learning  $\mathbf{V}$  via the intermediate  $\mathcal{K}$  enjoys both of their advantages by establishing the connectivity among skeletal points in  $\mathcal{K}$  to form skeletal curves and skeletal sheets in  $\mathbf{V}$ . We empirically find that the alternative ways of either learning  $\mathcal{K}$  alone or learning  $\mathbf{V}$  directly from  $\mathbf{I}$  are less effective to have high-quality skeletons.

**Image Encoder** The *SkeletonNet* starts with a CNN based image encoder, which produces a latent vector  $\mathbf{f}_I \in \mathbb{R}^m$  encoding surface shape of the object in  $\mathbf{I}$ .  $\mathbf{f}_I$  will be used as input of the module for learning  $\mathcal{K}$ , as described shortly.

### 4.1 Learning of Skeletal Points

Learning  $\mathcal{K}$  from the latent vector  $\mathbf{f}_I$  is a standard point set regression problem. For example, one may choose the technique in PointSetGen [14] to directly regress the coordinates of skeletal points in  $\mathcal{K}$ . However, we note that skeletal points are those sampled from their underlying skeletal curves and skeletal sheets, and as such, distribution of these points satisfies geometric constraints that prevent them from free positioning in the 3D space. To implement such constraints, we introduce a novel, parallel design of *CurSkeDecoder* and

*SurSkeDecoder*, which are respectively responsible for regression of skeletal points on 1D curves, denoted collectively as  $\mathcal{K}_{Cur}$ , and those on 2D sheets, denoted as  $\mathcal{K}_{Sur}$ .

Specifically, we define a set of 2D primitives of unit square  $[0, 1]^2$ . Except for  $f_I$ , *SurSkeDecoder* takes as input regularly sampled points from each of the unit squares, and learns to deform the sampled points such that they form points on skeletal sheets in the 3D space; in other words, *SurSkeDecoder* learns a function  $\psi : [0, 1]^2 \rightarrow \mathbb{R}^3$ . Similarly, we define a set of 1D primitives of line segment  $[0, 1]$ . *CurSkeDecoder* learns a function  $\phi : [0, 1] \times \mathbb{R}^3$  by sampling regular points from the segments, and then learning to deform the sampled points such that they form points on skeletal curves in the 3D space. This way of using multiple 1D or 2D primitives respectively as inputs of *CurSkeDecoder* or *SurSkeDecoder* is similar to the strategy in *AtlasNet* [12]. In this work, both *CurSkeDecoder*  $\phi(\cdot)$  and *SurSkeDecoder*  $\psi(\cdot)$  are implemented as multilayer perceptrons (MLPs), whose layer specifics are given in Sec. 7. Note that neighboring relations are fixed for connected points sampled from each 1D or 2D primitive. To maintain the spatial relations among neighboring points during training, we propose the following regularized loss function to learn *CurSkeDecoder*  $\phi(\cdot)$ .

$$L_{\phi} = D(\mathcal{K}_{Cur}, \mathcal{K}_{Cur}^*) + \alpha R_{laplacian}(\mathcal{K}_{Cur}), \quad (1)$$

with

$$D(\mathcal{K}_{Cur}, \mathcal{K}_{Cur}^*) = \sum_{\mathbf{p} \in \mathcal{K}_{Cur}} \min_{\mathbf{p}^* \in \mathcal{K}_{Cur}^*} \|\mathbf{p} - \mathbf{p}^*\|_2^2 + \sum_{\mathbf{p}^* \in \mathcal{K}_{Cur}^*} \min_{\mathbf{p} \in \mathcal{K}_{Cur}} \|\mathbf{p} - \mathbf{p}^*\|_2^2, \quad (2)$$

$$R_{laplacian}(\mathcal{K}_{Cur}) = \sum_{\mathbf{p} \in \mathcal{K}_{Cur}} \left\| \mathbf{p} - \frac{1}{|\mathcal{N}(\mathbf{p})|} \sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} \mathbf{p}' \right\|_2^2, \quad (3)$$

where  $\alpha$  is a penalty parameter, and  $\mathcal{K}_{Cur}^*$  denotes the ground-truth point set consisting of those on skeletal curves. We use Chamfer distance to define the loss term (2), and laplacian smoothness [65] to define the regularizer (3), where  $\mathcal{N}(p)$  contains the neighboring points of any  $p \in \mathcal{K}_{Cur}$  whose pre-images are connected to that of  $p$  on a 2D unit square. We similarly define the following loss to learn SurSkeDecoder  $\psi(\cdot)$

$$L_\psi = D(\mathcal{K}_{Sur}, \mathcal{K}_{Sur}^*) + \alpha R_{laplacian}(\mathcal{K}_{Sur}), \quad (4)$$

where  $D(\mathcal{K}_{Sur}, \mathcal{K}_{Sur}^*)$  and  $R_{laplacian}(\mathcal{K}_{Sur})$  are similarly defined as (2) and (3). We note that  $\mathcal{K}^* = \mathcal{K}_{Cur}^* \cup \mathcal{K}_{Sur}^*$  and  $\mathcal{K}_{Cur}^* \cap \mathcal{K}_{Sur}^* = \emptyset$ . To prepare  $\mathcal{K}_{Cur}^*$  and  $\mathcal{K}_{Sur}^*$ , we present the ShapeNet-Skeleton dataset that contains the annotations of  $\mathcal{K}_{Cur}^*$  and  $\mathcal{K}_{Sur}^*$  for each object instance in ShapeNet [40].

**The ShapeNet-Skeleton Dataset** To facilitate training of CurSkeDecoder and SurSkeDecoder, we prepare training data of skeletal points for object instances in ShapeNet [40] as follows: 1) for each CAD model in ShapeNet, we convert it into a point cloud; 2) we extract the point set  $\mathcal{K}^*$  of meso-skeleton using the method [39]; 3) to have  $\mathcal{K}_{Cur}^*$  and  $\mathcal{K}_{Sur}^*$ , we classify each  $p^* \in \mathcal{K}^*$  as either belonging to skeletal curves or belonging to skeletal sheets based on principal component analysis of its neighboring points. Specifically, a point that has only one dominating eigenvalue is classified as that of skeletal curves, and as that of skeletal sheets otherwise. We have made this dataset publicly available at <https://github.com/tangjiapeng/SkeletonNet>. Given the ground truth  $\mathcal{K}^*$  of an object instance, we also prepare its volumetric version  $\mathbf{V}^*$  by first quantizing the skeletal points in  $\mathcal{K}^*$  as voxels and then conducting the operation of interior filling to have a solid volume.

## 4.2 Volumetric Skeleton Refinement

Given  $\mathcal{K}$ , SkeletonNet further learns a high-resolution skeletal volume  $\mathbf{V} \in \mathbb{R}^{r \times r \times r}$  as the final output, given the ground-truth skeletal volume  $\mathbf{V}^* \in \mathbb{R}^{r \times r \times r}$  whose preparation has been described in Sec. 4.1. This is efficiently achieved by a proposed module of *globally guided subvolume synthesis*, similar to the scheme used in [66]. The obtained  $\mathbf{V}$  will be subsequently used to recover the surface  $\mathcal{M}$  either explicitly or implicitly, as respectively to be presented in Sec. 5 and Sec. 6.

Specifically, the proposed module is composed of two parallel streams of volume synthesis (cf. Fig. 2). To prepare the module inputs, we first convert  $\mathcal{K}$  as a volume  $\mathbf{U}_{in} \in \mathbb{R}^{r' \times r' \times r'}$ , with  $r' < r$ ; we then downsample  $\mathbf{U}_{in}$  to have  $\mathbf{U}_{in}^\downarrow \in \mathbb{R}^{r'/2 \times r'/2 \times r'/2}$ . To make the conversion from  $\mathcal{K}$  to  $\mathbf{U}_{in}$  differentiable, we present a layer of *Point2Voxel* as described shortly in Sec. 4.3. The top, global stream learns to refine the input  $\mathbf{U}_{in}^\downarrow$  to have the output  $\mathbf{U}_{out}^\downarrow \in \mathbb{R}^{r'/2 \times r'/2 \times r'/2}$ , given the ground-truth  $\mathbf{U}^{\downarrow*}$  which is also obtained from  $\mathbf{V}^*$ ; note that  $\mathbf{U}_{out}^\downarrow$  has the same resolution as that of  $\mathbf{U}_{in}^\downarrow$ . The bottom, local stream uniformly samples overlapped subvolumes from  $\mathbf{U}_{in}^\downarrow$  and learns to super-resolve them to have their upsampled versions; for each sampled subvolume  $\mathbf{P}_{in} \in \mathbb{R}^{s' \times s' \times s'}$ , the bottom stream learns to generate the upsampled  $\mathbf{P}_{out} \in \mathbb{R}^{s \times s \times s}$ , given the ground-truth  $\mathbf{P}^*$  sampled from the corresponding 3D

position of  $\mathbf{V}^*$ , and the global  $\mathbf{U}_{out} \in \mathbb{R}^{r \times r \times r}$  (i.e.,  $\mathbf{V}$ ) is obtained by averaging these locally upsampled subvolumes at overlapping voxels. To implement the globally guided subvolume synthesis, we first use an independent auto-encoder (e.g., 3D-R2N2 [9]) that learns to map the input image  $\mathbf{I}$  as a feature volume of the size  $r'/4 \times r'/4 \times r'/4$ , which is used as the augmented input of the top, global stream; we then use the global refinement of  $\mathbf{U}_{out}^\downarrow$  as the concatenated input of the bottom, local stream, when it processes each sampled subvolume  $\mathbf{P}_{in}$ . These heuristics aim to both correct the possible errors in the predicted  $\mathcal{K}$ , and improve the consistency among synthesized subvolumes in  $\mathbf{V}$ ; we empirically find that they are effective for generation of a high-quality  $\mathbf{V}$ .

In this work, we implement both the global and local streams based on the 3D U-Net architecture [67], whose specifics are given in Sec. 7. We set  $r' = r/2$ ,  $s' = r'/4 + 4$ , and  $s = 2s'$ , which, when  $r = 256$ , gives  $|\mathbf{U}_{in}^\downarrow| = |\mathbf{U}_{out}^\downarrow| = 64^3$ ,  $|\mathbf{P}_{in}| = 36^3$ ,  $|\mathbf{P}_{out}| = 72^3$ ,  $|\mathbf{U}_{in}| = 128^3$ , and  $|\mathbf{U}_{out}| = |\mathbf{V}| = 256^3$ . To train the module, we use a per-voxel binary cross-entropy loss defined as

$$L_{refine} = \sum_{i \in \{j | j \in [1:r]^3\}} \mathbf{V}^*(\mathbf{i}) \log \mathbf{V}(\mathbf{i}) + (1 - \mathbf{V}^*(\mathbf{i})) \log(1 - \mathbf{V}(\mathbf{i})). \quad (5)$$

In practice, we first train the top, global stream and the bottom, local stream separately, and then train them together in a joint optimization manner.

**Remarks** We emphasize that our use of the globally guided subvolume synthesis is both effective and efficient for the generation of high-quality skeletal volumes. Alternative solutions include (1) directly quantizing the skeletal points in  $\mathcal{K}$  as voxels, which is, however, not guaranteed to connect the quantized voxels as skeletal curves and skeletal sheets in the 3D volumetric space, and may lose the chance to correct the possible errors in the predicted  $\mathcal{K}$ ; (2) predicting  $\mathbf{V}$  directly from  $\mathcal{K}$  via a 3D CNN, which is, however, computationally too expensive to generate a high-resolution  $\mathbf{V}$ . We present empirical studies that show the advantages of our used module over these alternatives.

## 4.3 End-to-End Training

The key to make our SkeletonNet an end-to-end trainable system is a differentiable Point2Voxel layer, which converts  $\mathcal{K}$  as an input of skeletal volume synthesis.

**A Point2Voxel layer** To assign a value to any voxel  $\mathbf{U}_{in}(\mathbf{i})$ , we use the following soft scheme

$$\mathbf{U}_{in}(\mathbf{i}) = \exp \left( -M \min_{\mathbf{p} \in \mathcal{K}} \|\mathbf{c}(\mathbf{i}) - \mathbf{p}\|_2^2 \right), \quad (6)$$

where  $M$  is a scaling constant and  $\mathbf{c}(\mathbf{i}) \in \mathbb{R}^3$  denotes coordinates of the voxel center. Consider a large scaling  $M$ ; for voxels whose spatial locations are closest to any skeletal points in  $\mathcal{K}$ , the scheme (6) assigns values approaching to 1, and for those far away, the assigned values decay rapidly to 0. To improve the practical efficiency, we only consider those voxels whose center points fall in a small neighborhood of any  $\mathbf{p} \in \mathcal{K}$ , while directly nulling other voxels.

Combining the loss terms (1), (4), and (5) gives the overall training objective of our proposed SkeletonNet

$$L_{\text{SkeletonNet}} = L_\phi(\mathcal{K}_{\text{Cur}}) + L_\psi(\mathcal{K}_{\text{Sur}}) + \beta L_{\text{refine}}(\mathbf{V}; \mathcal{K}), \quad (7)$$

where  $\beta$  is a balancing parameter.

## 5 AN EXPLICIT LEARNING OF MESH RECOVERY FROM SKELETON

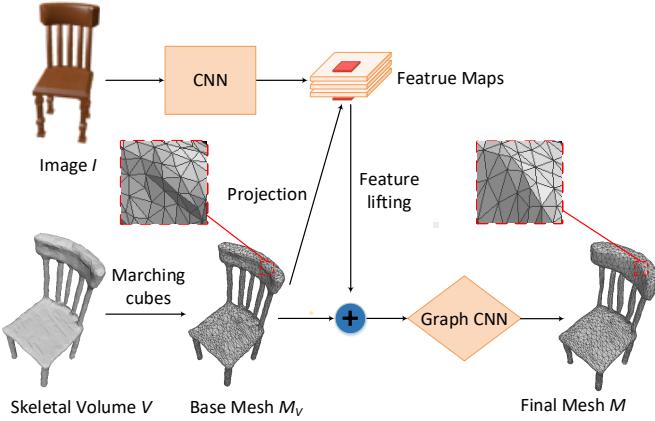


Fig. 3: The pipeline of our proposed Skeleton-Based Graph CNN (SkeGCNN) for explicit mesh recovery from an input image.

In this section, we study how to explicitly learn a recovery of surface mesh from the skeletal volume  $\mathbf{V}$  produced by SkeletonNet. This is possible considering the property that a surface mesh and its corresponding skeleton share the same topological structure. Technically, we consider extraction of an initial mesh  $\mathcal{M}_V$  from  $\mathbf{V}$ , and then learning to deform the vertices of  $\mathcal{M}_V$  such that  $\mathcal{M}_V$  is inflated to fit the object surface. We present specific steps of the process as follows.

**Extraction of the Initial Mesh** We use marching cubes [34] to produce the initial  $\mathcal{M}_V$  from  $\mathbf{V}$ . Since the volume  $\mathbf{V}$  is in high resolution,  $\mathcal{M}_V$  would contain a large number of vertices and faces. This makes the subsequent mesh inflation possible to fit an object surface that has complex surface geometries.

**Mesh Deformation using Graph CNNs** The mesh  $\mathcal{M}_V = \{\mathcal{T}_V, \mathcal{E}_V, \mathcal{F}_V\}$  itself has a graph data structure; we thus choose to use a graph CNN (GCNN) to learn the deformations of its vertices in  $\mathcal{T}_V$ . For any  $t \in \mathcal{T}_V$ , denote its associated feature vector at layer  $l$  of the GCNN as  $\mathbf{f}_t^l$ , and  $\mathcal{N}(t)$  be its local neighborhood; we simply compute its features at layer  $l+1$  as

$$\mathbf{f}_t^{l+1} = w_t^l \mathbf{f}_t^l + \sum_{t' \in \mathcal{N}(t)} w_{t'}^l \mathbf{f}_{t'}^l, \quad (8)$$

where  $w_t^l$  and  $\{w_{t'}^l\}$  are the network weights. Assuming that the network has  $L$  layers, it is trained to output the coordinate offsets  $\delta t \in \mathbb{R}^3$  for each  $t$  at the last layer, i.e.,  $\mathbf{f}_t^L = \delta t$ . To enhance the deformation learning, we learn feature maps from the input  $\mathbf{I}$  using image encoder, and lift the learned pixel-wise features to each vertex  $t \in \mathcal{T}_V$ ;

we concatenate the lifted features with vertex coordinates to form the input features of GCNN. For feature lifting, we use the camera pose prediction model of DISN [23] to estimate camera pose, and use the estimated camera pose to project each vertex  $t \in \mathcal{T}_V$  to the 2D image domain. We term such a GCNN based approach for deforming vertices of an initial mesh from skeleton input as Skeleton GCNN (SkeGCNN); Fig. 3 gives the illustration.

**Loss Functions and Network Training** Training of SkeGCNN is to learn network weights that drive the deformations  $\Delta \mathcal{T}_V = \{\delta t | t \in \mathcal{T}_V\}$ , such that the resulting vertex set  $\mathcal{T} = \{\delta t + t | t \in \mathcal{T}_V\}$ , together with the vertex connections in  $\{\mathcal{E}, \mathcal{F}\}$ , defines a mesh prediction  $\mathcal{M}$  that is close to the ground-truth mesh  $\mathcal{M}^*$  under a certain measure of distance. Mesh distance is usually approximated as the distance between point sets sampled from the respective meshes. It is thus straightforward to use the Chamfer distance as the measure, similar to (2). In this work, we use a weighted version of Chamfer distance in order to emphasize the learning more on surface regions whose geometries have higher complexities; we technically achieve this by assigning larger weights to points sampled from the high-curvature regions on  $\mathcal{M}^*$ , giving rise to

$$\tilde{D}(\Delta \mathcal{T}_V; \mathcal{M}, \mathcal{M}^*) = \sum_{q \in \mathcal{M}} \min_{q^* \in \mathcal{M}^*} \kappa_{q^*} \|q(\Delta \mathcal{T}_V) - q^*\|_2^2 + \sum_{q^* \in \mathcal{M}^*} \kappa_{q^*} \min_{q \in \mathcal{M}} \|q(\Delta \mathcal{T}_V) - q^*\|_2^2, \quad (9)$$

where  $\kappa_{q^*}$  denotes the weight assigned to a point  $q^*$  sampled from  $\mathcal{M}^*$ , and  $q(\Delta \mathcal{T}_V)$  is a function of vertex deformations. We set  $\kappa_{q^*}$  empirically by thresholding the largest angle variation of normals associated with points in the local neighborhood of  $q^*$ . Note that a point  $q \in \mathcal{M}$  would distribute its supervision signal to the three vertices defining the mesh face where it is sampled. We practically implement (9) by sampling a fixed number of points respectively from  $\mathcal{M}$  and  $\mathcal{M}^*$ . Except for fitting the predicted  $\mathcal{M}$  to  $\mathcal{M}^*$ , we also impose regularization on the learning of SkeGCNN, in order for the learned model to generalize to testing instances. The regularization is mainly to promote the smoothness of resulting meshes. We practically follow [15] and use a first regularizer that constrains the lengths of edges in  $\mathcal{E}$ , and a second one that constrains the normal variations for points in local neighborhoods (i.e., promotion of small local curvatures). Our regularized SkeGCNN learning objective is as follows

$$L_{\text{SkeGCNN}} = \tilde{D}(\Delta \mathcal{T}_V; \mathcal{M}, \mathcal{M}^*) + \lambda_1 R_{\text{edge}}(\Delta \mathcal{T}_V; \mathcal{E}) + \lambda_2 R_{\text{curvature}}(\Delta \mathcal{T}_V; \mathcal{M}), \quad (10)$$

where  $\lambda_1$  and  $\lambda_2$  are penalty parameters. One may refer to [15] for definitions of the two regularizers.

## 6 DEEP IMPLICIT SURFACE FIELD LEARNING WITH REGULARIZATION OF SKELETON

In this section, we present how to use the skeletal volume  $\mathbf{V}$  produced by SkeletonNet to regularize the learning of

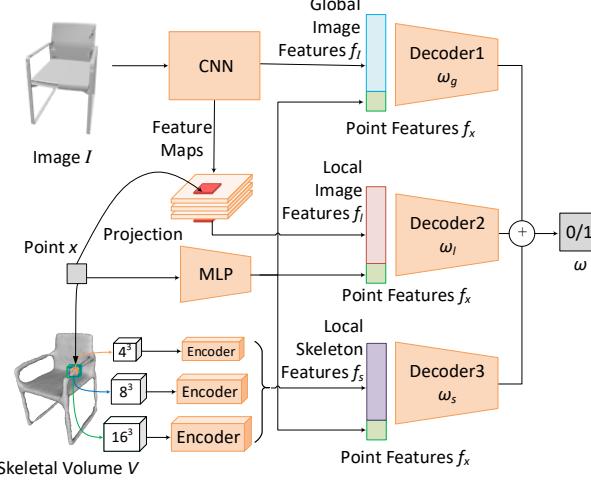


Fig. 4: An illustration of our proposed Skeleton-Regularized Deep Implicit Surface Network (SkeDISN).

an implicit surface field. We first introduce the necessary background.

**Deep Implicit Field Learning** Except for explicit representation, an object surface can also be represented implicitly as a field function  $\omega : \mathbb{R}^3 \rightarrow [0, 1]$  of occupancy probability, which indicates the surface boundary by classifying any  $x \in \mathbb{R}^3$  as either inside or outside the object, or alternatively as a signed distance function  $\omega : \mathbb{R}^3 \rightarrow (-\infty, \infty)$  whose zero-level isosurface is the exact object surface. Recent methods [20], [21], [22] propose to learn a deep network (typically an MLP) as the implicit function of object surface, such that it can output field values given sampled input points in the 3D space; the obtained field values at sampled 3D points would enable mesh recovery via a final step of marching cubes [34]. They apply deep implicit field learning to the task setting of RGB-to-mesh recovery, where a CNN based image encoder is typically used to learn latent shape code from the input image, which is used together with coordinates of points in the 3D space as the inputs of MLP based field function. The framework is improved in [23] by feeding extra, pixel-wise features extracted from the learned feature maps of the input image.

**Skeleton-Regularized Learning** In this work, we are interested in investigating whether the skeletal volume  $V$  given by SkeletonNet would provide additional benefits to the learning of implicit surface function. To this end, we adopt a network architecture in Fig. 4, which largely follows [23] except the skeleton part. We term the proposed model as Skeleton-Regularized Deep Implicit Surface Network (SkeDISN). The network is composed of three parallel streams, which for any input of sampled  $x \in \mathbb{R}^3$ , output 2-dimensional in-out probability vectors of the same format; they also share a same MLP that learns a feature vector  $f_x$  from coordinates of the input  $x$ . The first stream learns a field function  $\omega_g : \mathbb{R}^3 \rightarrow [0, 1]^2$  that has an additional input of latent vector  $f_I$ , encoding the global surface shape of the object in the input image  $I$ . The second stream extracts pixel-wise features  $f_I$  from feature maps of the image encoder, for which estimated camera pose is used to project

the 3D point  $x$  to the image domain, and concatenates the extracted features with  $f_x$  to form the input of a field function  $\omega_l : \mathbb{R}^3 \rightarrow [0, 1]^2$ . The last stream implements learning from the skeletal volume  $V$ ; for the sampled  $x$ , it extracts from  $V$  multi-scale subvolumes centered at  $x$ , and uses 3D CNNs to learn their respective feature vectors; it concatenates the learned feature vectors to form  $f_s$ , which is then concatenated with  $f_x$  to form the input of a field function  $\omega_s : \mathbb{R}^3 \rightarrow [0, 1]^2$ . The network finally sums up the three probability vectors as  $\omega_g(x) + \omega_l(x) + \omega_s(x)$ , and feeds it into a softmax layer to get the final 0/1 classification. All the three field functions are implemented as MLPs. With a slight abuse of notation, we write the overall field function of occupancy probability implemented by SkeDISN as  $\omega : \mathbb{R}^3 \rightarrow [0, 1]^2$ .

**Network Training and Inference** Given ground-truth object meshes, we use the following objective to train SkeDISN

$$L_{SkeDISN} = \sum_{x \in \mathbb{R}^3} \begin{cases} CE(\omega(x), \mathbb{I}_{M^*}(x)) & \text{if } d_{M^*}(x) < \varepsilon, \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

where  $CE(\cdot, \cdot)$  stands for a two-way cross-entropy loss,  $\mathbb{I}_{M^*}(x) \in [0, 1]^2$  is an indicator whose two entries are set as either 1 or 0 depending on whether  $x$  is inside or outside  $M^*$  in the 3D space, and  $d_{M^*}(x)$  represents the distance between  $x$  and its projection onto  $M^*$ ; by thresholding the distance at a small  $\varepsilon$ , the loss (11) is activated only when the sampled points are close to the object surface. Given a trained SkeDISN, one would sample points in the 3D space and forward-propagate them through the network to get their in-out classification results; the mesh recovery can be finally obtained via marching cubes.

## 7 EXPERIMENTS

**Dataset** Our training dataset is mainly from ShapeNet [40], where we collect 43,784 3D shapes in 13 categories. The dataset is split into two parts, 80% shapes are used for training and the remaining for testing. For each 3D model, we rendered 24 RGB images of size 224 \* 224 under different viewpoints, forming the image-model pairs. The shape categories include: Plane, Bench, Cabinet, Car, Chair, Monitor, Lamp, Speaker, Firearm, Couch, Table, Cellphone, Watercraft. Note that, as all existing approaches, the models in our dataset are aligned in a unified object-centric coordinate frame.

**Model Parameters and Implementation Details** For SkeletonNet, we use ResNet18 [28] as the image encoder, which produces 512-dimensional feature vectors as the latent shape encoding. Both the MLPs of CurSkeDecoder and SurSkeDecoder stack 4 fully-connected (FC) layers, whose numbers of output neurons are respectively 512, 256, 128 and 3, where ReLU activation is used for the first 3 layers and tanh activation is used for the last layer. To generate points of skeletal curves in each  $\mathcal{K}_{Cur}$ , we use 20 1D primitives of the segment  $[0, 1]$  as the input of CurSkeDecoder; we also use 20 2D primitives of the unit square  $[0, 1]^2$  as the input of SurSkeDecoder, for generation of points in each  $\mathcal{K}_{Sur}$ . We set the penalty parameter in (1) and (4) as  $\alpha = 0.2$ , and that in (7) as  $\beta = 1$ . Module specifics for volumetric

skeleton refinement are as follows. Implementations of both the global and local streams are based on 3D U-Net [67]; that for the global stream consists of 4 convolution layers with the respective numbers of channels as 32, 64, 128, and 128, and 4 deconvolution layers with the respective numbers of channels as 128, 64, 32, and 2; for local stream, the 3D U-Net consists of 4 convolution layers with the respective numbers of channels as 32, 64, 128, and 128, and 5 deconvolution layers with the respective numbers of channels as 128, 64, 32, 16, and 2; all the convolution/deconvolution layers are with kernels of size 3 and stride 2. For the layer of Point2Voxel, we set the scaling constant in (6) as  $M = 10$ . We implement the end-to-end training of SkeletonNet as follows. We first train CurSkeDecoder and SurSkeDecoder separately, for which we use a learning rate of 1e-3 for 150 epochs, and then train the module for volumetric refinement using a learning rate of 1e-4 for 20 epochs; we finally fine-tune the entire SkeletonNet with a learning rate of 1e-5 for 20 epochs.

Our SkeGCNN for explicit mesh recovery consists of 6 graph convolution layers, each of which learns weights to perform feature learning, according to (8). The first layer receives a pixel-wise feature vector of dimension 963; all the hidden layers work with feature vectors of dimension 192; the last layer outputs 3-dimensional coordinate offsets. We use VGG-16 [25] as the image encoder of SkeGCNN, and project pixel-wise features at ‘conv4’, ‘conv7’, ‘conv10’, and ‘conv13’ of VGG-16 to vertices of the initial mesh. To set values of  $\kappa_{q^*}$  in (9), we construct the local neighborhood of size 16 around  $q^*$ , and set  $\kappa_{q^*} = 5$  when the largest angle variation associated with points in the neighborhood is beyond  $60^\circ$ , and set  $\kappa_{q^*} = 1$  otherwise. We set the penalty parameters  $\lambda_1 = 7e^{-1}$  and  $\lambda_2 = 3e^{-4}$  in the SkeGCNN training objective (10). SkeGCNN is trained with an initial learning rate of 1e-4, which is decayed by a factor of 10 every 20 epochs until a total of 60 epochs.

Our SkeDISN uses the same image encoder as in SkeGCNN. The MLP of feature embedding for coordinates of sampled points consists of three FC layers respectively of 64, 128, and 512 neurons. Each of the MLPs implementing the three parallel field functions  $\omega_g$ ,  $\omega_l$ , and  $\omega_s$  consists of three FC layers respectively of 512, 256, and 2 neurons, except for their respective input layers. For use of features in the input skeletal volume, we extract its respective subvolumes of sizes  $4^3$ ,  $8^3$ , and  $16^3$  centered at each sampled point. We set the threshold  $\varepsilon = 0.1$  in the SkeDISN training objective (11). SkeDISN is trained with an initial learning rate of 1e-4, which decays at a factor of 0.9 every 2 epochs until a total of 60 training epochs.

**Evaluation Metrics** We use Chamfer distance (CD) and intersection-over-union (IoU) as the primary metrics for evaluation of mesh recoveries from RGB images. For computation of CD, we do mesh2point conversion by uniformly sampling 10,000 points on each surface mesh. For computation of IoU, we follow [23] and do voxelization in a grid of  $64^3$ . To evaluate results of skeletal point sets and skeletal volumes given by SkeletonNet, we use CD and IoU as the metrics as well.

Variant method	CD ( $\times 0.001$ )	IoU ( $\times 100$ )
Point-wise fitting	2.218	28.97
Line-wise fitting	2.242	35.48
Square-wise fitting	2.016	32.81
Shared line-and-square fitting	2.451	33.10
SkeletonNet w/o Laplacian regu.	1.911	31.10
SkeletonNet w/o end-to-end training	1.601	37.73
SkeletonNet	<b>1.485</b>	<b>39.50</b>

TABLE 1: Quantitative comparisons between the module of SkeletonNet and its variants for producing the intermediate results of skeletal point set from input RGB images. Results are obtained on the ShapeNet *chair* category. Please refer to the main text for specific settings of these variants.

Variant method	IoU ( $\times 100$ )
Quantization and morphological dilation	27.52
SkeletonNet via subvolume synthesis alone	37.64
SkeletonNet via globally guided subvol. synthesis	<b>39.46</b>

TABLE 2: Quantitative comparisons between the SkeletonNet module of volumetric refinement and its variants. Results are obtained on ShapeNet *chair* category. Please refer to the main text for specific settings of these variants.

## 7.1 Evaluation of the Proposed SkeletonNet

In this section, we evaluate the efficacy of our proposed SkeletonNet for learning skeletal point sets and skeletal volumes from the input RGB images, by comparing with alternative designs. We intensively use the ShapeNet category of *chair* for these comparisons, and give the summarized results on other categories.

We first evaluate the SkeletonNet module that produces an intermediate result of skeletal point set  $\mathcal{K}$  from an input image. The module uses CurSkeDecoder and SurSkeDecoder that respectively regress points on skeletal curves and skeletal sheets, where a regularizer of Laplacian smoothness is used to preserve the neighboring relations of points connected on the input 1D primitives of line or 2D primitives of square. The efficacy of such a design is verified by comparing with 1) “point-wise fitting” that directly adopts PSG [14] to regress the skeletal points, 2) “line-wise fitting” that removes the SurSkeDecoder and only deforms the input 1D primitives for regression of skeletal points, 3) “square-wise fitting” that removes the CurSkeDecoder and only deforms the input 2D primitives for regression of skeletal points, and 4) “shared line-and-square fitting” that uses a single MLP, instead of using the parallel CurSkeDecoder and SurSkeDecoder, to learn deformations of 1D lines and 2D squares. Note that end-to-end training is enabled in the above variants via our proposed point2voxel layer, and the Laplacian regularizer is also used in the last three of the above variants; to investigate the efficacy by their own, we further compare with 5) SkeletonNet whose training is not regularized by Laplacian smoothness, and 6) SkeletonNet whose first module for the intermediate skeletal point set and second module of volumetric refinement are separately trained. We make the comparisons based on two measures: 1) CD between the predicted  $\mathcal{K}$  and the ground-truth  $\mathcal{K}^*$ , and 2) IoU between the final output  $\mathbf{V}$  from each of

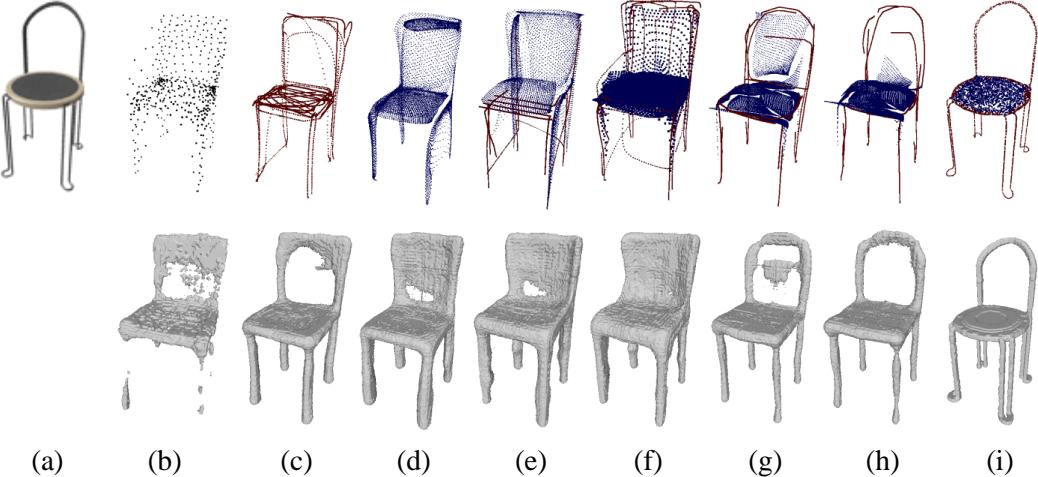


Fig. 5: Qualitative comparisons between SkeletonNet and its variants. These variants replace the module of SkeletonNet for producing the intermediate results of skeletal point set from input RGB images. (a) Input image; (b) point-wise fitting; (c) line-wise fitting (with Laplacian regularization); (d) square-wise fitting (with Laplacian regularization); (e) shared line-and-square fitting (with Laplacian regularization); (f) SkeletonNet without Laplacian regularization; (g) SkeletonNet; (h) Ground truth. Please refer to the main text for specific settings of these variants.

category	plane	bench	cabinet	car	chair	monitor	lamp
CD ( $\times 0.001$ )	1.153	1.245	1.901	0.918	1.473	1.879	3.357
IoU ( $\times 100$ )	38.49	36.72	55.00	67.68	38.78	35.82	35.12
category	speaker	firearm	couch	table	cellphone	watercraft	mean
CD ( $\times 0.001$ )	2.787	0.882	1.608	1.728	1.124	1.530	1.660
IoU ( $\times 100$ )	54.35	38.91	47.38	40.02	55.94	44.55	45.29

TABLE 3: Quantitative results of our SkeletonNet and its intermediate predictions of skeletal point set on all of 13 categories in ShapeNet [40]. Skeletal point set is measured by Chamfer distance (CD), and skeletal volume is measured by intersection-over-union (IoU).

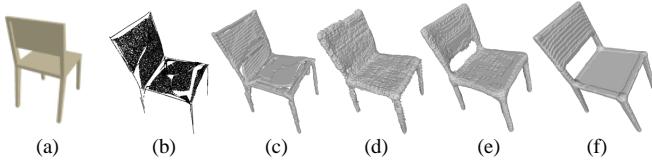


Fig. 6: Qualitative comparisons between the SkeletonNet module of volumetric refinement and its variants. (a) Input image; (b) the intermediate result of skeletal point set; (c) quantization and morphological dilation; (d) subvolume synthesis alone; (e) our used module of globally guided subvolume synthesis; (f) ground truth.

these variants and the ground-truth  $V^*$ , where the same SkeletonNet module of volumetric refinement is used for producing  $V$  from  $\mathcal{K}$ . Table 1 gives the comparative results, which are obtained on the ShapeNet *chair* category. Clearly, point-, line-, and square-wise deformations may not well fit the skeletal point sets, and shared line-and-square fitting is inferior to be responsible for regression of both skeletal curves and skeletal sheets. It is also observed in Table 1 that Laplacian regularization is important to constrain the deformations by connecting neighboring points on lines and

squares, and gives a high-quality final result of skeletal volume. The end-to-end training enabled by Point2Voxel layer gives rise to more accurate predictions of skeletal point cloud and volume. These observations are further demonstrated by examples of qualitative results in Fig. 5.

We further evaluate the SkeletonNet module of volumetric refinement, which produces the final output  $V$  from the intermediate  $\mathcal{K}$ . Given  $\mathcal{K}$ ,  $V$  can be alternatively obtained by 1) directly quantizing skeletal points in  $\mathcal{K}$  into voxels and performing morphological dilation, and 2) using subvolume synthesis of the module without global guidance. Quantitative results in Table. 2 confirm the efficacy of our used module. Examples of qualitative comparisons are shown in Fig. 6, where we observe that the global guidance is important to recover the topological structure that may have been lost in the intermediate result of skeletal point set.

We finally report in Table 3 our results of skeletal point set and skeletal volume on other categories of ShapeNet. These results tell that the difficulties of learning skeletons for different categories vary greatly, with cabinet, cellphone, and couch as the easier ones; this is reasonable since most of the instances from these categories have simple topologies. Example results are also shown in Fig. 7; complex topologies and thin structures, such as rods of chair, legs of table, and

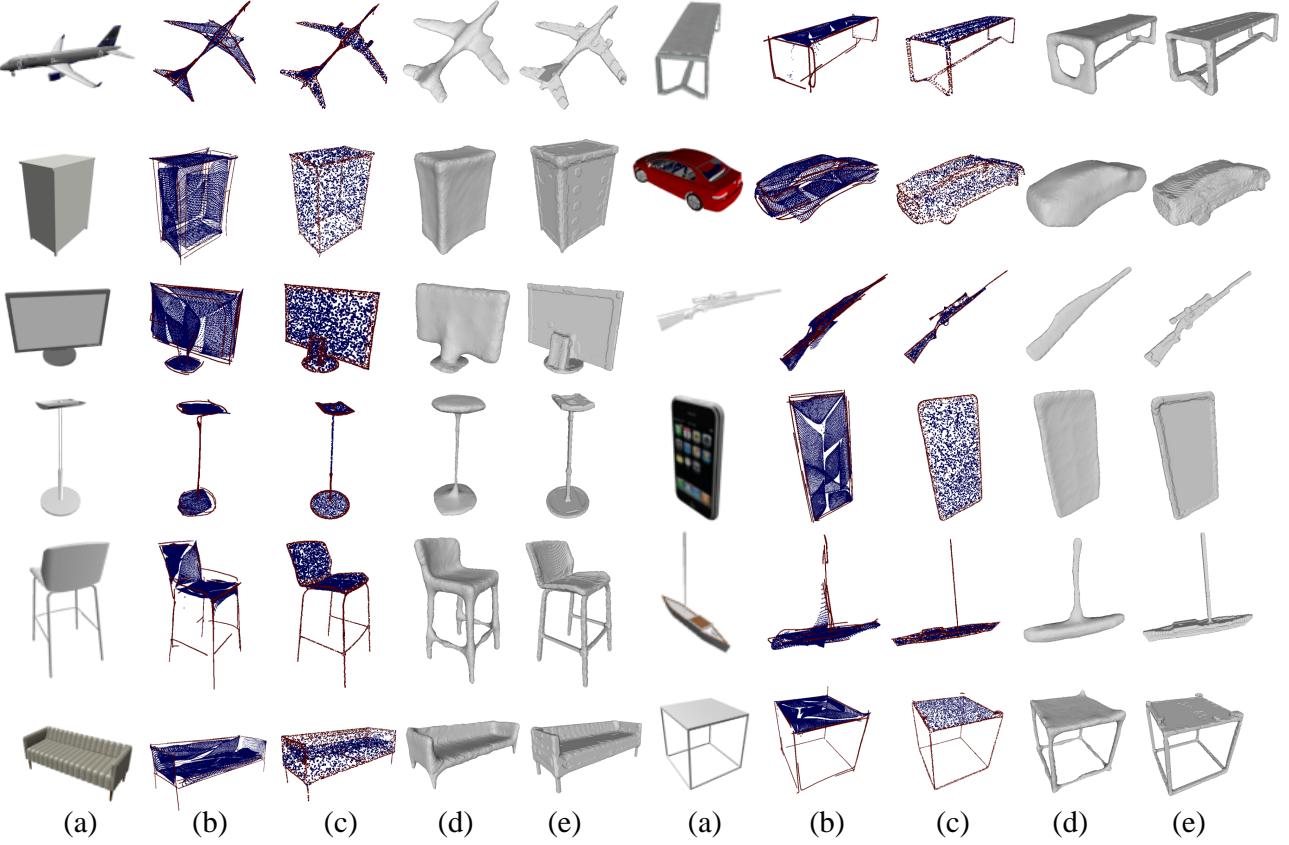


Fig. 7: Visualization results of our proposed SkeletonNet and its intermediate predictions of skeletal point set. (a) Input images; (b) The produced skeletal points; (c) The ground-truth skeletal points; (d) The refined skeletal volume; (e) The ground-truth skeletal volume.

holder of lamp, are successfully recovered in the resulting skeletons, which are expected to help the downstream task of mesh recovery.

## 7.2 Explicit and Implicit Mesh Recoveries from Skeletons

Skeleton is an interior representation of object surface that preserves the surface topology; in this section, we conduct experiments to study its benefits to recovery of surface mesh. The studies are conducted under the task setting of learning mesh recovery from a single RGB image. Specifically, for explicit mesh recovery, we use the skeletal volume produced by SkeletonNet as input of our SkeGCNN model introduced in Sec. 5; for learning an implicit surface field, we use the SkeDISN introduced in Sec. 6, which extracts features from the skeletal volume to regularize the field learning. We compare SkeGCNN and SkeDISN with the corresponding groups of state-of-the-art methods.

Learning a surface mesh explicitly via vertex deformation is pioneered by AtlasNet [12]; due to its use of multiple initial meshes, the method fails to produce a closed, watertight mesh. Pix2Mesh [15] resolves this issue by deforming a single mesh of fixed topology. TMNet [18] further improves the technique to generate object surface of complex topology, by designing a module of face pruning that is able

to dynamically adjust the topology during the deformation process. We quantitatively compare our SkeGCNN with these methods in Table 4 under the measure of CD and Table 5 under the measure of IoU, where results are obtained by 13 ShapeNet categories. We use the publicly released codes to implement these methods; for a fair comparison, their training hyper-parameters have been optimally tuned to the data used in our experiments. Under both the measures of CD and IoU, our results of deforming the input skeletons are significantly better than those of existing methods on most of the 13 categories, except the categories of *couch* and *cellphone*, whose topologies are relatively simple and for which TMNet achieves the best results. Examples of qualitative results in Fig. 8 tell that in terms of generating meshes of complex topologies, our method and TMNet are able to generate mesh results with some thin structures, and our results tend to be better than those of TMNet, while AtlasNet and Pix2Mesh fail to do so.

Mesh recovery via implicit field learning gains an increased recent popularity, with methods proposed for learning deep field functions whose outputs are in different formats. For example, IMNet [21] and OccNet [20] learn shape-decoding networks whose binary outputs indicate whether a sampled point in the 3D space is inside or outside the object surface; DISN [23] learns a network of signed distance function whose output represents the distance of

Category	Implicit					Explicit			
	OGN	IMNet	OccNet	DISN	SkeDISN	P2M	AtlasNet	TMNet	SkeGCNN
plane	4.765	1.929	1.967	1.674	<b>1.243</b>	2.816	1.733	1.459	<b>0.771</b>
bench	7.645	2.925	2.152	1.794	<b>0.944</b>	2.809	2.427	2.020	<b>1.037</b>
cabinet	2.531	2.705	1.773	2.177	<b>1.921</b>	2.463	2.362	1.857	<b>1.468</b>
car	1.129	1.704	1.354	0.962	<b>0.925</b>	2.880	3.426	1.692	<b>0.675</b>
chair	6.418	2.718	2.258	1.538	<b>1.248</b>	2.763	1.622	1.485	<b>1.138</b>
monitor	3.726	2.518	2.080	2.150	<b>1.084</b>	3.027	2.005	2.637	<b>1.316</b>
lamp	11.274	8.156	15.899	6.144	<b>3.958</b>	5.357	4.609	5.450	<b>2.540</b>
speaker	5.772	4.583	3.392	3.327	<b>2.582</b>	7.142	3.173	3.486	<b>2.446</b>
firearm	1.935	2.259	2.128	1.310	0.645	1.016	2.754	1.706	<b>0.685</b>
couch	4.488	2.574	1.755	1.850	<b>1.141</b>	2.482	2.038	<b>1.049</b>	1.256
table	7.178	4.239	2.739	3.127	<b>1.917</b>	5.470	2.246	2.540	<b>1.718</b>
cellphone	2.137	1.322	1.366	1.036	<b>0.839</b>	1.487	1.699	<b>1.088</b>	1.127
watercraft	3.941	3.819	3.319	3.165	<b>1.885</b>	3.049	2.428	2.318	<b>1.064</b>
mean	4.842	3.189	3.249	2.327	<b>1.564</b>	3.263	2.502	2.214	<b>1.326</b>

TABLE 4: Quantitative comparisons (Chamfer distance  $\times 0.001$ ) of our method against state-of-the-arts on ShapeNet dataset. The lower is better.

Category	Implicit					Explicit			
	OGN	IMNet	OccNet	DISN	SkeDISN	P2M	AtlasNet	TMNet	SkeGCNN
plane	50.44	58.33	53.45	65.01	<b>66.41</b>	31.43	53.73	59.06	<b>61.06</b>
bench	48.85	56.54	56.81	61.61	<b>64.30</b>	42.11	48.79	53.42	<b>58.31</b>
cabinet	61.98	57.63	61.92	64.45	<b>67.05</b>	58.43	41.23	52.31	<b>59.22</b>
car	81.50	78.57	79.22	83.56	<b>84.36</b>	55.25	51.15	72.38	<b>76.03</b>
chair	54.70	55.14	61.10	60.54	<b>64.97</b>	50.70	48.56	51.02	<b>61.71</b>
monitor	55.74	61.48	57.04	63.53	<b>67.37</b>	45.75	49.88	56.75	<b>57.07</b>
lamp	44.15	41.73	50.17	48.32	<b>53.39</b>	41.83	36.49	40.73	<b>52.24</b>
speaker	56.90	61.29	61.99	65.34	<b>70.00</b>	54.17	43.37	54.74	<b>60.25</b>
firearm	59.08	59.88	53.96	71.86	73.46	56.24	56.07	60.16	<b>65.76</b>
couch	63.95	66.13	71.09	70.88	<b>74.72</b>	59.70	49.17	58.76	<b>63.13</b>
table	55.36	50.77	59.78	50.44	<b>61.62</b>	40.99	46.52	47.92	<b>58.35</b>
cellphone	69.59	73.44	70.96	76.11	<b>75.55</b>	69.36	53.60	<b>72.18</b>	69.43
watercraft	55.95	61.78	58.47	71.86	<b>69.09</b>	44.52	56.12	58.44	<b>63.31</b>
mean	58.32	60.21	61.23	65.65	<b>68.64</b>	50.04	48.82	56.68	<b>62.46</b>

TABLE 5: Quantitative comparisons (Intersection over Union  $\times 100$ ) of our method against state-of-the-arts on ShapeNet dataset. The higher is better.

the sampled point to the underlying surface. Our SkeDISN largely follows DISN; we adapt its architecture a bit to output binary results of in-out classification, which is more compatible to include features from the input skeletal volume. As a baseline, we also compare with OGN [31], which is the state-of-the-art method to output octree-based surface volume. Again, these methods are implemented using their publicly released codes, with hyper-parameters optimally tuned for the data of our experiments. Quantitative results in Tables 4 and 5 confirm the efficacy of using skeleton features to regularize the implicit field learning, where results of our SkeDISN are significantly better than those of existing methods under both the measures of CD and IoU. Qualitative results in Fig. 8 suggest that DISN and our skeleton-regularized version are the better methods to reconstruct object meshes of complex topologies.

By comparing the results in Tables 4 and 5 between explicit and implicit methods, one may observe that there

exists no a generally good choice of current measure to quantitatively evaluate different methods from all perspectives; CD may better evaluate the algebraic quality of recovered meshes, for which explicit methods perform better, and IoU may better evaluate the topological quality of recovered meshes, for which implicit methods seem prevail. However, visual comparisons in Fig. 8 suggest that explicit methods still have their merits even in terms of recovering object meshes of complex topologies. These results suggest that further studies in future research are necessary to understand the respective pros and cons of explicit and implicit surface recovery methods.

**Ablation studies** Our proposed SkeGCNN and SkeDISN use as inputs the skeletal volumes produced by SkeletonNet. In fact, skeletal volumes may also be obtained by training a state-of-the-art volume synthesis method, e.g., OGN [11], using the ground-truth skeletal volumes in our ShapeNet-

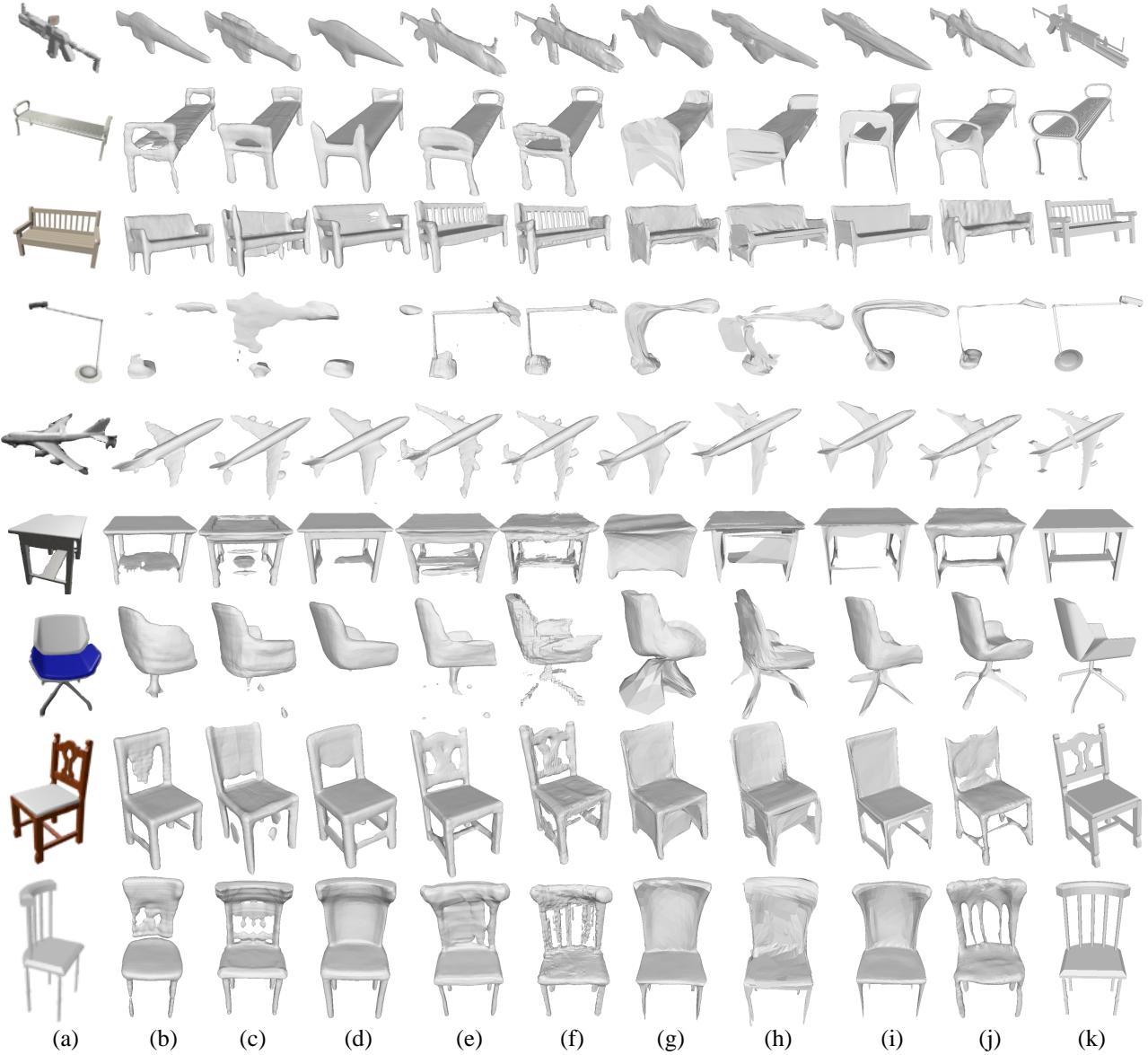


Fig. 8: (a) Input Images (b) OGN; (c) IMNet; (d) OccNet; (e) DISN; (f) SkeDISN; (g) Pixel2Mesh; (h) AtlasNet; (i) TMNet; (j) SkeGCNN; (k) Ground Truth.

Method	CD ( $\times 0.001$ )	IoU ( $\times 100$ )
OGN + SkeGCNN	4.485	49.84
SkeletonNet + SkeGCNN	<b>1.154</b>	<b>62.90</b>
OGN + SkeDISN	1.534	61.70
SkeletonNet + SkeDISN	<b>1.230</b>	<b>65.20</b>

TABLE 6: Ablation studies on the effectiveness of SkeletonNet for the downstream tasks of explicit and implicit mesh recoveries. Results are obtained on the ShapeNet *chair* category.

Skeleton dataset. We thus conduct ablation studies by replacing SkeletonNet with an OGN based volume synthesis network. We note that OGN is considered as an efficient method for synthesizing high-resolution volumes (e.g., volumes of  $256^3$ ). We conduct the ablation experiments on the

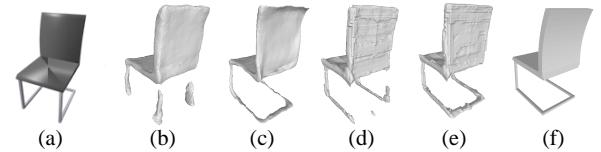


Fig. 9: Example results of the ablation studies on the effectiveness of SkeletonNet for the downstream tasks of explicit and implicit mesh recoveries. (a) Input image; (b) OGN + SkeGCNN; (c) SkeletonNet + SkeGCNN; (d) OGN + SkeDISN; (e) SkeletonNet + SkeDISN; (f) Ground truth.

ShapeNet category of *chair*. Quantitative results in Table 6 confirm that for both explicit and implicit mesh recoveries, our proposed SkeletonNet provides skeletal volumes that

are more effective for the downstream tasks. Our advantage can be attributed to the learning of the intermediate skeletal point sets. Example results in Fig. 9 reflect this advantage.

### 7.3 Additional Results

In this section, we present additional results, mostly in variant task settings, to show the general usefulness of our proposed methods.

**Interpolation of skeletons** SkeletonNet takes an RGB image of an object instance as input, and outputs its skeletons in both point set and volume based representations. In general, inputs of different instances correspond to different skeletal results. It is interesting to observe the results in between, i.e., the interpolations of skeletons between two results corresponding to two RGB inputs of different instances, possibly of different object categories. To this end, given two RGB inputs, we simply interpolate their global feature vectors from the image encoder, and then use the interpolated features for inference of skeletal point sets and the refined skeletal volumes. Example results in Fig. 10 show that, by using different weights of combination, the interpolated results demonstrate a gradual transition of plausible shapes from skeletons of *chair* to those of *table*.

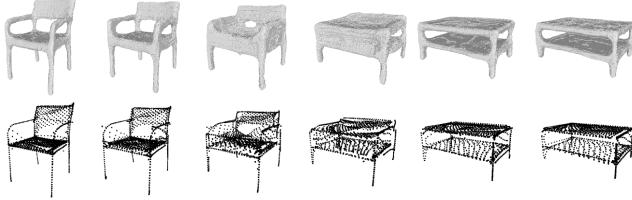


Fig. 10: Example results of skeleton interpolation in the forms of skeletal point set (bottom) and skeletal volume (top).

**Result gallery** In Fig. 11, we show additional results of our proposed SkeGCNN and SkeDISN respectively for explicit and implicit mesh recoveries. Generally speaking, the explicit method of SkeGCNN gives smoother mesh results, and the implicit method of SkeDISN gives results with more surface details, which may translate as improved quality for surface of complex topologies.

**Mesh Recovery from multi-view inputs** Our methods can be easily extended to take multiple RGB images of different views as inputs, and the results are expected to be improved. To achieve skeleton based mesh recovery from multi-view images, we modify SkeletonNet, SkeGCNN, and SkeDISN simply by conducting parallel feature encodings of the input views using their respective image encoders, and then aggregating the obtained global (and local) feature vectors via max pooling. Example results in Fig. 12 tell that for objects of complex topologies, using additional views help mesh recoveries of both explicit and implicit methods.

**Generalization on real images** We finally validate our SkeDISN and SkeGCNN on Stanford online product images without further finetuning. Example results in Fig. 13 show that both our explicit and implicit methods are able to generalize to novel instances in real images.

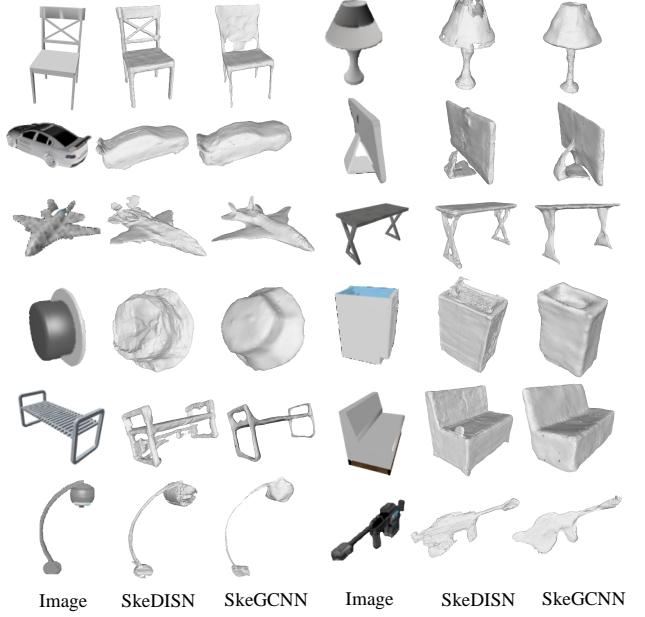


Fig. 11: Additional results of explicit and implicit mesh recoveries given by our proposed SkeGCNN and SkeDISN.



Fig. 12: Example results of multi-view mesh recovery from our proposed SkeGCNN and SkeDISN.

## 8 CONCLUSION

Recovering the surface shape of an object from one or a few perspectives of images is a fundamental yet challenging computer vision task. This work proposes to learn the topology-preserved, skeletal shape representation, and uses the learned representation to assist the surface recovery task of interest. In order to learn the skeletal representation, we propose an end-to-end, trainable *SkeletonNet* that is able to generate a high-quality skeletal volume via a bridged learning of a skeletal point set. The obtained skeletal volume can be used either as a bridge to explicitly recover a surface mesh in our proposed SkeGCNN, or as a constraint to regularize the learning of an implicit surface field in our proposed SkeDISN. Thorough experiments confirm both the efficacy of *SkeletonNet* for producing high-quality skeletal representations, and its usefulness for the downstream surface recovery tasks. In future research, we are interested in discovering the usefulness of skeletal representations for other shape analysis and reconstruction tasks.



Fig. 13: Example results of SkeGCNN and SkeDISN on novel instances contained in real images of Stanford online product images. For each of five instances, top row shows the input images, middle row shows the results of SkeDISN, and bottom row shows the results of SkeGCNN.

## REFERENCES

- [1] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 204, no. 1156, pp. 301–328, 1979.
- [2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2000.
- [3] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2007.
- [4] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," in *European Conference on Computer Vision*. Springer, 2008, pp. 766–779.
- [5] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Machine Vision and Applications*, vol. 23, no. 5, pp. 903–920, 2012.
- [6] P. Merrell, A. Albarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys, "Real-time visibility-based fusion of depth maps," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.
- [7] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust tv-l 1 range image integration," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.
- [8] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 127–136.
- [9] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," *European conference on computer vision*, pp. 628–644, 2016.
- [10] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *European Conference on Computer Vision*. Springer, 2016, pp. 484–499.
- [11] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2107–2115.
- [12] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Atlasnet: A papier-mâché approach to learning 3d surface generation," *computer vision and pattern recognition*, 2018.
- [13] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," *computer vision and pattern recognition*, pp. 3907–3916, 2018.
- [14] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2463–2471.
- [15] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," *arXiv preprint arXiv:1804.01654*, 2018.
- [16] J. Tang, X. Han, J. Pan, K. Jia, and X. Tong, "A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4541–4550.
- [17] J. Pan, J. Li, X. Han, and K. Jia, "Residual meshnet: Learning to deform meshes for single-view 3d reconstruction," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 719–727.
- [18] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, "Deep mesh reconstruction from single rgb images via topology modification networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9964–9973.
- [19] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotagh, and A. Eriksson, "Deep level sets: Implicit surface representations for 3d shape inference," *arXiv preprint arXiv:1901.06802*, 2019.
- [20] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.
- [21] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5939–5948.
- [22] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.
- [23] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "Disn: Deep implicit surface network for high-quality single-view 3d reconstruction," *arXiv preprint arXiv:1905.10711*, 2019.
- [24] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in Neural Information Processing Systems*, 2016, pp. 82–90.
- [25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *international conference on learning representations*, 2015.
- [26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [29] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3, 2017.
- [30] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "Octnetfusion: Learning depth fusion from data," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 57–66.
- [31] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3d object reconstruction," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 412–420.
- [32] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 72, 2017.
- [33] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Adaptive o-cnn: a patch-based deep representation of 3d shapes," in *SIGGRAPH Asia 2018 Technical Papers*. ACM, 2018, p. 217.
- [34] W. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *Computers Graphics*, 1987.
- [35] A. Arsalan Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, "Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1511–1519.
- [36] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3d object reconstruction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [37] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in

- European Conference on Computer Vision.* Springer, 2016, pp. 322–337.
- [38] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, “Mean curvature skeletons,” in *Computer Graphics Forum*, vol. 31, no. 5. Wiley Online Library, 2012, pp. 1735–1744.
- [39] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or, “Deep points consolidation,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 176, 2015.
- [40] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, “Shapenet: An information-rich 3d model repository,” *arXiv preprint arXiv:1512.03012*, 2015.
- [41] L. Jiang, S. Shi, X. Qi, and J. Jia, “Gal: Geometric adversarial loss for single-view 3d-object reconstruction,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 802–816.
- [42] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, “Learning representations and generative models for 3d point clouds,” in *ICLR 2018 : International Conference on Learning Representations 2018*, 2018.
- [43] D. Jack, J. K. Pontes, S. Sridharan, C. Fookes, S. Shirazi, F. Maire, and A. Eriksson, “Learning free-form deformations for 3d object reconstruction,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 317–333.
- [44] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, and C. Fookes, “Image2mesh: A learning framework for single image 3d reconstruction,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 365–381.
- [45] W. Wang, D. Ceylan, R. Mech, and U. Neumann, “3dn: 3d deformation network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1038–1046.
- [46] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. J. Zhang, “Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image.” *arXiv preprint arXiv:2002.12212*, 2020.
- [47] A. Kar, C. Häne, and J. Malik, “Learning a multi-view stereo machine,” in *Advances in neural information processing systems*, 2017, pp. 365–376.
- [48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [49] Z. Chen, A. Tagliasacchi, and H. Zhang, “Bsp-net: Generating compact meshes via binary space partitioning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 45–54.
- [50] J. Lei and K. Jia, “Analytic marching: An analytic meshing solution from deep implicit surface networks,” *arXiv preprint arXiv:2002.06597*, 2020.
- [51] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, “Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization,” *arXiv preprint arXiv:1905.05172*, 2019.
- [52] H. Blum, *A transformation for extracting new descriptors of shape*. MIT press Cambridge, 1967, vol. 4.
- [53] K. Siddiqi and S. Pizer, *Medial representations: mathematics, algorithms and applications*. Springer Science & Business Media, 2008, vol. 37.
- [54] D. Attali and A. Montanvert, “Modeling noise for a better simplification of skeletons,” in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3. IEÉE, 1996, pp. 13–16.
- [55] F. Chazal and A. Lieutier, “The  $\lambda$ -medial axis,” *Graphical Models*, vol. 67, no. 4, pp. 304–331, 2005.
- [56] B. Miklos, J. Giesen, and M. Pauly, “Discrete scale axis representations for 3d geometry,” in *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–10.
- [57] P. Li, B. Wang, F. Sun, X. Guo, C. Zhang, and W. Wang, “Q-mat: Computing medial axis transform by quadratic error minimization,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 1, pp. 1–16, 2015.
- [58] Y. Pan, B. Wang, X. Guo, H. Zeng, Y. Ma, and W. Wang, “Q-mat+: An error-controllable and feature-sensitive simplification algorithm for medial axis transform,” *Computer Aided Geometric Design*, vol. 71, pp. 16–29, 2019.
- [59] B. Yang, J. Yao, B. Wang, J. Hu, Y. Pan, T. Pan, W. Wang, and X. Guo, “P2mat-net: Learning medial axis transform from sparse point clouds,” *Computer Aided Geometric Design*, p. 101874, 2020.
- [60] J. Hu, B. Wang, L. Qian, Y. Pan, X. Guo, L. Liu, and W. Wang, “Matnet: Medial axis transform network for 3d object recognition.” in *IJCAI*, 2019, pp. 774–781.
- [61] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, “On-the-fly curve-skeleton computation for 3d shapes,” in *Computer Graphics Forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 323–328.
- [62] A. Bucksch, R. Lindenbergh, and M. Menenti, “Skeltre,” *The Visual Computer*, vol. 26, no. 10, pp. 1283–1300, 2010.
- [63] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, “L1-medial skeleton of point cloud.” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 65–1, 2013.
- [64] K. Yin, H. Huang, D. Cohen-Or, and H. Zhang, “P2p-net: Bidirectional point displacement net for shape transform,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–13, 2018.
- [65] D. A. Field, “Laplacian smoothing and delaunay triangulations,” *Communications in applied numerical methods*, vol. 4, no. 6, pp. 709–712, 1988.
- [66] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, “High-resolution shape completion using deep neural networks for global structure and local geometry inference,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 85–93.
- [67] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *medical image computing and computer assisted intervention*, pp. 234–241, 2015.