A Skeleton-bridged Deep Learning Approach for Generating Meshes of Complex Topologies from Single RGB Images

Jiapeng Tang [*1], Xiaoguang Han [*2], Junyi Pan[1], Kui Jia [†1], and Xin Tong[3]

[1]School of Electronic and Information Engineering, South China University of Technology
[2]Shenzhen Research Institute of Big Data, the Chinese University of Hong Kong (Shenzhen)
[3]Microsoft Research Asia

msjptang@mail.scut.edu.cn, hanxiaoguang@cuhk.edu.cn, eejypan@mail.scut.edu.cn,
kuijia@scut.edu.cn, xtong@microsoft.com

Abstract

This paper focuses on the challenging task of learning 3D object surface reconstructions from single RGB images. Existing methods achieve varying degrees of success by using different geometric representations. However, they all have their own drawbacks, and cannot well reconstruct those surfaces of complex topologies. To this end, we propose in this paper a skeleton-bridged, stage-wise learning approach to address the challenge. Our use of skeleton is due to its nice property of topology preservation, while being of lower complexity to learn. To learn skeleton from an input image, we design a deep architecture whose decoder is based on a novel design of parallel streams respectively for synthesis of curve- and surface-like skeleton points. We use different shape representations of point cloud, volume, and mesh in our stage-wise learning, in order to take their respective advantages. We also propose multi-stage use of the input image to correct prediction errors that are possibly accumulated in each stage. We conduct intensive experiments to investigate the efficacy of our proposed approach. Qualitative and quantitative results on representative object categories of both simple and complex topologies demonstrate the superiority of our approach over existing ones. We will make our ShapeNet-Skeleton dataset publicly available.

1. Introduction

Learning 3D surface reconstructions of objects from single RGB images is an important topic from both the aca-

Figure 1. Our proposed approach can generate a closed surface mesh from a single view RGB image, by correctly recovering the complex topology.

demic and practical perspectives. This inverse problem is extremely challenging due to the arbitrary shapes of different object instances and their possibly complex topologies. Recent methods [4, 7, 27, 30, 8, 11, 24, 6, 16, 28, 19] leverage the powerful learning capacities of deep networks, and achieve varying degrees of success by using different shape representations, e.g., volume, point cloud, or mesh. These methods have their own merits but also have their respective drawbacks. For example, volume-based methods [4, 7, 30] exploit the establishment of Convolutional Neural Networks (CNNs) [23, 14, 26, 10], and simply extend CNNs its 3D versions to generate volume representations of 3D shapes; however, both of their computational and memory complexities are high enough which prohibit them to be deployed to generate high-resolution outputs. On the other hand, point cloud based methods [6, 16] are by nature difficult to generate smooth and clean surfaces.

Given the fact that mesh representation is a more effi-

---

[*]Equal contributions
[†]Corresponding author

cient, discrete approximation of the continuous manifold of an object surface, a few recent methods [6, 16] attempt to directly learn mesh reconstructions from single input images. These methods are inherently of mesh deformation, since they assume that an initial meshing over point cloud is available; for example, they typically assume unit square/sphere as the initial mesh. In spite of the success achieved by these recent methods, they still suffer from generating surface meshes of complex topologies, e.g., those with thin structures as shown in Fig 1.

To this end, we propose in this paper a *skeleton-bridged, stage-wise* deep learning approach for generating mesh reconstructions of object surfaces from single RGB images. We particularly focus on those object surfaces with complex topologies, e.g., chairs or tables that have local, long and thin structures. Our choice of the *meso-skeleton* [1] is due to its nice property of topology preservation, while being of lower complexity to learn when compared with learning the surface meshes directly. Our proposed approach is composed of three stages. The first stage learns to generate skeleton points from the input image, for which we design a deep architecture whose decoder is based on a novel, parallel design of CurSkeNet and SurSkeNet, which are respectively responsible for the synthesis of curve- and surface-like skeleton points. To train CurSkeNet and SurSkeNet, we compute skeletal shape representations for instances of ShapeNet [3]. *We will make our ShapeNet-Skeleton dataset publicly available.* In the second stage, we produce a base mesh by firstly converting the obtained skeleton to its coarse volumetric representation, and then refining the coarse volume using a learned 3D CNN, where we adopt a strategy of independent sub-volume synthesis with regularization of global structure, in order to reduce the complexity of producing high-resolution volumes. In the last stage, we generate our final mesh result by extracting a base mesh from the obtained volume [17], and deforming vertices of the base mesh using a learned Graph CNN (GCNN) [12, 5, 1, 22]. Learning and inference in three stages of our approach are based on different shape representations, which take the respective advantages of point cloud, volume, and mesh. We also propose multi-stage use of the input image to correct prediction errors that are possibly accumulated in each stage. We conduct intensive ablation studies which show the efficacy of stage-wise designs of our proposed approach.

We summarize our main contributions as follows.

- Our approach is based on an integrated stage-wise learning, where learning and inference in different stages are based on different shape representations by

---

taking the respective advantages of point cloud, volume, and mesh. We also propose multi-stage use of the input image to correct prediction errors that are possibly accumulated in each stage.

- We propose in this paper a skeleton-bridged approach for learning object surface meshes of complex topologies from single RGB images. Our use of skeleton is due to its nice property of topology preservation, while being of lower complexity to learn. We design a deep architecture for skeleton learning, whose decoder is based on a novel design of parallel streams respectively for the synthesis of curve- and surface-like skeleton points. To train the network, we prepare ShapeNet-Skeleton dataset and will make it publicly available.

- We conduct intensive ablation studies to investigate the efficacy of our proposed approach. Qualitative and quantitative results on representative object categories of both simple and complex topologies demonstrate the superiority of our approach over existing ones, especially for those objects with local thin structures.

2. Related Works

In this section, we only focus on the related works about deepnets-based algorithms for fully object reconstruction. The literature reviews are studied in the following three aspects.

Volume-based Generator Voxels, extended from pixels, are usually used in the form of binary values or signed distances to represent a 3D shape. Because of its regularity, most of existing deepnets-based shape analysis [32, 2] or shape generation [4, 7, 30] methods adopt it as the primary representation. For example, the work of [4] combines 3D convolutions with long short-term memory (LSTM) units to achieve volumetric grid reconstruction from single-view or multi-view RGB images. These methods tend to predict a low-resolution volumetric grid due to the high computational cost of 3D convolution operators. Based on the observation that only a small portion of regions around the boundary surface contain the shape information, the Octree representation has been adopted in recent shape analysis works [20, 29]. A convolutional Octree decoder is also designed in [27] to support high-resolution reconstruction with a limited time and memory cost. In our work, we aim to generate the surface mesh of the object instead of its solid volume. As its efficiency and topology-insensitivity, we also leverage volumetric-based generator to convert the inferred skeletal point cloud to a solid volume, effectively bridging the gap between the skeleton and the surface mesh.

Surface-based Generator Point cloud, sampled from the object' surface, is one of the most popular representations of 3D shapes. Fan et al. [6] proposes the first point could

---

[1]Skeletal shape representation is a kind of medial axis transform (MAT). 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.
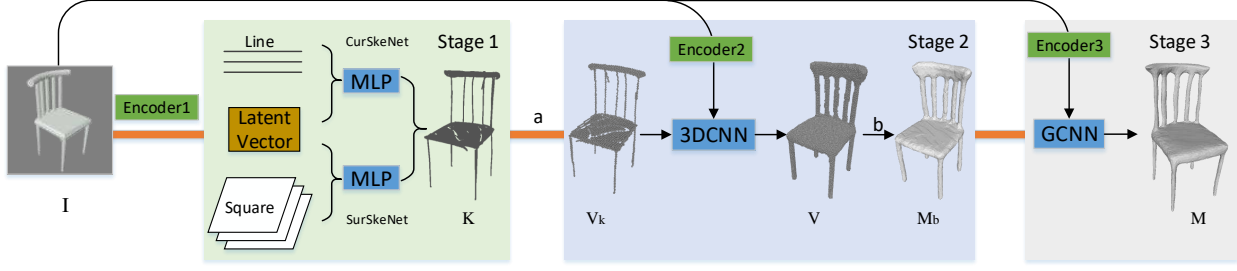
Figure 2. Our overall pipeline. Given an input image I, we employ two parallel MLPs to infer skeletal points K in stage one. After converting K to a coarse volume V_k, we refine V_k to get V by 3D CNN and extract base mesh M_b from V in stage two. We further optimize vertices of M_b using GCNN to acquire a final mesh M. The operation a means voxelization and the operation b stands for Marching Cubes.

generation neural network, which is built upon a deep regression model trained with the loss functions that evaluate the similarity of two unordered point set, such as chamfer distance. Although the rough shape can be captured, the generated points are placed sparse and scattered. Mesh, as the most natural discretization of a manifold surface, has been widely used in many graphics applications. Due to its irregular structure, CNN is difficult to be directly applied to mesh generation. To alleviate this challenge, the methods of [11, 28] take an extra template mesh as input and attempt to learn the deformations to approximate the target surfaces. Limited to the requirement of an initial mesh, they cannot deal with topology-free reconstruction. Another recent method, called Atlasnet [8], proposes to deform multiple 2D planar patches to cover the surface of the object. Residual prediction and progressive deformation are adopted in [19], which decrease the complexity of learning and make more details added. It is free of complex topology yet causes severe patch overlaps and holes. In our work, we aim not only to generate a clean mesh but also to capture the correct topology. To do so, we firstly borrow the idea in [8] to infer the meso-skeleton points, which are then converted to a base mesh. Finally, the method of [28] is further adopted for generating geometric details.

Structure Inference Instead of estimating geometric shapes, many recent works attempt to recover the 3D structures of objects. From a single image, Zou et al. [33] presents a primitive recurrent neural network to sequentially predict a set of cuboid primitives to approximate the shape structure. A recursive decoder is proposed in [15] to generate shape parts and infer reasonable high-level structure information including part connectivity, adjacency and symmetry relation. This is further exploited in [18] for image-based structure inference. However, the cuboids are hard to be used for fitting curved shapes. In addition, these methods also require a large human-labeled dataset. We use meso-skeleton, a point cloud, to represent the shape struc-

ture which is easier to be obtained from the ground truth. The usage of parametric line and square elements also eases the approximation of the diverse local structures.

3. The Proposed Approach

We first overview our proposed skeleton-bridged approach for generating a surface mesh from an input RGB image, before explaining the details of stage-wise learning. Given an input image I of an object, our goal is to recover a surface mesh M that ideally captures the *possibly complex* topology of 3D shape of the object. This is an extremely challenging inverse task; existing methods [11, 28, 8] may only achieve partial success for objects with relatively simple topologies. To address the challenge, our key idea in this work is to bridge the mesh generation of object surface via learning of meso-skeleton. As discussed in Section 1, the rationale is that skeletal shape representation preserves the main topological structure of a shape, while being of lower complexity to learn.

More specifically, our mesh generation process is composed of the following three stages. In the first stage, we learn an encoder-decoder architecture that maps I to its meso-skeleton K, represented as a compact point cloud. In the second stage, we produce a volume V from K by firstly converting K to its coarse volumetric representation V_k, and then refining V_k using a learned 3D CNN (e.g., of the style [9]). In the last stage, we generate the final output mesh M by extracting a base mesh M_b from V, and further optimizing vertices of M_b using a learned graph CNN [22]. Each stage owns its own image encoder, and thus inferences in all the three stages are guided by the input image I. Fig 2 illustrates the whole pipeline of our approach.

3.1. Learning of Meso-Skeleton

As defined in Section 1, the meso-skeleton of a shape is represented as its medial axis, and the medial axis of a 3D model is made up of curve skeletons and median sheets,

which are adaptively generated from local regions of the shape. In this work, we utilize the skeleton representation introduced in [31], i.e., a compact point cloud. Fig 8 shows an example of skeleton that we aim to recover.

The ShapeNet-Skeleton dataset Training skeletons are necessary in order to learn to generate a skeleton from an input image. In this work, we prepare training data of skeleton for ShapeNet [3] as follows: 1) for each 3D polygonal model in ShapeNet, we convert it into a point cloud; 2) we extract meso-skeleton points using the method of [31]; 3) we classify each skeleton point as either curve-like or surface-like categories, based on principle component analysis of its neighbor points. *We will make our ShapeNet-Skeleton dataset publicly available.*

CurSkeNet and SurSkeNet Given the training skeleton points for the object in each image, we design an encoder-decoder architecture for skeleton learning, where the input I is firstly encoded to a latent vector that is then decoded to a point cloud of skeleton. Our encoder is similar to those in existing methods of point set generation, such as [6, 8]. In this work, we use ResNet-18 [10] as our image encoder. Our key contribution is a novel design of decoder architecture that will be presented shortly. We note that one may think of using existing methods [6, 8] to generate K from I; however, they tend to fail due to the complex, especially thin, structures of skeletons, as shown in Fig 8. Our decoder is based on two parallel streams of CurSkeNet and SurSkeNet, which are designed to synthesize the points at curve-shaped and surface-shaped regions respectively. Both CurSkeNet and SurSkeNet are based on multilayer perceptrons (MLPs) with the same settings as in AtlasNet [8], including 4 fully-connected layers with the respective sizes of 1024, 512, 256, and 3, where the non-linear activation functions are ReLU for the first 3 layers and tanh for the last layer. Our SurSkeNet learns to deform a set of 2D primitives defined on the open unit square $[0, 1]^2$, producing a local approximation of the desired sheet skeleton points. Our CurSkeNet learns to deform a set of 1D primitives defined on the open unit line $[0, 1]$; it thus conducts affine transformations on them to form curves, and learns to assemble generated curves to approximate the curve-like skeleton part. In our current implementation, we use 20 line primitives in CurSkeNet and 20 square primitives in SurSkeNet. In Section 4.3, we conduct ablation studies that verify the efficacy of our design of CurSkeNet and SurSkeNet.

Network Training We use training data of curve-like and surface-like skeleton points to train CurSkeNet and SurSkeNet. The learning task is essentially of point set generation. Similar to [8], we use the Chamfer Distance (CD) as one of our loss functions. The CD loss is defined as:

$$\mathcal{L}_{cd} = \sum_{x \in K} \min_{y \in K^*} \|x - y\|_2^2 + \sum_{y \in K^*} \min_{x \in K} \|x - y\|_2^2, \quad (1)$$
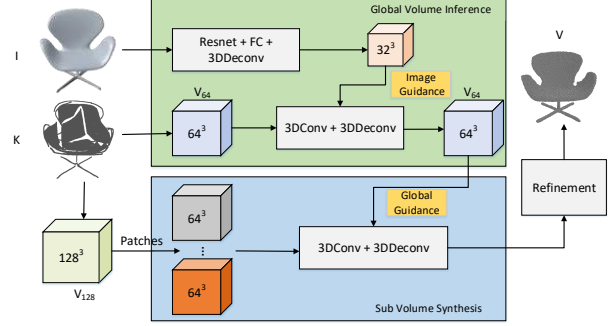

Figure 3. The pipeline of our high-resolution skeletal volume synthesis method. We convert the inferred skeletal points K to low-resolution volume V_{64} and high-resolution volume V_{128} in parallel. Given V_{64}, V_{128} paired with the input image I, a global-guided sub-volume synthesis network is proposed to output a refined volume of V_{128}. It consists of two subnetworks: one network generates a coarse skeletal volume from I and V_{64} while the other enhances V_{128} locally patch by patch under the guidance of the output from the first network.

where $\{x \in K\}$ and $\{y \in K^*\}$ are respectively the sets of predicted and training points. Besides, to ensure local consistency, regularizer of Laplacian smoothness is also used for generation of both curve- and surface-like points. It is defined as:

$$\mathcal{L}_{lap} = \sum_{x \in K} \left\| x - \frac{1}{|\mathcal{N}(x)|} \sum_{p \in \mathcal{N}(x)} p \right\|_2, \quad (2)$$

where $\mathcal{N}(x)$ is the neighbor of point x.

3.2. From Skeleton to Base Mesh

We present in this section how to generate a base mesh M_b from the obtained skeleton K. To do so, a straightforward approach is to coarsen K to a volume directly with hand-crafted methods, and then to produce the base mesh using the method of Marching Cubes [17]. However, such an approach may accumulate stage-wise prediction errors. Instead, we rely on the original input I to correct the possible stage-wise errors, by firstly converting K to its volumetric representation V_k, and then using a trained 3D CNN for a finer and more accurate volumetric shape synthesis, resulting in a volume V. Base mesh M_b can then be obtained by applying Marching Cubes to the finer V.

Sub-volume Synthesis with Global Guidance To preserve the topology captured by K, a high-resolution volume representation is required. However, this is not easy to satisfy due to the expensive computational cost of 3D convolution operations. OctNet [20] may alleviate the computational burden, it is however complex and difficult to implement. We instead partition the volume space into overlapped sub-volumes, and conduct refinement on them in parallel. We

also follow [9] and employ a global guidance to preserve spatial consistency across sub-volumes. More specifically, we firstly convert K to two volumes of varying scales, denoted as V_k^l and V_k^h. We set $|V_k^l| = 64^3$ and $|V_k^h| = 128^3$ in this work. We use two networks of 3D CNNs for global and local synthesis of skeletal volumes. The global network is trained to refine V_k^l and generate a skeletal volume V^l of the size 64^3. The local network takes as inputs sub-volumes of the size 64^3, which are uniformly cropped from V_k^h, and then conduct their refinement individually. Both of our global and local refinement networks are based on 3D U-Net architecture [21]. When refining each sub-volume of V_k^h, the corresponding 32^3-sized sub-volume of V^l is concatenated to provide structural regularization. The overall pipeline of our method is shown in Fig 3. As seen in Fig 4, our method not only supports high-resolution synthesis but also preserves global structure.
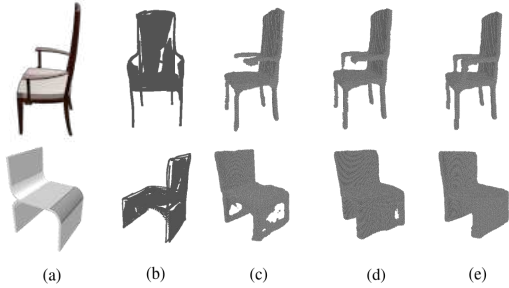

Figure 4. (a)Input images; (b)Inferred skeletal points; (c)sub-volume synthesis only; (d) adding global guidance; (e) adding image guidance.

Image-guided Volume Correction To correct the possibly accumulated prediction errors from the stage of skeleton generation, we reuse the original input I by learning an independent encoder-decoder network, which is trained to map I to a 32^3-sized volume. We use ResNet-18 as the encoder and several 3D de-convolution layers as the decoder. The output of the decoder is incorporated into the aforementioned global synthesis network, aiming for a more accurate V^l, which ultimately contributes to the generation of a better V. From the perspective of learning task for generating 3D volumes from single images [4, 7, 30, 27], our method is superior to existing ones by augmenting with an additional path of skeleton inference. As shown in Fig 4, our usage of I for error correction greatly improves the synthesis results.

Base Mesh extraction Given V, we use Marching Cubes [17] to produce the base mesh M_b, which ideally preserves the same topology as that of the skeleton K. Because V is in high resolution, M_b would contain a large number of vertices and faces. To reduce the computational burden of the last stage, we apply QEM algorithm [13] on M_b to get a simplified mesh for subsequent processing.
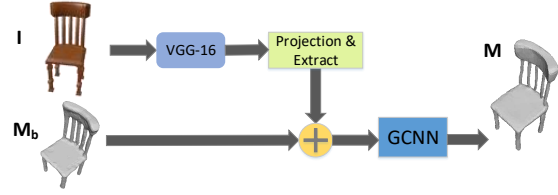
3.3. Mesh Refinement

Figure 5. Our mesh refinement network. Given an image I and an initial mesh M_b, we concatenate pixel-wise features of I (extracted by VGG-16) to vertices' coordinates and form vertex-wise features which are followed by a graph-CNN to generate the geometric details.

We have up to now the base mesh M_b that captures the topology of the underlying object surface, but may lack surface details. To compensate M_b with surface details, we take the approach of mesh deformation using graph CNNs [12, 5, 1, 22].

Mesh Deformation using Graph CNNs Take M_b as the input, our graph CNN is simply composed of a few graph convolutional layers, each of which apply spatial filtering operation to local neighborhood associated with each vertex point of M_b. The graph-based covolutional layer is defined as:

$$h_p^{l+1} = w_0 h_p^l + \sum_{q \in \mathcal{N}(p)} w_1 h_q^l, \qquad (3)$$

where h_p^l, h_p^{l+1} are the feature vectors on the vertex p before and after applying a convolution operation, and $\mathcal{N}(p)$ is the neighbor of p. w_0 and w_1 are the learnable parameter matrices that are applied to all vertices.

Similar to [28], we also concatenate pixel-wise VGG features extracted from I with coordinates of the corresponding vertices to enhance learning. We again use CD loss to train our graph CNN. Several smoothness terms are also added to regularize the mesh deformation. One is edge regularization, used to avoid large deformations, by restricting the length of output edges. Another one is normal loss, used to guarantee the smoothness of the output surface. The geometric details commonly exist at the regions where the normals are changed obviously. Regarding this fact, to guide the GCNN to better learn the surface in those areas, we accordingly construct weighted loss functions. Fig 5 shows the efficacy of this weighting strategy, where the sharp edges are better synthesized.

4. Experiments

Dataset To support the training and testing of our proposed approach, we collect 17705 3D shapes from five categories in ShapeNet [3]: plane(1000), bench(1816), chair(5380),
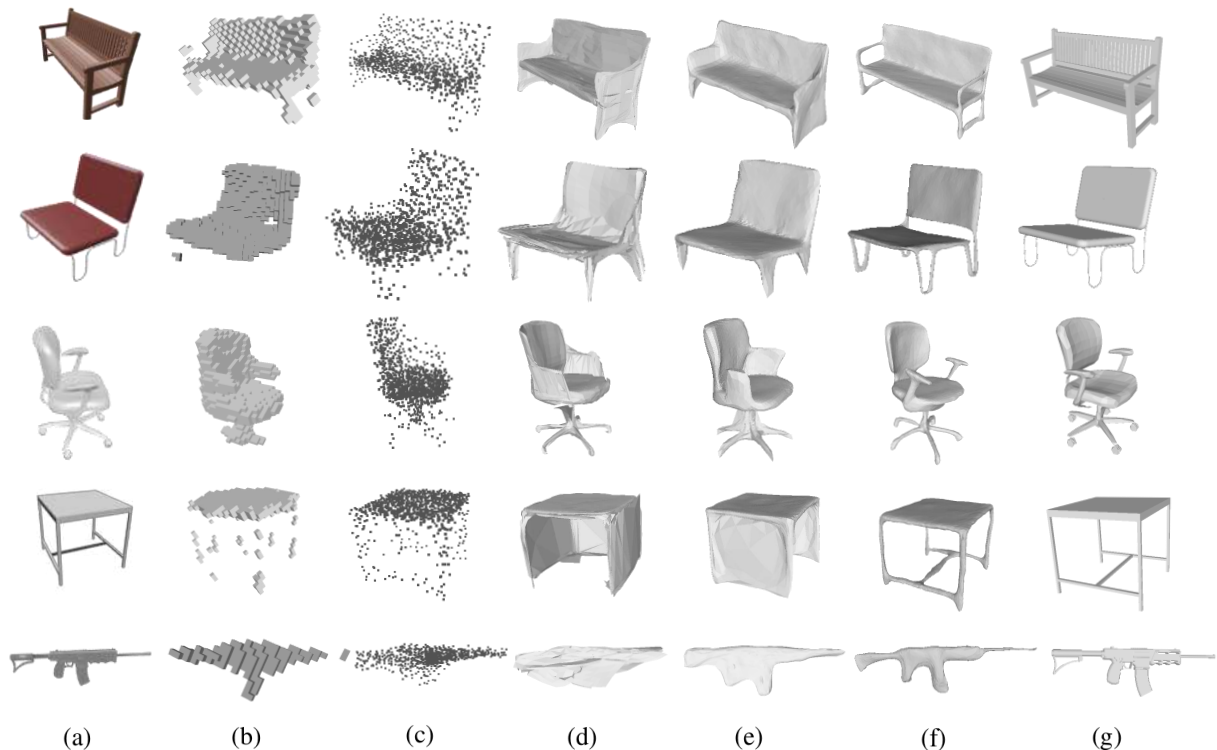
Figure 6. (a)Input images; (b)R2N2; (c)PSG; (d)AtlasNet; (e)Pixel2Mesh; (f)Ours; (g)Ground truth

table(8509), firearm(1000). The dataset is split into two parts, 80% shapes are used for training and the other for testing. We take as the inputs the rendered images provided by [4], where each 3D model is rendered into 24 RGB images. Each shape in the dataset is converted to a point cloud (10, 000 points are sampled on the surface) as the ground truth for mesh refinement network.

Implementation details The input images are all in the size of 224*224. We train CurSkeNet and SurSkeNet using a batch size of 32 with a learning rate of 1e-3 (dropped to 3e-4 after 80 epochs) for 120 epochs. The skeletal volume refinement network is trained in three steps: 1) the global volume inference network is trained alone with learning rate 1e-4 for 50 epochs(dropped to 1e-5 after 35 epochs); 2) we train the sub-volume synthesis network with learning rate 1e-5 for 10 epochs; 3) the entire network is fine-tuned. The mesh refinement network is trained with learning rate 3e-5 for 50 epochs(dropped to 1e-5 after 20 epochs) using a batch size of 1.

4.1. Comparisons against State-of-the-Arts

We first evaluated our overall pipeline against existing methods on singe-view reconstruction. 3D-R2N2 [4], PSG [6], AtlasNet [8], Pixel2Mesh [28] are chosen for their popularity: 3D-R2N2 is one of the most famous volumet-ric shape generators, PSG is the first point set generator based on a deep regression model, and both AtlasNet and Pixel2Mesh are current state-of-the-art mesh generator. For fair comparison, these models are retrained under our pre-processed dataset.

Qualitative results The visual comparisons are shown in Fig 6. As seen, 3D-R2N2 always produces low-resolution volumes which cause broken structures. Their results show no surface details either. The point sets regressed by PSG are sparse and scattered, leading to the difficulty of extract-ing triangular meshes from them. AtlasNet is capable of generating mesh representations without a strong restric-tion on the shape's topology. Yet, the outputs are of non-closed and suffer from surface self-penetration, which also gives rise to a challenge to convert it to a manifold mesh. Limited to the requirement of a genus-0 template mesh in-put, Pixel2Mesh is difficult to reach an accurate reconstruc-tion for the objects with complex topologies, as the chairs shown. Our method shows great superiority than the others from the visual appearances, as it generates closed meshes with accurate topologies and more details. For the exam-ples of firearm as shown, our approach also outperforms Pixel2Mesh, which in another aspect, indicates the pro-posed approach is also good at recovering the shapes with complex structures no mention to topology.

Category	CD					EMD				
	R2N2	PSG	AtlasNet	Pixel2Mesh	Ours	R2N2	PSG	AtlasNet	Pixel2Mesh	Ours
plane	10.434	3.824	1.529	1.890	**1.364**	11.060	13.945	8.981	7.728	**6.026**
bench	10.511	3.504	2.264	1.774	**1.639**	10.555	8.053	9.143	7.083	**6.059**
chair	4.723	2.553	1.342	1.923	**1.002**	7.762	10.222	7.866	8.312	**5.484**
firearm	10.176	**1.473**	2.276	1.793	1.784	9.760	12.555	9.825	6.887	**6.413**
table	12.230	5.466	1.751	2.109	**1.321**	11.160	9.561	9.053	7.442	**5.688**
mean	9.615	3.364	1.832	1.898	**1.422**	10.059	10.867	8.974	7.490	**5.934**

Table 1. Quantitative comparisons of our method against state-of-the-arts. The Chamfer distance($\times 10^3$) and Earth Mover's distance($\times 0.01$) are used. The lower is better on all metrics.
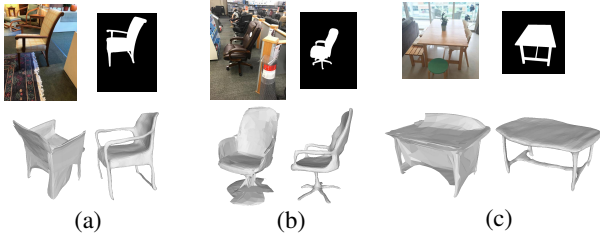

(a)　　　　　　(b)　　　　　　(c)

Figure 7. From real photographs and object masks (top row), our method successfully reconstructs 3D object meshes. The results of AtlasNet (left of bottom row) v.s ours (right of bottom row).

Quantitative results Similar to Pixel2Mesh [28], we adopt Chamfer Distance(CD) and Earth Mover's Distance(EMD) to evaluate the reconstruction quality. Both of them are calculated between the point set ($10,000$ points) sampled on the predicted mesh surface and the ground truth point cloud. The quantitative comparison results are reported in Tab 1. Notably, on both metrics, our approach outperforms all the other methods across almost all listed categories, especial on the models with complex topologies like chairs and tables.

Generalization on real images Fig 7 illustrates 3D shapes reconstructed by our method on three real photographs from Pix3D [25], where the chairs and tables in the images are manually segmented. The results' quality is similar to the results obtained from synthetic images. As seen in Fig 7 (a), the real-world images has no relation with ShapeNet, while the chair rod can still be well reconstructed. This validates the generalization ability of our method.

4.2. Ablation Studies on Mesh Generation

Our whole framework contains multiple stages. In this section, we conduct the ablation studies by alternatively removing one of them, to verify the necessity of each stage.

w/o skeleton inference Based on our pipeline, an alternative solution without using skeleton inference is firstly generating a volume directly from the image and then applying our mesh refinement model to output the final result. Then, we implement this approach by using OGN [27] as the image-based volume generator, for high-resolution(128^3)

reconstruction. This method is compared with ours visually in Fig 9. As seen, the OGN-based mesh generation method fails to capture the thin structures which causes incorrect topologies. In contrast, our approach gives rise to much better performance.

w/o voxel-based correction After inferring skeleton from our first stage, it is a straightforward approach to acquire a base mesh by directly applying the corrosion technique for volume generation, and the base mesh can be extracted. The visual comparisons of this method against ours are shown in Fig 10. It can be seen, without volume correction, the wrong predictions caused by skeleton inference will be transferred to the mesh refinement stage, affecting the final output. Our proposed voxel-based correction network addresses this issue effectively.

4.3. Evaluation on Skeleton Inference

In this section, we conduct comparisons with several variants of our skeleton inference approach, to verify our final model is the optimal choice. These variants include: "Point-only fitting" method directly adopts PSG [6] to regress the skeletal points; "Line-only fitting" method removes the square stream of our model and only deforms multiple lines to approximate the skeleton; "Square-only fitting" removes the line stream of our model and deforms multiple squares to fit the skeleton; "Line-and-Square fitting" method learns the deformation of multiple lines and squares together using a single MLP to approximate the skeleton; "Ours w/o laplacian" stands for our model without laplacian smoothness term. Note that, laplacian smoothness loss is also used for the training of "Line-only fitting", "Square-only fitting" and "Line-and-Square fitting".

Quantitative results All of these methods are evaluated on CD metric and the results are shown in Tab 2. It can be seen that our final model outperforms all the others. Another discovery is that laplacian regularizer is very helpful to reach better accuracy.

Qualitative results We then report the visual comparisons of these methods on a sampled example in Fig 8. As shown,
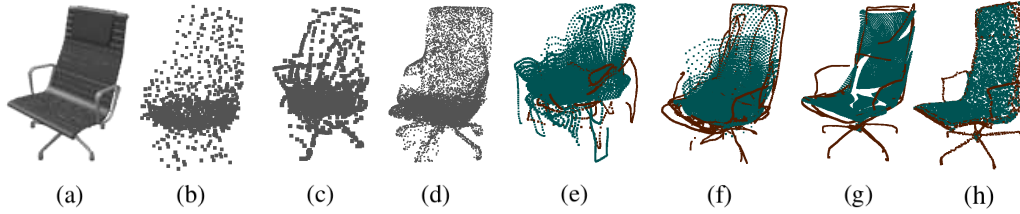
Figure 8. (a)Input images; (b)Point-fitting only; (c)Line-fitting only; (d)Square-fitting only; (e)Line-and-square fitting; (f)Ours w/o laplacian; (g)Ours final; (h)Ground truth.
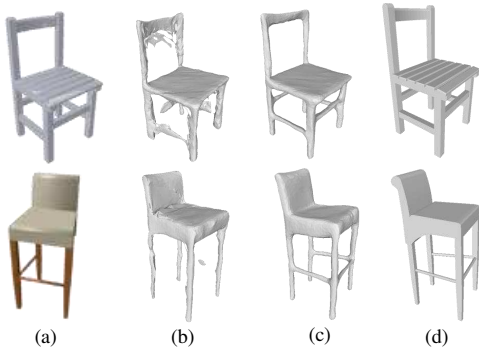

Figure 9. (a)Input images; (b)Final meshes whose base meshes are generated using OGN; (c)The generated meshes of our method; (d)Ground truth.
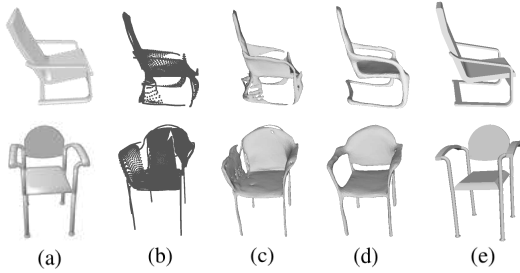

Figure 10. (a)Input images; (b)Inferred skeleton points; (c)The sythesized meshes whose base meshes are extracted from the coarsened skeletal volume using the corrosion techinique; (d) The generated meshes of our method; (e)Ground truth.

Methods	CD
Point-only fitting	1.185
Line-only fitting	1.649
Square-only fitting	1.185
Line-and-Square fitting	1.252
Ours w/o laplacian	1.621
Ours	**1.103**

Table 2. The quantitative comparisons on the variants of our skeleton inference method. The Chamfer Distance($\times 10^3$) are reported.

5. Conclusion

Recovering the 3D shape of an object from one of its perspectives is a very fundamental yet challenging task in computer vision field. The proposed framework splits this challenge task into three stages. It firstly recovers a 3D meso-skeleton represented as points, these skeletal points are then converted to its volumetric representation and passed to a 3DCNN for a solid volume synthesis. From which, a coarse mesh can be extracted. A GCNN is finally trained to learn the mesh deformation for producing geometric details. As demonstrated in our experiments both qualitatively and quantitatively, the proposed pipeline outperforms all existing methods. There are two directions worth being explored in the future: 1)how to change the whole pipeline to be an end-to-end network; 2) trying to apply adversarial learning on skeletal point inference, volume generation, and mesh refinement, for further improving the quality of final output mesh.

6. Acknowledge

point-only fitting results in scattered points no mention to the structures. Line only fitting fails to recover the surface-shaped skeleton parts. Square-only fitting can not capture the long and thin rods and legs. The method of Line-and-Square fitting causes messy outputs since a single MLP is difficult to approximate diverse local structures. As observed, the involvement of laplacian loss effectively improves the visual appearance of the results.

References

[1] D. Boscaini, J. Masci, E. Rodolà, and M. M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. *neural information processing systems*, pages 3189–3197, 2016.

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016.

[3] 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.

[4] 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*, pages 628–644, 2016.

[5] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *neural information processing systems*, pages 3844–3852, 2016.

[6] 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)*, pages 2463–2471, 2017.

[7] 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*, pages 484–499. Springer, 2016.

[8] 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.

[9] 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)*, pages 85–93, 2017.

[10] 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*, pages 770–778, 2016.

[11] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. *computer vision and pattern recognition*, pages 3907–3916, 2018.

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *international conference on learning representations*, 2017.

[13] L. Kobbelt, S. Campagna, and H.-P. Seidel. A general framework for mesh decimation. In *Graphics interface*, volume 98, pages 43–50, 1998.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.

[15] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):to appear, 2017.

[16] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. *arXiv preprint arXiv:1706.07036*, 2017.

[17] W. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computers Graphics*, 1987.

[18] C. Niu, J. Li, and K. Xu. Im2struct: Recovering 3d shape structure from a single rgb image. *computer vision and pattern recognition*, 2018.

[19] 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)*, pages 719–727. IEEE, 2018.

[20] 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*, volume 3, 2017.

[21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *medical image computing and computer assisted intervention*, pages 234–241, 2015.

[22] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *international conference on learning representations*, 2015.

[24] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 791–800, 2017.

[25] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018.

[26] 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)*, pages 1–9, 2015.

[27] 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)*, pages 2107–2115, 2017.

[28] 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.

[29] 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)*, 36(4):72, 2017.

[30] 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*, pages 82–90, 2016.

[31] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or. Deep points consolidation. *ACM Transactions on Graphics (TOG)*, 34(6):176, 2015.

[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[33] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.