Object Wake-up: 3D Object Rigging from a Single Image

Ji Yang[1*], Xinxin Zuo[1*], Sen Wang[1,2], Zhenbo Yu[3], Xingyu Li[1], Bingbing Ni[2,3], Minglun Gong[4], and Li Cheng[1]

[1] University of Alberta
[2] Huawei HiSilicon
[3] Shanghai Jiao Tong University
[4] University of Guelph
{jyang7,xzuo,sen9,xingyu,lcheng5}@ualberta.ca

Abstract. Given a single chair image, could we wake it up by reconstructing its 3D shape and skeleton, as well as animating its plausible articulations and motions, similar to that of human modeling? It is a new problem that not only goes beyond image-based object reconstruction but also involves articulated animation of generic objects in 3D, which could give rise to numerous downstream augmented and virtual reality applications. In this paper, we propose an automated approach to tackle the entire process of reconstruct such generic 3D objects, rigging and animation, all from single images. A two-stage pipeline has thus been proposed, which specifically contains a multi-head structure to utilize the deep implicit functions for skeleton prediction. Two in-house 3D datasets of generic objects with high-fidelity rendering and annotated skeletons have also been constructed. Empirically our approach demonstrated promising results; when evaluated on the related sub-tasks of 3D reconstruction and skeleton prediction, our results surpass those of the state-of-the-arts by a noticeable margin. Our code and datasets are made publicly available at the dedicated project website.

Keywords: Object Reconstruction, Object Rigging

1 Introduction

Presented with a single image of a generic object, say an airplane or a chair, our goal is to wake it up in the 3D virutal world: this entails reconstructing its 3D shape and the skeleton, as well as animating its plausible articulations and motions, such as an airplane flapping its wings or a chair walking as a quadruped, as illustrated in Fig. 1. This is a relatively new problem that may have many downstream applications in virtual and augmented reality scenarios. It is worth noting that there has been research efforts [15] performing 3D manipulations

* equal contribution
Project website: https://kulbear.github.io/object-wakeup/

from single input images, where the main focus is toward rigid body transformations. To generate non-rigid shape deformations, it is usually necessary to involve intensive user interactions and dedicated software tools. Instead, our aim in this paper is to automate the entire process of object 3D reconstruction, rigging, and animation. The generic objects considered here are articulated, such that their shapes are capable of being deformed by a set of skeletal joints. In a way, our problem may be considered as a generalization of image-based 3D human shape and pose reconstruction to generic objects encountered in our daily life, as long as they could be endowed with a skeleton.

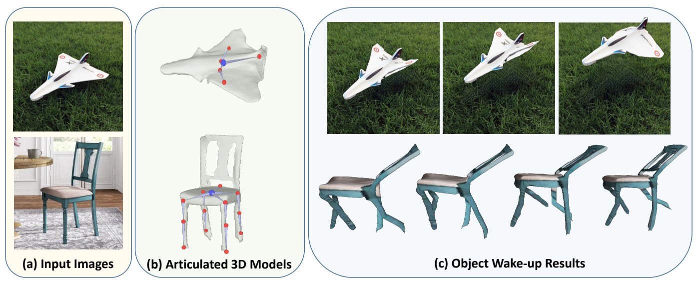

Fig. 1. Two exemplar visual results of our approach: presented with an input image of an airplane or a chair, our approach is capable of reconstructing its 3D shape and skeleton, then animating its plausible articulated motions.

Compared with the more established tasks of human shape and pose estimation [40], there are nevertheless new challenges to tackle with. To name one, there is no pre-existing parametric 3D shape model for generic objects. Besides, the human template naturally comes with its skeletal configuration for 3D motion control, and the precise skinning weights designed by professionals. However, such skeletal joints are yet to be specified not to mention the skinning weights in the case of generic objects, which usually possess rather diverse geometric structures even within the same object category.

These observations have motivated us to propose an automated pipeline consisting of two stages. Stage one involves 3D shape reconstruction from a single image. It includes a transformer-based [34] encoder as the feature extractor, followed by a location occupancy prediction decoder and an auxiliary 3D voxel decoder module with improved loss function [21]. Stage two focuses on predicting the corresponding skeleton. By formulating it as estimating the multi-head probability field, a novel multi-head skeleton prediction module is proposed, inspired by the deep implicit functions of [21]. Specifically, compared with previous skeleton prediction methods with voxel-based [44] or mesh-based representations [43], our approach predict occupancy probability of joints and bones in a continuous 3D space. Moreover, a joint-aware instance segmentation module is also used as an auxiliary task to incorporate regional features of neighboring points.

Our major contributions are two folds. 1) A new object wake-up problem is considered, for which an automated pipeline is proposed to reconstruct 3D objects and their skeletons from single images. 2) A novel and effective skeleton prediction approach with a multi-head structure is developed by utilizing the deep implicit functions. Moreover, two in-house 3D datasets (SSkel & ShapeRR) of typical objects are constructed, containing annotated 3D skeletal joints and photo-realistic re-rendered images, respectively. Empirically our approach is shown to achieve promising results. Quantitative evaluations on benchmark datasets also demonstrate the superior performance of our approach on the related sub-tasks of image-based shape reconstruction and skeleton prediction.

2 Related Work

Image-based Object Reconstruction. There exist numerous studies on image-based 3D object reconstruction with various 3D shape representations, including voxel, octree [29,33,38], deep implicit function, mesh and point cloud [9,18,28,22]. Methods based on different representations have their own benefits and shortcomings. For example, as a natural extension of 2D pixels, voxel representation [10,36] has been widely used in early efforts due to its simplicity of implementation and compatibility with the convolutional neural network. However, these approaches often yield relatively coarse results, at the price of significant memory demand and high computational cost. Mesh-based representations [13,20,37,16], on the other hand, become more desirable in real applications, as they are able to model fine shape details, and are compatible with various geometry regularizers. It is however still challenging to work with topology changes [37,25]. Deep implicit 3D representations [26,6,19,35] have recently attracted wide attention as a powerful technique in modeling complex shape topologies at arbitrary resolutions.

Skeleton Prediction and Rigging. The task of skeleton prediction has been investigated in various fields and utilized in a variety of applications for shape modeling and analysis. The best-known example is the medial axis [1,2], which is an effective means for shape abstraction and manipulation. Curve skeleton or meso-skeleton [12,45] have been popular in computer graphics, mostly due to their compactness and ease of manipulation. It is worth noting the related research around detecting 3D keypoints from input point clouds, such as skeleton merger [31]. Pinocchio [3] is perhaps the earliest work on automatic rigging, which fits a pre-defined skeletal template to a 3D shape, with skinning obtained through heat diffusion. These fittings, unfortunately, tend to fail as the input shapes become less compatible with the skeletal template. On the other hand, hand-crafting templates for every possible structural variation of an input character is cumbersome. More recently, Xu et al.[44] propose to learn a volumetric network for inferring skeletons from input 3D characters, which however often suffers from the limited voxel resolution. Exploiting the mesh representation, RigNet [43] utilizes a graph neural network to produce the displacement map for joint estimation, which is followed by the additional graph neural networks to predict joint connectivity and skinning weights. Its drawback is they assume strong requirements for the input mesh such as a watertight surface with evenly distributed vertices can be satisfied. Besides, they predict the joints and kinematic chains successively causing error propagation from stages. In contrast, a deep implicit function representation [21] which is capable of predicting the joints and bones over a continuous 3D space is considered in this paper for inferring skeleton.

Image based Object Animation. An established related topic is photo editing, which has already been popular with professional tools such as PhotoShop. Existing tools are however often confined to 2D object manipulations in performing basic functions such as cut-and-paste and hole-filling. A least-square method is considered in [30] to affine transform objects in 2D. The work of [11] goes beyond linear transformation, by presenting an as-rigid-as-possible 2D animation of a human character from an image, it is however manual intensive. In [42], 2D instances of the same visual objects are ordered and grouped to form an instance-based animation of non-rigid motions. Relatively few research activities concern 3D animations, where the focus is mostly on animals, humans, and human-like objects. For example, photo wake-up [40] considers reconstruction, rig, and animate 3D human-like shapes from input images. This line of research benefits significantly from the prior work establishing the pre-defined skeletal templates and parametric 3D shape models for humans and animals. On the other hand, few efforts including [15,5] consider 3D manipulations of generic objects from images, meanwhile, they mainly focus on rigid transformations. Our work could be regarded as an extension of automated image-based human shape reconstruction & animation to reconstruct & articulate generic lifeless objects from single images.

3 Our Approach

Given an input image, usually in the form of a segmented object, first the 3D object shape is to be reconstructed; its skeletons are then extracted to form a rigged model. In this section, we will present the stage-wise framework in detail.

3.1 Image-based 3D Shape Reconstruction

A Transformer-based occupancy prediction network is developed here, which performs particularly well on real images when compared with existing methods [21,41,17]. As illustrated in Fig. 2, it consists of a 2D transformer encoder, an auxiliary 3D CNN decoder, and an occupancy decoder. The DeiT-Tiny [34] is used as our transformer encoder network. Similar to the Vision Transformer [8], the encoder first encodes fixed-size patches splitted from the original image and processes extract localized information from each of the patches, then outputs a universal latent representation for the entire image by jointly learning the patch representation with multi-head attention. An auxiliary 3D CNN decoder

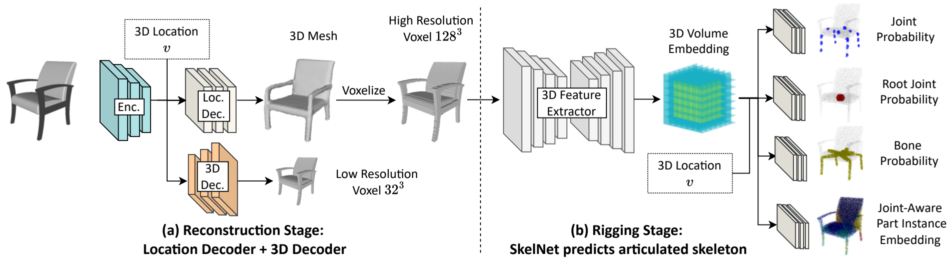

Fig. 2. An illustration of our overall pipeline. (a) a DeiT image encoder, an auxiliary 3D CNN voxel prediction branch and the location occupancy decoder; (b) SkelNet accepts a high resolution 3D shape voxel based on the reconstructed 3D mesh, and predicts articulated skeleton with a multi-head architecture.

is used for reconstructing a low-resolution voxel-based 3D model as well as helping to encode 3D information for the latent representation extracted from the Transformer encoder. The occupancy decoder then uses the latent representation as the conditional prior to predict the occupancy probability for each point by introducing fully connected residual blocks and conditional batch normalization [27,24].

It is worth noting that although the voxel prediction branch is only used for auxiliary training, the highly unbalanced labels where most of the voxel occupancy are zeros will always make the training more difficult. To this end, while most of the methods for voxel-based 3D reconstruction simply use the (binary) cross-entropy loss which is directly related to IoU metric [32], in this work, the Dice loss is extended to gauge on both the 3D voxel prediction and the point-based occupancy prediction,

$$\mathcal{L}_{dice} = 1 - \frac{\sum_{n=1}^{N^3} \hat{y}_n y_n}{\sum_{n=1}^{N^3} \hat{y}_n + y_n} - \frac{\sum_{n=1}^{N^3} (1-\hat{y}_n)(1-y_n)}{\sum_{n=1}^{N^3} 2 - \hat{y}_n - y_n}, \tag{1}$$

where y_n is the ground-truth occupancy score, $\hat{y}_n$ is the predicted occupancy score of the n-th element.

3.2 Skeleton Prediction and Automatic Rigging

Our key insight here is instead of predicting the joints inside fixed voxel locations [44] or indirectly regressing the joints location by estimating the displacement from the mesh [43], we train a neural network utilizing the deep implicit function to assign every location with a probability score in $[0, 1]$, indicating the existence of a skeletal joint and bone. Taking the 3D model and any sampled 3D point location as input, the network produces the joint and bone existence probabilities. In addition, we incorporate joint-aware instance segmentation as an auxiliary task considering the regional features over neighboring points. In inference, the feature embedding output from the instance segmentation branch is further used in the subsequent step to infer joint locations from the incurred joints' probability maps.

As in Fig. 2, four output heads are utilized, which are for predicting the probability of skeletal joints, the root joint, the bones, and the joint-aware instance segmentation, respectively. The output from the instance segmentation is a feature embedding.

Feature Extraction. The predicted 3D shape, represented as an occupancy grid with the dimension of 128^3, is converted to a 3D feature embedding grid by a 3D UNet structure. Inspired by the design of Squeeze and Excitation (SE) block in 2D image classification, a 3D adaptive channel activation module is developed as a plug-in module, to be attached after each of the encoder and decoder blocks of the 3D UNet, detailed design is described in the supplementary. The ablative study demonstrated the usefulness of this 3D adaptive channel activation module.

Multi-head Implicit Functions. Given aggregated features from the feature extraction, we acquire the feature vector for any 3D point v via the trilinear interpolation from 3D feature embedding. For each of the output heads, a fully-connected network (empirically it is implemented as 5 fully-connected ResNet blocks and ReLU activation [27,24]) is engaged to take as input the point v and its feature vector. The concurrent multi-head strategy eliminates the possible issue with error propagation of successive prediction [43].

Sampling. In general, the animation joints and bones should lie inside the convex hull of the object. Therefore, different from previous efforts that uniformly sample points in a 3D volume [21,27], points in our 3D space are adaptively sampled. Specifically, for each sample in the training batch, we sampled K points with 10% of the points lying outside but near the surface, and the rest 90% points entirely inside the object.

Joints and Bones Loss. First, for every query point, its joint probability is computed under a 3D Gaussian distribution measured by its distance to nearest annotated joint locations. To generate the bone probability field, for every query point we compute a point-to-line distance to its nearest line segment of the bones, and the bone probability is computed under the Gaussian distribution of the measured distance. In training, with the query points $v \in \mathbb{R}^3$ acquired through sampling, the network predicts their probabilities of being a joint or lying on bones. Different from the occupancy prediction [21] task where the binary cross-entropy loss is used, we use the L1 loss to measure the difference of the predicted joint probability and their ground-truth values as we are dealing with the continuous probability prediction: for the i-th sample in training, the loss function is defined as,

$$\begin{aligned}\mathcal{L}^i_{joint}(\hat{P}_J, P_J) &= \sum_{v \in \mathcal{V}^i} |\hat{P}_J(v) - P_J(v)| \\ \mathcal{L}^i_{jointR}(\hat{P_{JR}}, P_{JR}) &= \sum_{v \in \mathcal{V}^i} |\hat{P_{JR}}(v) - P_{JR}(v)|\end{aligned} \tag{2}$$

In the above equation, $\hat{P_J}$ is the predicted joints probability field, and P_J is the ground-truth probability field. $\hat{P_{JR}}$ and P_{JR} denote for the probability field of the root joint. $\mathcal{V}^i$ denotes the sampled points for the i-th model.

Similarly, for the sampled points, L1 loss is also applied between predicted bones probability $\hat{P_B}$ and the ground-truth P_B. The loss function of the bones is denoted as $\mathcal{L}^i_{bone}(\hat{P_B}, P_B)$.

Symmetry Loss. Since the objects of interest often possess symmetric 3D shapes, a symmetry loss is used here to regularize the solution space, as follows,

$$\mathcal{L}^i_{sym}(\hat{P_J}, \hat{P_B}) = \mathbf{1}_{\Omega'}(i) \sum_{v \in \mathcal{V}^i} |\hat{P_J}(v) - \hat{P_J}(\phi(v))| + \mathbf{1}_{\Omega'}(i) \sum_{v \in \mathcal{V}^i} |\hat{P_B}(v) - \hat{P_B}(\phi(v))|, \tag{3}$$

Here $\phi(v)$ denotes the mapping from point v to its symmetric point. To detect the symmetry planes, as the input 3D mesh models are in the canonical coordinates, we flip the mesh model according to the xy-, xz- and yz-planes. The symmetry plane is set as the one with the smallest Chamfer distance computed between the flipped model and the original model. $\mathbf{1}_{\Omega'}$ is an indicator function where Ω' is the subset of training models with symmetry planes detected.

Joint-aware Instance Segmentation Loss. The joint-aware instance segmentation maps the sampled point from Euclidean space to a feature space, where 3D points of the same instance are closer to each other than those belonging to different instances. To maintain consistency between the clustered feature space and the joints probability maps, the part instance is segmented according to the annotated ground-truth joints. Basically, for each sampled point we assign an instance label as the label or index of its closest joint. Following the instance segmentation method of [39], our joint-aware instance segmentation loss is defined as a weighted sum of three terms: (1) $\mathcal{L}_{var}$ is an intra-cluster variance term that pulls features belonging to the same instance towards the mean feature; (2) $\mathcal{L}_{dist}$ is an inter-cluster distance term that pushes apart instances with different part labels; and (3) $\mathcal{L}_{reg}$ is a regularization term that pulls all features towards the origin in order to bound the activation.

$$\begin{aligned}
\mathcal{L}^i_{var}(\mu, x) &= \frac{1}{|J^i|} \sum_{c=1}^{|J^i|} \frac{1}{N_c} \sum_{j=1}^{N_c} [\ \|\mu^i_c - x^i_j\| - \delta_{var}]\ ^2_+, \\
\mathcal{L}^i_{dist}(\mu) &= \frac{1}{|J^i|(|J^i|-1)} \sum_{c_a=1}^{|J^i|} \sum_{\substack{c_b=1 \\ c_b \neq c_a}}^{|J^i|} [\ 2\delta_{dist} - \|\mu^i_{c_a} - \mu^i_{c_b}\|]\ ^2_+, \\
\mathcal{L}^i_{reg}(\mu) &= \frac{1}{|J^i|} \sum_{c=1}^{|J^i|} \|\mu^i_c\|.
\end{aligned} \tag{4}$$

Here $|J^i|$ denotes the number of joints or clusters for the i-th sample model. N_c is the number of elements in cluster c. x^i_j is the output feature vector for the query point. $[\ x]_+$ is the hinge function. The parameter δ_{var} describes the maximum allowed distance between a feature vector and the cluster center. Likewise, $2\delta_{dist}$ is the minimum distance between different cluster centers to avoid overlap.

Joints and Kinematic Tree Construction. In inference, the joints and bones are obtained from the corresponding probability maps by mean-shift clustering. Instead of clustering over the euclidean space as in classical mean-shift clustering, we implement the clustering on the feature space with the kernel defined over the feature embedding output from the joint-aware instance segmentation. In this way, the points belonging to the same joint-aware instance will all shift towards the corresponding joints. The kernel is also modulated by the predicted joint probability to better localize the joint location. Mathematically, at each mean-shift iteration, for any point v it is displaced according to the following vector:

$$m(v) = \frac{\sum_{u \in \mathcal{N}(v)} P_J(v)\kappa(\|x(u) - x(v)\|)u}{\sum_{u \in N(v)} P_J(v)\kappa(\|x(u) - x(v)\|)} - v \tag{5}$$

where $\mathcal{N}(v)$ denotes the neighboring points of v, $x(v)$ is the feature embedding output from our joint-aware instance segmentation. Besides, $\kappa()$ is a kernel function and in our case we choose to use the RBF kernel. Following [44], the object kinematic tree (or chains) are constructed using a minimum spanning tree by minimizing a cost function defined over the edges connecting the joints pairwisely. It is realized as a graph structure, with the detected joints as the graph nodes, and the edges connecting the pairwise joints computed from the probability maps. Specifically, for every edge, its weight is set by the negative-log function of the integral of the bones probability for the sampled points over the edge. The MST problem is solved using Prim's algorithm [7].

Skinning Weight Computation. For automatic rigging of the reconstructed 3D model, the last issue is to compute the skinning weights that bind each vertex to the skeletal joints. To get meaningful animation, instead of computing the skinning weights according to the Euclidean distance [3], we choose to assign the skinning weights by utilizing the semantic part segmentation [39]. Specifically, for every segmented part, we assign its dominant control joint to the one closest to the center of the part. In some cases where the center of the part could have about the same distance to more than one skeletal joint, we choose the parent joint as the control joint. The skinning weights around the segmentation boundaries are smoothed out afterwards. It is worth noting that some semantic parts are further segmented if skeleton joints are detected inside the part.

3.3 Our In-house Datasets

As there is no existing dataset of general 3D objects with ground-truth skeletons, we collect such a dataset (named SSkel for *ShapeNet skeleton*) by designing an annotation tool to place joints and build kinematic trees for the 3D shapes. To ensure consistency, a predefined protocol is used for all object categories. For example, for chairs, we follow the part segmentation in PartNet dataset [23] to segment a chair into the chair seat, back, and legs. The root joint is annotated at the center of the chair seat, followed by child joints which are the intersection between chair seat and back, chair seat and legs. More details about the annotation tool and some sampled annotations are presented in the supplementary. Without loss of generality, we only consider four categories of objects from ShapeNet [4], namely *chair*, *table*, *lamp* and *airplane*. Our SSkel dataset contains a total of 2,150 rigged 3D shapes including 700 for chair, 700 for table, 400 for lamp and 350 for airplane.

Moreover, in improving the input image resolution and quality of the original ShapeNet, we use the UNREAL 4 Engine to re-render photo-realistic images of the 3D ShapeNet models with diverse camera configuration, lighting conditions, object materials, and scenes, named ShapeRR dataset for *ShapeNet of realistic rendering*. More details are relegated to the supplementary file.

4 Experiments

Datasets. A number of datasets are considered in our paper. In terms of image-based reconstruction, it contains our ShapeRR dataset for synthetic images and the Pix3D dataset of real images. In terms of rigging performance, we use the RigNetv1 dataset for 3D shape-based rigging, and our SSkel dataset for image-based rigging.

	Chamfer Distance (↓)					Volumetric IoU (↑)				
ShapeNet	Chair	Table	Lamp	Airplane	Avg.	Chair	Table	Lamp	Airplane	Avg.
OccNet [21]	1.9347	1.9903	4.5224	1.3922	2.3498	0.5067	0.4909	0.3261	0.5900	0.4918
DVR [24]	1.9188	2.0351	4.7426	1.3814	2.5312	0.4794	0.5439	0.3504	0.5741	0.5029
D^{2}IM-Net [17]	**1.8847**	1.9491	4.1492	1.4457	2.0346	**0.5487**	0.5332	0.3755	0.6123	0.5231
Ours	1.8904	**1.7392**	**3.9712**	**1.2309**	**1.9301**	0.5436	**0.5541**	**0.3864**	**0.6320**	**0.5339**
Pix3D	Table	Chair	Desk	Sofa	Avg.	Table	Chair	Desk	Sofa	Avg.
OccNet [21]	7.425	9.399	15.726	14.126	11.625	0.215	0.201	0.143	0.152	0.190
DVR [24]	8.782	6.452	12.826	11.543	9.901	0.187	0.237	0.165	0.187	0.185
D^{2}IM-Net [17]	8.038	7.592	11.310	9.291	9.057	0.205	0.244	0.183	0.207	0.215
Ours	**6.449**	**6.028**	**8.452**	**8.201**	**7.282**	**0.239**	**0.277**	**0.219**	**0.241**	**0.242**

Table 1. Image-based 3D mesh reconstruction on ShapeRR (i.e. re-rendered ShapeNet dataset) and Pix3D dataset. Metrics are Chamfer Distance ($\times 0.001$, the smaller the better) and Volumetric IoU (the larger the better). Best results are in **bold face**.

The Pix3D dataset contains 3D object shapes aligned with their real-world 2D images. Similar to ShapeRR, we focus on a subset of 4 categories in the dataset, i.e. chair, sofa, desk, and table. The RigNetv1 dataset (i.e. ModelsResource-RigNetv1 [44]), on the other hand, contains 2,703 rigged 3D characters of humanoids, quadrupeds, birds, fish, robots, and other fictional characters.

4.1 Evaluation on Image-based Reconstruction

For evaluation metrics, we follow the previous works [21] and use volumetric IoU and Chamfer-L1 distance. We first compare with several state-of-the-art methods with released source code on single image object reconstruction where each of the methods is trained and tested, namely OccNet [21], DVR [24] and D^{2}IM-Net [17]. We follow the common test protocol on ShapeNet as it has been a standard benchmark in the literature. All methods are re-implemented (when the code is not available) and re-trained then evaluated directly on the test split. We can observe that our method performs reasonably well compared with other recent methods, and outperforms existing methods in 3 of the 4 categories. And we are able to achieve a significant advantage over other methods in terms of the average performance across all 4 categories of our interests.

Considering that our 3D reconstruction is primarily for supporting rigging and animation purposes on real images, to better compare the generalization ability with such a situation, we use the complete Pix3D dataset as the test set.

We report both quantitative and visual comparison on Pix3D in Tab. 1 and in Fig. 3 respectively. As shown in Tab. 1, our proposed method has outperformed all previous approaches on Pix3D with a large margin in terms of the two metrics. To validate the effectiveness of our feature encoder and the incorporated auxiliary voxel prediction task, we also conduct a group of ablative studies, and the experiment results are included in the supplementary material.

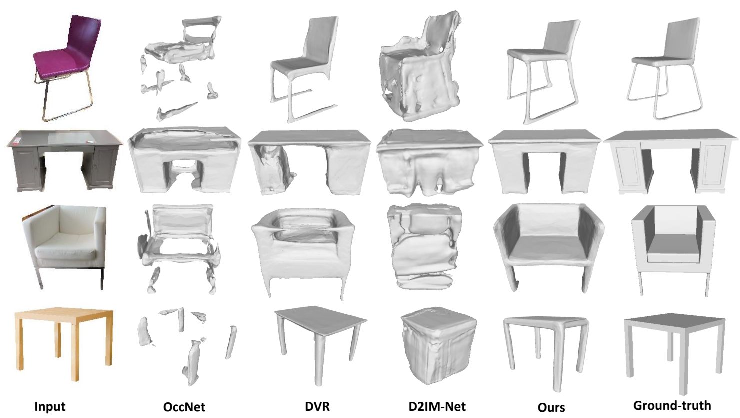

Fig. 3. Visualization of image-based 3D reconstruction on the Pix3D dataset. Our method shows excellent generalization performance on the real images.

4.2 Evaluation on Skeleton Prediction

The evaluation is conducted on both the RigNetv1 dataset and our SSkel dataset, where our approach is compared with two state-of-the-art methods, RigNet [43] and VolumetricNets [44].

Metrics. First, we measure the accuracy of the predicted joints by computing the Chamfer distance between the predicted joints and the ground-truth which is denoted as CD-J2J. Similarly, the predicted bones are evaluated by computing the Chamfer distance between the densely sampled points over the estimated bones and the ground-truth, which is denoted as CD-B2B. CD-J2B is also considered here by computing the Chamfer distance between predicted joints to bones. For all metrics, the lower the better.

	CD-J2J (↓)	CD-J2B (↓)	CD-B2B (↓)
Pinocchino [3]	0.072	0.055	0.047
Volumetric [44]	0.045	0.029	0.026
RigNet [43]	0.039	0.024	0.022
Ours	**0.029**	**0.019**	**0.017**

Table 2. Comparison of skeleton prediction on the RigNetv1 dataset.

Quantitative evaluation. In Tab. 2 we show the comparison results of the predicted skeleton on the RigNetv1 dataset [44]. For the RigNetv1 dataset, we follow the same train and test split as previous works [44,43]. In Tab. 3 we show the quantitative evaluation and comparison results of the predicted skeleton on our SSkel dataset. We have re-trained the RigNet [43], which is the most current work on auto-rigging, on our SSkel dataset. As shown in the tables, our proposed skeleton prediction method has outperformed the current state-of-the-art approaches with the smallest error on all reported metrics on both the RigNetv1 dataset and our SSkel dataset.

	——Chair——			——-Table——-			——Lamp——			——Airplane——			——Average——		
metrics	J2J	J2B	B2B	J2J	J2B	B2B	J2J	J2B	B2B	J2J	J2B	B2B	J2J	J2B	B2B
RigNet-GT	0.052	0.042	0.035	0.061	0.049	0.040	0.132	0.110	0.098	0.096	0.081	0.073	0.061	0.046	0.041
Ours-GT	**0.030**	**0.023**	**0.021**	**0.044**	**0.032**	**0.028**	**0.097**	**0.071**	**0.063**	**0.075**	**0.062**	**0.056**	**0.047**	**0.038**	**0.033**
RigNet-rec	0.048	0.035	0.033	0.060	0.046	0.038	0.143	0.116	0.102	0.103	0.084	0.076	0.063	0.047	0.042
Ours-rec	**0.036**	**0.024**	**0.022**	**0.047**	**0.033**	**0.029**	**0.101**	**0.073**	**0.065**	**0.081**	**0.065**	**0.059**	**0.051**	**0.041**	**0.036**

Table 3. Quantitative comparison of skeleton prediction on our SSkel dataset. The J2J, J2B, B2B are the abbreviation for CD-J2J, CD-J2B and CD-B2B respectively. For these values, the smaller the better. Best results are in **bold face**.

It is worth noting that the evaluation on the SSkel dataset is conducted with two different inputs. First, we report the skeleton error(RigNet-GT, Ours-GT) when taking the ground-truth 3D models as input. To evaluate the performance of the overall pipeline, we also calculate the skeleton error(RigNet-rec, Ours-rec) when 3D models reconstructed from the color images are taken as input. Our skeleton prediction performance on the reconstructed 3D models degraded slightly due to imperfect reconstruction.

Visual results on skeleton prediction. In Fig. 4 and Fig. 5 we demonstrate the qualitative comparison of the predicted skeleton. First, in Fig. 4, the

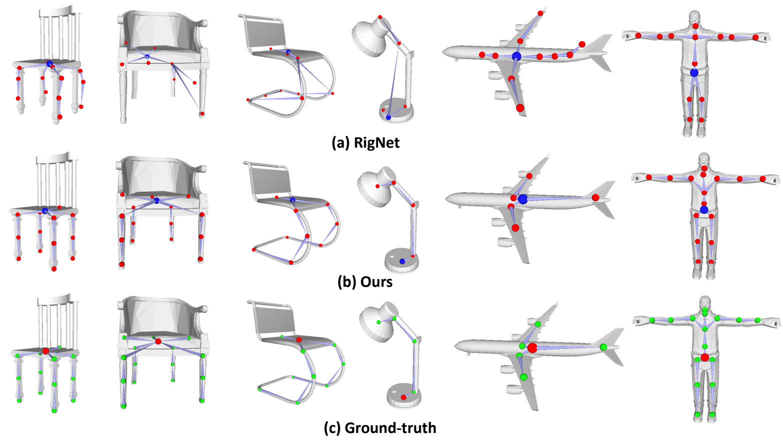

Fig. 4. Visual comparison on skeleton prediction. The rightmost model comes from the RigNetv1 dataset and the others are from our SSkel dataset.

skeletons are predicted with ground-truth 3D models as input. We also evaluated the overall pipeline when taking a single image as input, and the results are shown in Fig. 5. As shown in the figures, compared with the most current work, our proposed approach can produce more reasonable results that correctly predicted the joints' location and constructed the kinematic chains. On the other hand, the RigNet method fails to localize the joints. The reason is that their mesh-based approach requires the vertices to be evenly distributed over the mesh and they rely on the mesh curvature to pre-train an attention model. But for the models from the SSkel dataset, there is no close connection between the mesh curvature and the joint locations.

	RigNetv1	SSkel
Baseline	0.037	0.065
Baseline + joint-aware seg	0.033	0.055
Baseline + symmetry loss	0.034	0.058
Baseline + 3D adaptive activation	0.033	0.056
Ours	**0.029**	**0.047**

Table 4. Ablation study on joints prediction. CD-J2J metric is used.

Ablation study. To validate the effectiveness of several key components of the proposed method, we conduct several ablation studies with the quantitative evaluation results shown in Table 4. We denote our method without the 3D channel adaptive activation, symmetry loss, and joint-aware instance segmentation as the Baseline method.

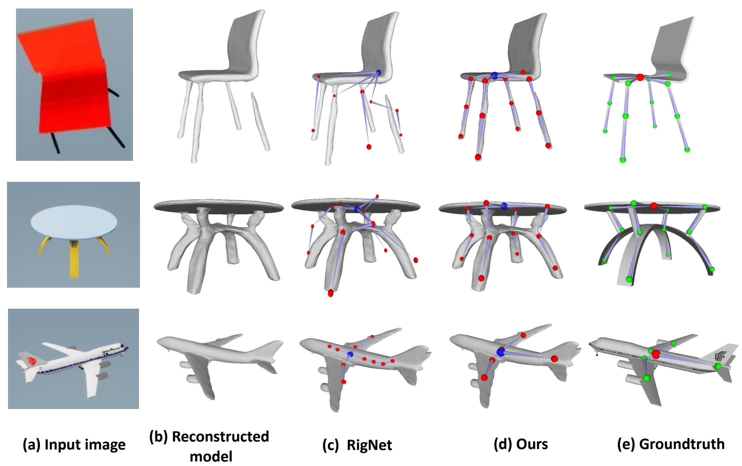

Fig. 5. Visual results on articulated 3D models from input images. Taking the color image (a) as input, we reconstruct the 3D model (b) and predict its skeleton (d), and also compare with the RigNet [43] on skeleton prediction (c).

4.3 Applications on Animation

After obtaining the rigged 3D models from the input images, in this section, we present interesting applications of animating the rigged 3D objects. To get the texture for the 3D models, similar to [41] we have trained a deep neural network to predict the projection matrix represented as a 6D rotation vector aligning the 3D models from canonical space to image space. Our reconstructed 3D model is further refined and deformed according to the object silhouettes [40]. The mirrored texture is applied to the invisible part of the 3D model.

In Fig. 6, we demonstrate the animation of objects as driven by the source motion of reference articulated models. Specifically, in the upper rows of Fig. 6 we map the motion of a Jumping human to two Chairs as well as a Lamp. The details of the skeleton mapping from the human template to the animated objects are shown in each corresponding row of Fig. 6(d). Likewise, in the lower part of Fig. 6, we demonstrate the manipulation of a Chair and Table driven by a quadruped. It is conducted by mapping the joints of four legs on the Dog skeleton to the legs of the chair and table. In addition, the joint of the neck is mapped to the joint on the chair back. The motion sequence of the dog is from RGBD-Dog dataset [14]. More results can be seen in the supplementary video.

5 Conclusion and Limitations

We consider an interesting task of waking up a 3D object from a single input image. An automated pipeline is proposed to reconstruct the 3D object, predict the articulated skeleton to animate the object with plausible articulations.

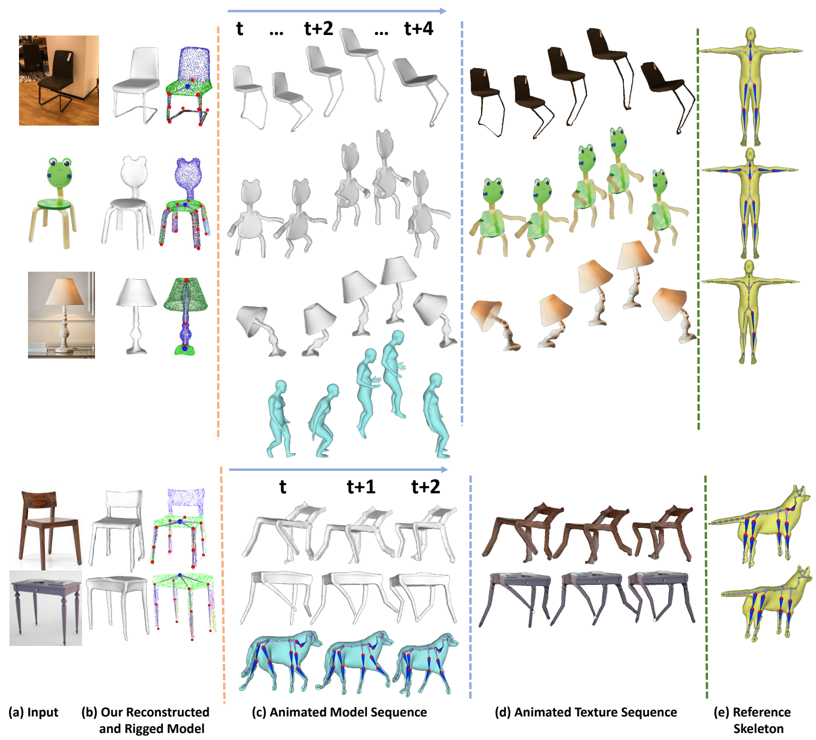

Fig. 6. Object animation. Given an input image (i.e. the object segment), its 3D shape is reconstructed and rigged, followed by the animated sequence (re-targeted from human or quadruped motions, which is not the main focus of this work). We map the joints from the human or quadruped skeleton to the objects, and the mapped joints are marked in red (c). The source human/dog motion is shown in the bottom row.

Quantitative and qualitative experiments demonstrate the applicability of our work when unseen real-world images are presented at test time.

Limitations. First, the domain gap between synthetic to real images still exists. Second, in our current stage-wise framework, the skeleton prediction and final animation rely on the success of 3D shape reconstruction. For future work, we would like to combine shape reconstruction and skeleton prediction in a unified network structure to facilitate each task. Moreover, the collected SSkel dataset is limited in the number of objects and the range of object categories. For future work, we plan to work with a large-scale dataset containing a much broader range of generic object categories.

ACKNOWLEDGEMENTS

This research was partly supported by the NSERC Discovery, CFI-JELF and UAHJIC grants. We also thank Priyal Belgamwar for her contribution to the dataset annotation.

References

1. Amenta, N., Bern, M.: Surface reconstruction by voronoi filtering. Discrete & Computational Geometry **22**(4), 481–504 (1999)
2. Attali, D., Montanvert, A.: Computing and simplifying 2D and 3D continuous skeletons. Computer Vision and Image Understanding **67**(3), 261–273 (1997)
3. Baran, I., Popović, J.: Automatic rigging and animation of 3d characters. ACM Transactions on graphics(TOG) **26**(3), 72 (2007)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, T., Zhu, Z., Shamir, A., Hu, S.M., Cohen-Or, D.: 3-sweep: Extracting editable objects from a single photo. ACM Transactions on Graphics (TOG) **32**(6), 1–10 (2013)
6. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
7. Cheriton, D., Tarjan, R.E.: Finding minimum spanning trees. SIAM Journal on Computing **5**(4), 724–742 (1976)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
9. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
10. Gkioxari, G., Malik, J., Johnson, J.: Mesh R-CNN. In: IEEE/CVF International Conference on Computer Vision (2019)
11. Hornung, A., Dekkers, E., Kobbelt, L.: Character animation from 2d pictures and 3d motion data. ACM Transactions on Graphics(TOG) **26**(1), 1 (2007)
12. Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., Chen, B.: L1-medial skeleton of point cloud. ACM Transactions on Graphics (TOG) **32**(4), 65–1 (2013)
13. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
14. Kearney, S., Li, W., Parsons, M., Kim, K.I., Cosker, D.: Rgbd-dog: Predicting canine pose from rgbd sensors. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8336–8345 (2020)
15. Kholgade, N., Simon, T., Efros, A., Sheikh, Y.: 3d object manipulation in a single photograph using stock 3D models. ACM Transactions on Graphics (TOG) **33**(4), 1–12 (2014)
16. Kulon, D., Guler, R.A., Kokkinos, I., Bronstein, M.M., Zafeiriou, S.: Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
17. Li, M., Zhang, H.: D2im-net: Learning detail disentangled implicit fields from single images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10246–10255 (2021)
18. Lin, C.H., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3D object reconstruction. In: AAAI Conference on Artificial Intelligence (2018)
19. Lin, C.H., Wang, C., Lucey, S.: Sdf-srn: Learning signed distance 3D object reconstruction from static images. In: Advances in Neural Information Processing Systems (2020)

20. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: IEEE/CVF International Conference on Computer Vision (2019)
21. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
22. Mi, Z., Luo, Y., Tao, W.: Ssrnet: scalable 3D surface reconstruction network. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
23. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
24. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
25. Pan, J., Han, X., Chen, W., Tang, J., Jia, K.: Deep mesh reconstruction from single rgb images via topology modification networks. In: IEEE International Conference on Computer Vision (2019)
26. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
27. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: European Conference on Computer Vision (ECCV). pp. 523–540 (2020)
28. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems (2017)
29. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3D representations at high resolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
30. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. ACM Transactions on Graphics(TOG) **25**(3), 533–540 (2006)
31. Shi, R., Xue, Z., You, Y., Lu, C.: Skeleton merger: an unsupervised aligned keypoint detector. In: IEEE conference on computer vision and pattern recognition (2021)
32. Shi, Z., Meng, Z., Xing, Y., Ma, Y., Wattenhofer, R.: 3d-retr: End-to-end single and multi-view 3D reconstruction with transformers. In: The British Machine Vision Conference (2021)
33. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: IEEE International Conference on Computer Vision (2017)
34. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers and distillation through attention. In: International Conference on Machine Learning. vol. 139, pp. 10347–10357 (2021)
35. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C.: Patchnets: Patch-based generalizable deep implicit 3d shape representations. In: European Conference on Computer Vision (2020)
36. Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)

37. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: European Conference on Computer Vision (2018)
38. Wang, P.S., Liu, Y., Tong, X.: Deep octree-based cnns with output-guided skip connections for 3D shape and scene completion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2020)
39. Wang, W., Yu, R., Huang, Q., Neumann, U.: Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2569–2578 (2018)
40. Weng, C.Y., Curless, B., Kemelmacher-Shlizerman, I.: Photo wake-up: 3d character animation from a single photo. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5908–5917 (2019)
41. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In: Advances in Neural Information Processing Systems (2019)
42. Xu, X., Wan, L., Liu, X., Wong, T.T., Wang, L., Leung, C.S.: Animating animal motion from still **27**(5), 1–8 (2008)
43. Xu, Z., Zhou, Y., Kalogerakis, E., Landreth, C., Singh, K.: Rignet: Neural rigging for articulated characters. ACM Transactions on Graphics(TOG) **39**(58) (2020)
44. Xu, Z., Zhou, Y., Kalogerakis, E., Singh, K.: Predicting animation skeletons for 3d articulated models via volumetric nets. In: International Conference on 3D Vision. pp. 298–307 (2019)
45. Yin, K., Huang, H., Cohen-Or, D., Zhang, H.: P2p-net: Bidirectional point displacement net for shape transform. ACM Transactions on Graphics (TOG) **37**(4), 1–13 (2018)