AutoSweep: Recovering 3D Editable Objects from a Single Photograph

Xin Chen Yuwei Li Xi Luo Tianjia Shao Jingyi Yu Kun Zhou Youyi Zheng†

Abstract—This paper presents a fully automatic framework for extracting editable 3D objects directly from a single photograph. Unlike previous methods which recover either depth maps, point clouds, or mesh surfaces, we aim to recover 3D objects with semantic parts and can be directly edited. We base our work on the assumption that most human-made objects are constituted by parts and these parts can be well represented by generalized primitives. Our work makes an attempt towards recovering two types of primitive-shaped objects, namely, generalized cuboids and generalized cylinders. To this end, we build a novel instance-aware segmentation network for accurate part separation. Our GeoNet outputs a set of smooth part-level masks labeled as profiles and bodies. Then in a key stage, we simultaneously identify profile-body relations and recover 3D parts by sweeping the recognized profile along their body contour and jointly optimize the geometry to align with the recovered masks. Qualitative and quantitative experiments show that our algorithm can recover high quality 3D models and outperforms existing methods in both instance segmentation and 3D reconstruction.

Index Terms—Editable objects, instance-aware segmentation, sweep surfaces.

1 Introduction

There is an emerging demand on automatic extraction of high quality 3D objects from a single photograph. Applications are numerous, ranging from image manipulation [1], [2], [3], to emerging 3D printing [4], [5] and virtual reality and augmented reality [6], [7]. For example, in e-commerce, it is highly desirable to automatically and quickly recover the 3D model of a commercial product from its 2D image (e.g., in advertisement). Further, the geometry and the texture map should be of high quality to be useful. The problem, however, remains challenging: any successful solution should be able to reliably segment an object from the image and then recover its shape and structure whereas both problems are ill-posed and generally require imposing priors and using sophisticated optimization.

A photograph is inherently "flat" and does not contain associated depth information. Traditional solutions rely on multi-view stereo or volumetric reconstructions to recover the point cloud, normal, or visual hull of the object. They require using multiple images of an object which is most likely inaccessible in applications such as e-commerce. More importantly, the recovered 3D geometry is of low quality even with the most advanced reconstruction algorithms. Alternative solutions [2], [8], [9], [10] treat an object as a composition of simple, primitive components [11], [12] and set out to estimate each individual component. Most existing methods in this category require extensive human inputs for partitioning the object. Most recently, end-to-end methods [13], [14] have leveraged generative neural networks to directly infer point cloud or volumetric representations of an object from a single image. They are able to produce coarse geometry that resembles the actual shape. Yet, the quality of the resulting model still barely meet the one of a CAD model or a parametric mesh.

In this paper, we present a fully automatic, single-image based technique for producing very high quality 3D geometry of a specific class of objects: objects composed of generalized cuboids and generalized cylinders or *GC-GCs*, for short. Both a generalized cuboid and a generalized cylinder could be represented as a profile (i.e., a circle or a rectangle) sweeping along a trajectory axis as in traditional CAD systems. Normally, the profile is allowed to scale and the trajectory axis is curved [8]. An intriguing benefit of our reconstruction pipeline is that each cuboid and cylindrical part can be directly edited by altering the profile or the trajectory axis and then composed together to form a new GC-GC. See Fig. 1, 5 for examples of GC-GCs.

In our solution, we first partition and recognize each semantic part of a GC-GC object. We exploit instance segmentation network Mask R-CNN [15] which is capable of handling "invisible profiles" that caused by occlusions of the foreground or even self occlusions. However, due to a small receptive field, the output often contains erroneous boundaries and incomplete masks that do not agree with the actual object mask. We extend the structure of Mask R-CNN and construct our Geometry Network (*GeoNet* for short) by incorporating contour and edge maps into a concatenating network which we call the deformable convolutional

Fig. 1: Exemplar 3D models generated using our method.

† *corresponding author*

- *X. Chen, Y. Li, X. Luo, and J. Yu are with School of Information Science and Technology,ShanghaiTech University, China. E-mail: {chenxin,liyuwei,luoxi,jingyiyu1}@shanghaitech.edu.cn*
- *T. Shao is with the School of Computing, University of Leeds, UK. E-mail: T.Shao@leeds.ac.uk*
- *Y. Zheng, and K. Zhou are with the State Key Lab of CAD&CG, Zhejiang University, China. E-mail: zyy, kunzhou@cad.zju.edu.cn*

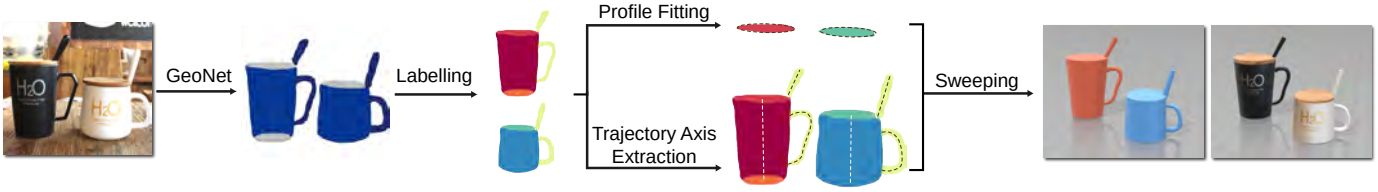


Fig. 2: The pipeline. Our method takes as input a single photograph and extracts its semantic part masks labeled as *cylinder profile*, *cuboid profile*, *cylinder body*, etc., which are then used in a sweeping procedure to construct a textured 3D model.

network (DCN) derived from [16], [17]. The edge maps and 2D contours are used to better learn the boundary of the body and face regions which are crucial in the subsequent modeling process. Our network outputs smooth masks around the boundary regions.

Once we segment each component, we conduct reconstruction via a volume sweeping scheme. We decouple the process into two stages of profile fitting and profile sweeping. To estimate the 3D profile, we jointly optimize the profile with the camera pose. We then extract the trajectory axis of each body mask and map it to 3D with the estimated camera pose to guide the optimization of the profile sweeping.

We demonstrate our approach to various images. Our system is capable of automatically generating 3D models from a single photograph which can then be used for editing and rearranging. Qualitative and quantitative experiments are conducted to verify the effectiveness of our method.

2 Related Work

Semantic Segmentation. Recent deep neural networks have shown great success in improving traditional classification and semantic segmentation tasks. The classifier in the fully convolutional networks (FCNs) [18] can conduct inference and learning on an arbitrary sized image but does not directly output individual object instances. Mask R-CNN [15] extends Faster R-CNN [19] by adding a branch for predicting object masks on top of bounding box extraction. [20], [21] use a multi-task cascaded structure to identify instances with position-sensitive score maps. FCIS [22] proposed inside and outside maps to preserve the spatial extent of the original image. [23] observed that the large receptive fields and the amount of pooling layers of these networks can degrade the quality of instance masks, causing aliasing effect. Region proposal network (RPN) [19] can only capture the rough shape of the object and its extensions [23], [24], [25] aim to improve the segmentation boundary.

Single-Image Depth Estimation. Classic methods on monocular depth estimation mainly relied on hand-crafted features and graphical models [26], [27]. More recently, several learning-based approaches boost the performance by utilizing deep models. [28] employed a multi-scale deep network with two stacks for both global and local prediction to achieve depth estimation on a single image. [29] used CNN for simultaneous depth estimation and semantic segmentation. However, these works only seek to obtain the relative 3D relationship between different layers and therefore their depth results are much less accurate and clearly insufficient for high quality reconstruction. In our reconstruction, the surface is highly curved but smooth and therefore the depth map needs to be at an ultra-high accuracy, which is extremely difficult to achieve even under the stereo setting, let alone single-image.

Single-Image 3D Reconstruction. Recovering 3D shape from a single image is a long standing problem in computer vision [11], stemming from image metrology [30], [31]. The problem is inherently ill-posed and tremendous efforts have focused on imposing constraints such as geometric priors [9], [32], symmetry [10], [33], [34], planarity constraints [35], shape priors [36], [37], etc., or relying on stock 3D models for 2D-3D alignments [3], [38], [39], [40]. Latest approaches leverage deep learning techniques [41], [42], [43] on large datasets. Eigen *et al*. [44] infer depth maps using a multi-scale deep network. The 3D-R2N2 [13] attempts to recover 3D voxels from a single and multiple photographs. [14] recovers a dense set of 3D point cloud using a generation network. It is also possible to incorporate 3D geometry proxies such as volumetric abstraction [45], [46], hierarchical CSG tree [47], part models [48], etc. Results from these techniques are promising but still fall short compared with CSG models. Closest to ours is the work from the Magic Leap group, with a clear interest in virtual and augmented reality, to recognize and reconstruct 3D cuboids in a single photograph [43]. Our approach is able to recover more general shapes, namely generalized cuboid and cylindrical objects.

Sweep-based 3D Modeling. A core technique we employ is 3D sweeping. Sweeping a 2D profile along a specific 3D trajectory is a common practice for generating 3D models in computer-aided design (CAD). Early CAD systems [49] use simple linear sweeps (sweeping a 2D polygon along a linear path) to generate solid models. Shiroma *et al*. [50] develop a generalized sweeping method for CSG modeling. Their technique supports curved sweep axis with varying shapes to produce highly complex objects. [51] conducts volume preserving stretching while avoiding self-intersections. More recent 3-Sweep [8] and its extension, D-Sweep [52], pair sweeping with image snapping. All previous approaches require manual inputs from the user whereas we focus on fully automated shape generation.

3 Overview

The pipeline of our framework is shown in Fig. 2. We take a single photograph containing objects of interests and feed it into our GeoNet to produce instance masks labeled as *cuboid profile*, *cuboid body*, *cylinder profile*, and *cylinder body*. These instance masks are then used for estimating the 3D profile (a circle or a rectangle) and the camera pose, along with a trajectory axis (a planar 3D curve) for the profile to sweep to create the 3D model.

The architecture of our GeoNet is illustrated in Fig. 3. We build upon the instance segmentation network of Mask R-CNN. The output of Mask R-CNN, coupled with contour image and the edge map, is fed into a deformable convolutional network which is derived from [16] and [17]. With the information of contour

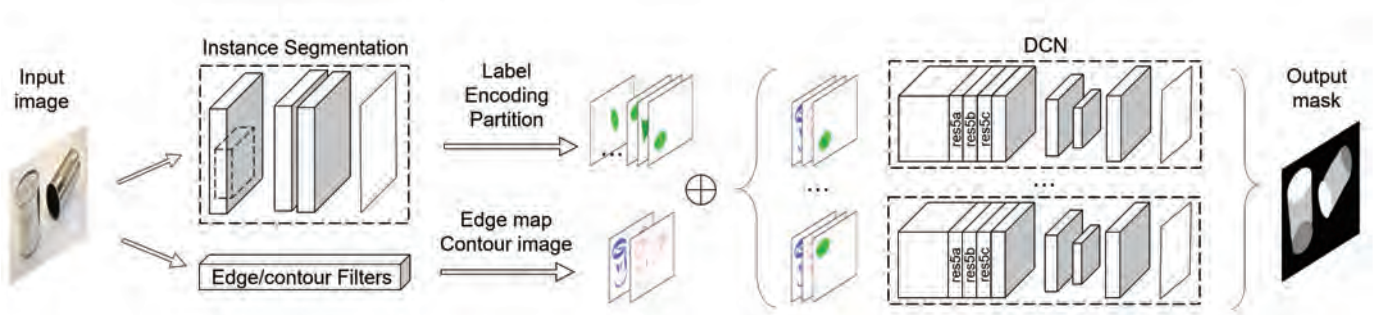


Fig. 3: The structure of our GeoNet is composed by an instance segmentation network (Mask R-CNN) and a deformable convolutional network derived from [16], [17]. The net outputs instance masks labeled as semantic parts (profiles, bodies).

and edge maps, DCN is capable of learning a better and smooth boundary. Details are given in Section 4.

To sweep a primitive part, we first co-relate profile/body masks which could constitute a 3D part. Given correlated profile-body masks, a 3D profile is optimized with camera FoV and a trajectory axis is computed from the body/profile masks. Then, sweeping is performed in 3D to progressively transform and place the estimated 3D profile along the trajectory axis to construct the final model.

4 Instance Segmentation

GeoNet. Our GeoNet takes an image as input and outputs the following four types of instance masks: *cuboid profile*, *cuboid body*, *cylinder profile*, and *cylinder body*. A direct instance segmentation network (Mask R-CNN) could lead to erroneous boundaries and incomplete masks that do not agree with the actual object mask, because the resolution of feature map are lower due to the ROI memory consumption [15]. (Fig. 4). Atrous convolution of Deeplab controls the respective fields under a reasonable range, while deformable convolution causes more effective respective fields which can help the net better handle the transformations of instances regarding scale, aspect ratio, and rotations and in the meanwhile can improve the detail of segmentation results. Thus, we integrate deformable convolution layers proposed in [17] into the network structure of Deeplab [16] and concatenate it with Mask R-CNN for segmentation refinement. We call the sub network concatenated to Mask R-CNN the deformable convolutional network (DCN).

To boost the performance of our GeoNet, instead of directly feeding into the DCN with the results from Mask R-CNN, we use more information from the original image to help GeoNet learn more boundary features. We have tested various case, including using different combination of the original image, the edge map of the original image, and the probability maps from Mask R-CNN, etc., to feed into DCN. Quantitative comparisons are demonstrated in Section 6. At last, we find combining the edge map [8] and contour map [53] of the input image with probability maps given by Mask R-CNN achieves the best performance. We thus combine these with each instance probability map and feed into DCN. Specifically, for each instance probability maps $I_p^n (n = 1, ..., N)$ from Mask R-CNN, we combine it with the edge map I_e and the contour map I_c and convert them into a single image (I_p^n takes the Green channel, I_e and I_c take the Red and Blue channel respectively, see Fig. 3 middle). We assign different green values (40 for cuboid body, 100 for cuboid profile, 150 for cylinder body, and 200 for cylinder profile) weighted with probability map I_p^n for different instance categories to distinguish the instances. The shape of instances in one category have quite similar geometrical characteristics, thus labeling the instance with different green values helps the network to learn a better geometrical feature within this category. We find this simple strategy greatly improves the performance of DCN.

The output of DCN is a refined instance mask $\hat{I_m^k}, k \in \{1, ..., N\}$. After getting through the DCN, we combine all instance masks $\hat{I_m^n}$ $(n = 1, ..., N)$ to form the final mask. To enforce feature learning, the beginning of our DCN is formatted by Res-Net with deformable convolution layers in res-5a, res-5b and res-5c, and connected with 2 convolution layers and 1 deconvolution layer.

Pre-training. Large nets are typically difficult to train. A good initial guess of the parameters usually leads to better convergence. Thus before using the real images, we pre-train the net with synthetic data. We manually construct a dataset containing 10 exemplar cuboids and generalized cylinders collected from ShapeNet [54] (see in Fig. 5). We render these examples from uniformly sampled view angles to generate 1000 images for each example, which gives us 10000 examples for pre-training. We render single instance per image for this task. Since we do not have a large number of instances in our dataset, we decrease the ROI number from 256 to 128 during the training of Mask R-CNN. We also enlarge our dataset with flipped images.

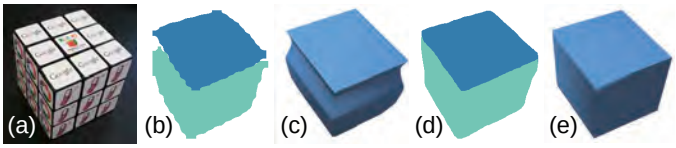


Fig. 4: (a) Input image. Segmentation (b) and modeling (c) results of Mask R-CNN. Our GeoNet is capable of filling the gaps and snapping to the boundary (d), (e).

5 Modeling

Given the output masks from GeoNet, our next task is to create a 3D model that agrees with the target masks. We first separate the masks into independent parts (i.e., primitives) constituted by profiles and body and then construct each part independently.

5.1 Instance labelling

Let us denote the set of instances segmented from the network as unlabelled profile faces $\Omega = \{f_1, f_2, ..., f_n\}$ and labelled bodies

Fig. 5: Representative synthetic models used in our pre-training. The second and third rows are the corresponding contour maps and label masks, respectively.

$\Gamma = \{l_1, l_2, ..., l_m\}$. Our task is to match each unlabelled profile f_i with its corresponding body l_j. This is essentially a labeling problem.

We formulate the following minimization problem:

$$\underset{\{k,l\}}{\operatorname{argmin}} E := \sum E_u(f_i \to l_k) + \lambda \sum E_b(f_i \to l_k, f_j \to l_l), \tag{1}$$

where $E_u(f_i \to l_k) = \alpha E_1(i,k) + (1-\alpha)E_2(i,k)$ is the unary term. E_1 measures the closest Euclidean distance between profile i and body k. We set it to a large constant $C = 1000$ if the distance exceeds a threshold D (3% of the image height in our implementation). E_2 measures the proximity of the face to the body. We define it as $E_2(i,k) = e^{-\phi(i,k)^2/2\sigma^2}$, where $\phi(i,k)$ is the portion of the points on profile i which are inside the oriented bounding box of body k. Both α and σ are set to 0.3.

The binary term is defined as $E_b(f_i \to l_k, f_j \to l_l) = C * \delta(f_i \bigotimes f_j | k, l)$, where $\delta(f_i \bigotimes f_j | k, l)$ is a function which takes value 1 if f_i and f_j overlaps and k is equal to l and takes value 0 otherwise. The binary term is basically set to penalize two overlapped (i.e., occluded) profiles being assigned to the same body. We solve the above optimization by MRF.

For bodies that have no corresponding profiles, such as the handle of a mug whose profile is invisible due to occlusion, we gather them to form a handle set Γ_H and attach them to the closest bodies in Γ. Fig. 2 left gives a brief illustration. We discard false detected handles if their distance is far away from any detected body (3% of the image height in our experiments).

To fit our 3D model, we use perspective projection rather than orthogonal (which was used in [8]) to create 3D models resembling real world objects. Direct global optimization of the primitive and camera parameters could easily render the problem difficult due to the large variable space. We thus decouple the problem into three steps: profile fitting, trajectory axis estimation, and 3D sweeping.

5.2 Profile fitting

As the object profiles in our case are circles and rectangles, this imposes strong priors for our optimization. We assume a fixed camera pose and camera-to-object distance. Below are details for fitting the 3D circle and rectangle respectively. The key is to find a plausible initial value for the optimization.

Circle. Circles in 3D become ellipses in 2D after projection. We use the PCA center c as the initial circle center, with a default depth value 10. The 3D position of the endpoints v_1, v_2 of the PCA major axis are also obtained at depth 10. The initial radius r is then assigned according to the length of the 3D major axis. For the circle orientation, we cast a ray from the camera to one of the endpoints of the minor axis to intersect with the sphere of radius r centered at c. Let s be the intersecting point. The orientation is set as the normal of the plane passing through v_1, v_2, and s.

Given the initial circle C, together with the mask outline, we optimize 5 variables using Levenberg-Marquardt. The 5 variables are $n = (n_x, n_y, n_z)$, r and f which is the field of view (FoV) of the camera. We define the following optimization formulation:

$$\operatorname{argmin} E := E_p + E_f \tag{2}$$

E_p stands for alignment error after projection, it is defined as $E_p := \tau(p|m) + \alpha/r$, where $\tau(p|m)$ denotes the portion of points which are not inside the mask. α is set to 40. E_p ensures the circle is inside the mask boundary while its radius is as large as possible after the projection. E_f stands for the error between profile normal and the starting direction of the trajectory axis (Section 5.3) under different FoVs. We define it as follows: $E_f := \Theta(n_p, n_s) + \eta(n_x, n_y, n_z)$, where $\Theta(n_p, n_s)$ is the acute angle between n_p and n_s, with n_p denoting the normal n projected to 2d and n_s denoting the starting direction of the medial axis mentioned in Section 5.3. $\eta(n)$ is a function that guarantees normal n has a square magnitude of 1.

In a second step, we optimize the circle position c separately using only the first term of the objective function E_p to get an updated c. With the new c, we go back to the optimization of radius, normal and camera FoV. The two steps are iterated until convergence.

Rectangle. Rectangles are optimized in a similar way. We first detect four vertices by fitting a quadrilateral to the profile mask. Then cast four rays from the camera to the four vertices. The 3D vertices v_1, v_2, v_3, v_4 (in clockwise) of the four vertices which lie on the four rays are then optimized as follows:

$$\operatorname{argmin} E := E_c + E_p + E_f \tag{3}$$

where E_c keeps the spatial information of the rectangle through the following constraints: (1) parallel edges have equal length, (2) adjacent edges are perpendicular to each other, (3) four vertices are coplanar. We define E_c as:

$$\begin{aligned} E_c := \textstyle\sum_{i=1}^{4} (\lambda_1(|e_i| - |e_{i+2}|) + \lambda_2 \Theta(e_i, e_{i+1}) \\ + \lambda_3 \Theta(e_i \times e_{i+1}, e_{i+1} \times e_{i+2}))) \end{aligned} \tag{4}$$

where e_i are the vector created by adjacent vertices v_i, v_{i+1}. Θ computes the cosine of the acute angle between two vectors. We add parameters λ_i to normalize each term. E_p and E_f are the same as above with radius replaced by side length. We rectify the 3D vertices to form a strict planar rectangle during iteration.

5.3 Trajectory axis extraction

We then extract a trajectory axis that approximates the main axis of the body. The curve will be a guiding line for the sweeping procedure. We use a morphology operation called thinning [55] to get a single width skeleton of the mask image, as shown in Fig. 6, (b). To better account for the completeness of the skeleton, we use both body and profile masks for thinning. To remove the spurious branches in the skeleton, we use a simple way to prune the branches. We mark the skeleton points as branching point and

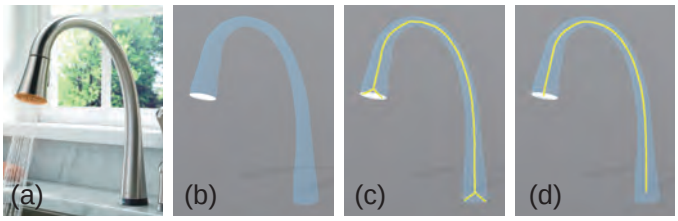


Fig. 6: (a) Original image with profile mask. (b) 3D profile (in white). (c) Trajectory axis after thinning. (d) Trajectory axis after pruning.

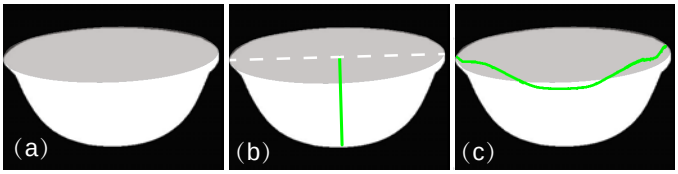


Fig. 7: Trajectory axis extraction. (a) The input mask image. (b) Our result. (c) Medial axis extracted using the method of [57]

end points using hit-or-miss [56]. Branches are identified as paths connecting end points and branching points. We progressively delete shortest branches until we get no branching point.

As our purpose is to reconstruct cylindrical and cuboid object whose trajectory axis is either a straight line or a curve. We perform trajectory axis classification. The goal is to classify whether the trajectory axis is a straight line or not. Simple heuristics such as using line fitting with specific thresholds could lead to erroneous estimations. For a more general solution, we utilize the training data available in our dataset. We employ the LeNet [58] and modify the last FC layer into 2 classes. We use both the body mask and the associated profile masks as input to provide the net with more contextual information. Specifically, we compute their bounding box and scale them to the size of 56×56 as input to the net. We get an accuracy of 96% for this task.

If the trajectory axis is labeled as a straight line, we rectify the axis direction w.r.t. profile axis in cases the thinning process gives erroneous skeleton (e.g., for a cylinder we simply set the axis to be orthogonal (in 2D) to the major axis, see Fig. 7). In case when the trajectory axis is labeled as a curve. We set the starting point to profile center and perform bilateral filtering to get the final curve axis. See Fig. 6 (d) for an example. We find this simple thinning-and-rectifying strategy to perform well in our experiments.

We also investigated previous medial axis extraction method of [57]. Since their method disregards the context information of the profile faces and thus could lead to erroneous estimations (see an example in Fig. 7).

5.4 Sweeping

Given the 3D profile and the trajectory axis, our next task is to sweep a 3D model which approximates the body mask. As in [8], we assume that the trajectory axis lies on a plane which is orthogonal to the profile plane and passes through the profile center. For simplicity, we set the plane orientation to be orthogonal to the camera direction if the object is a generalized cylinder. For a cuboid, we let the plane pass through one of the diagonal lines of the rectangle profile.

We project the 2D body mask and the trajectory axis on that plane and start to place the 3D profile uniformly along the projected trajectory axis. For each part to sweep, we start with the profile with a smaller fitting error if there are two. For each individual profile F_t, t stands for frame index, we cast a 3D ray from its center c_t to intersect with the projected body mask and regard this distance as an initial guess for the profile radius. The final radius r_t of F_t is optimized with

$$\min \sum_{k=1}^{n} M_{\mathbf{Pr}(F_t^k)} + \sum_{e,s=1,2} (\hat{P}_e - P_s) + \frac{\alpha}{r_t} \tag{5}$$

Here F_t is the intermediate sweeping profile. $F_t^k \in \mathbb{R}^3 (k = 1, 2...n)$ represents the sampling points of profile F_t. M is a 2D logical matrix representing the segmentation mask. $Pr(\cdot)$ is a 3D-to-2D projection function which outputs a 2 dimensional vector (x, y) in the camera space. The vector is regarded as the index of M with x and y representing row and column respectively. In Eqn 5, the first term measures how many sample points fall inside the body mask; the second term is the distance between the intersection points $\hat{P}_e$ and its nearest point P_s on the profile boundary as in [8]. $\hat{P}_e$ is computed by casting a 2D ray from the projected center $Pr(c_i)$, then intersect with the edges on the edge map I_e. Here we reuse the edges of I_e mentioned in Section4; the third term aims to ensure that the radius is not too small. α equals 0.025 in our experiment.

The above procedure optimizes the radius for individual sweeping profiles. To ensure the continuity of the geometry, we perform a global optimization on all swept profiles $\mathbf{F}$ after the individual frame optimization. For all $\mathbf{T}$ frames, the aim is to refine all centers $\mathbf{C} \in \mathbb{R}^{T\times 3}$ and orientations $\mathbf{D} \in \mathbb{R}^{T\times 3}$. We solve the following minimization problem:

$$\min \sum_{\mathbf{\Theta}=\mathbf{C},\mathbf{D}} \|\mathbf{\Delta}(\mathbf{\Theta})\|_2 + \mathbf{W}\|\mathbf{\Theta} - \mathbf{\Theta}'\|_2, \tag{6}$$

where $\mathbf{\Delta}$ is the Laplacian smoothing operator and $\|\cdot\|$ is the F-norm. The first term in Eqn 6 measures the smoothness of the geometry, and the second is the deviation of $\mathbf{C}$ and $\mathbf{D}$ to initial values from frames, every weight inside $\mathbf{W}$ is computed by the dot product between the tangential directions of the current and the next frame center on the trajectory axis. Eqn 5 and 6 are iterated to get the final result. In our experiments, both optimizations take around 1-3 iterations to converge.

For generalized cylinder or cuboid which have no associated profiles (e.g., a teapot handle), we estimate an initial position and radius for the profile by analyzing the contact region to the part of the already constructed 3D body. The sweeping process is performed similarly to finally create those parts (see Fig. 2, **??**). Note that before the sweeping process, we globally optimize the camera pose (FoV) with all estimated 3D profiles.

6 Experiments

Dataset. Besides the synthetic data described in Section 4, our real dataset contains multiple human-made primitive-shaped objects widely used in daily life such as mugs, bottles, taps, cages, books, and fridges, etc. There are 11657 real images and 10000 synthetic images (with 11590 generalized cuboids and 15008 generalized cylinders). The real dataset contains about 6000 unannotated images from ImageNet [59], 774 annotated images from Xiao et al. [41], and 4883 images collected from the Internet. The real dataset is further separated into 8183 training images and 3474 testing images. We perform evaluations of all experiments on the testing set of real images.

Method	cub	cuf	cyb	cyf	mAP@0.7	cub	cuf	cyb	cyf	mAP@0.9
FCIS	68.19	61.24	50.33	37.51	54.32	33.04	23.71	10.51	9.09	19.09
GeoNet w. FCIS	68.61	**61.47**	56.75	37.23	56.01	48.64	36.88	**17.14**	10.30	28.24
Mask R-CNN	68.36	61.22	55.93	**40.26**	56.44	35.73	30.13	7.29	10.17	20.83
GeoNet w. Mask R-CNN	**69.49**	61.04	**57.90**	37.84	**56.57**	**50.18**	**37.92**	13.89	**11.37**	**28.34**

TABLE 1: Evaluation of GeoNet with FCIS [22] and Mask R-CNN [15] at overlap thresholds of 0.7 and 0.9 respectively.

Experiment of GeoNet. In order to make full use of the information from original image as well as the outputs of instance segmentation network, We test various combination of gray map I_g, edge map I_e, contour map I_c of the image, mask I_m, probability map I_p from the network. We restrict the combination to form a three channel image, and duplicate channels when the assembled map number is less than 3. For this experiment of combination strategy, we adopt Mask R-CNN as the first stage of our GeoNet. We use mean intersection-over-union (mIoU) defined over image pixels as the evaluation metric, since we are focusing on boundary refinement because the instances are the same during these experiments. The results are shown in Table 3, the combination of I_c, I_p, I_e significantly outperforms the others.

Method	cub	cuf	cyb	cyf	mean
Mask R-CNN	77.56	80.51	68.68	75.74	75.62
GeoNet w. I_m	87.51	85.50	77.89	82.87	83.44
GeoNet w. I_p	89.34	85.84	79.01	83.19	84.34
GeoNet w. I_g, I_p	90.12	85.92	78.28	83.22	84.39
GeoNet w. I_e, I_m	89.67	86.03	79.78	83.82	84.83
GeoNet w. I_e, I_p	90.88	**86.84**	79.51	84.36	85.40
GeoNet w. I_c, I_m, I_e	91.80	86.24	**85.27**	85.37	87.17
GeoNet w. I_c, I_p, I_e	**92.47**	86.81	84.72	**87.02**	**87.76**

TABLE 2: Evaluation of GeoNet on different combinations of gray map I_g, edge map I_e, contour map I_c of image, mask I_m, probability map from Mask R-CNN.

Since our GeoNet is built upon existing instance segmentation networks, to evaluate its effectiveness, we experimented with generally accepted networks of FCIS [22] and Mask R-CNN [15]. We attach the DCN to both FCIS and Mask R-CNN and evaluate the performance of improvements in the segmentation results.

Accuracy is evaluated by mean average precision, mAP [60], at mask-level IOU (intersection-over-union) with overlap threshold set to 0.7 and 0.9 respectively. The results are shown in Table 1. DCN performs better at larger overlap thresholds. At threshold 0.9, DCN improves the performance by 9.15% and 7.51% (mAP), respectively, which shows that DCN is capable of refining the segmentation result on an adequate basis (see also Fig. 4 for a visual comparison). For a plausible comparison, we set the instance count to a fixed number for computing mAP. The chart in Fig. 8 shows the mAP at different overlap thresholds. DCN works better when the base results from FCIS and Mask R-CNN agree with the ground truth. We only visualize the range [0.6, 0.9] since DCN is capable of boosting the performance when the segmentation results are rather accurate w.r.t. the ground truth, while when mAP is lower than 0.6, we find that DCN is much less helpful for refining the boundary.

It is also noteworthy that our method is capable of segmenting and reconstructing objects from raw sketch inputs as shown in the last column of Fig. **??**. This indicates that our DCN network is able to learn cues from the input contour images and edge maps for predicting the final mask.

Comparisons to boundary refinement method. We compare

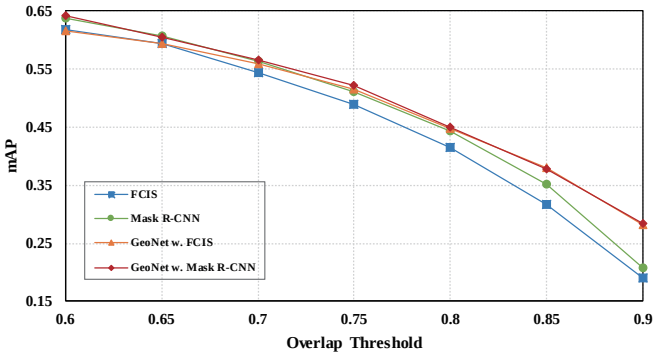


Fig. 8: DCN improves the performance of segmentation results when the base segmentation results are more faithful to the ground truth.

Metric	Method	cub	cuf	cyb	cyf	mean
PP-IOU	Baseline	80.97	76.66	78.75	59.85	74.06
	BNF [23]	**83.40**	**77.86**	79.02	58.46	74.69
	Ours	82.94	77.69	**80.70**	**60.62**	75.49
PI-IOU	Baseline	79.50	78.52	77.70	59.36	73.77
	BNF [23]	80.39	76.67	77.19	47.81	70.52
	Ours	**81.47**	**79.94**	**78.51**	**59.42**	74.84

TABLE 3: Semantic segmentation comparison on our dataset. Note that BNF has a significant drop on cylinder profile because it may fail when the boundaries are not clear, while many cylinder profiles have no clear boundaries due to self occlusion in our case.

GeoNet with Boundary Neural Fields [23] on semantic segmentation task on our test set containing 1614 cuboids and 1840 cylinders. We use the evaluation metrics pixel intersection-over-union averaged per pixels (PP-IOU) and pixel intersection-over-union averaged per image (PI-IOU) same as [23]. We also run the evaluation on the Mask R-CNN output as a baseline for the comparison.

According to this metric, PP-IOU is computed on a per pixel basis. As a result, the images that contain large object regions are given more importance. On the other hand, PI-IOU gives equal weight to each of the images. As shown in Table 3, BNF has lower accuracy on PI-IOU indicates that it is not able to segment small objects accurately. However our method outperforms Mask R-CNN and BNF on average accuracy on both metrics.

Comparisons to cuboid detection and reconstruction methods. We use the SUN primitive dataset [41] to evaluate our method on cuboid reconstruction and compare with the methods of [41] and [43]. For cuboid detection, a bounding box is correct if the Intersection over Union (IOU) overlap is greater than 0.5. For keypoint localization, we use re-projection accuracy (RA) used in a baseline approach Xiao *et al*. [41] as well as the Probability of Correct Keypoint (PCK) and Average Precision of Keypoint (AP-K) metrics used in the state-of-the-art method Dwibedi *et al*. [43]. The latter two are commonly used in the human pose estimation

Fig. 9: Representative results generated using our method. Our method is able to recover objects constituted by multiple semantic parts (e.g., teapots, lamps, water taps, etc.). The first row shows some of the editing results of the model created. The two examples (last column) show that our method can be directly applied to sketch input. We assume symmetry in texture maps, mirror the front texture to back, and finally stitch them together.

Method	AP	RA	APK	PCK
Xiao *et al*. [41]	24.00	38.00	-	-
Dwibedi *et al*. [43]	75.47	-	41.21	38.27
Ours	**79.56**	**49.79**	**47.56**	**45.11**

TABLE 4: Comparison of cuboid bounding box detection and keypoint localization. AP is the average precision for bounding box detection used in Xiao *et al*. [41].

task. We use the re-projection corners of the reconstructed cuboids as keypoints for this task. The comparison results are shown in Table 4. The numbers show that our approach performs better in both tasks.

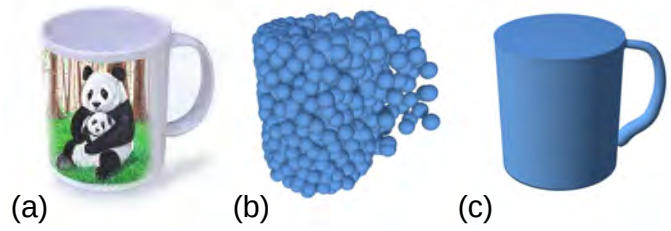


Fig. 10: Comparison with point/voxel-based image reconstruction methods. (a) The input image. (b) The result of point-based framework [14]. (c) Our result.

Comparisons to point/voxel-based and semi-automatic reconstruction methods. We compare our method with two single image reconstruction methods using neural networks, Choy *et al*. [13] and Fan *et al*. [14]. We also compare with Densely Connected 3D Autoencoder (DC3dA for short) in Li *et al*. [61] from the ShapeNet reconstruction challenge. All of them are able to generate a rough representation of the 3D object from a single photograph. The visual comparison examples are shown in Fig. 10 and Fig. 11. We train their network using the 2000 cup and 2000 lamp models collected from ShapeNet [54]. The models are generated with the code provided by the authors with default parameters. It can be seen that our result is cleaner and more accurate. In addition, our models can be directly textured and edited while theirs can not due to the lack of semantic part information.

Additionally, we conduct experiments to compare our approach vs. semi-automatic method 3-sweep [8] on 10 models (5 tables, 5 lamps) using our own implementation. The average reconstruction error (measured as Hausdorff distance) for 3-sweep is 1.263% whereas our is 1.262% (for the method of 3D-R2N2 and DC3dA in [61], the errors are 2.72% and 2.26%, respectively). Fig. 11 shows the qualitative examples.

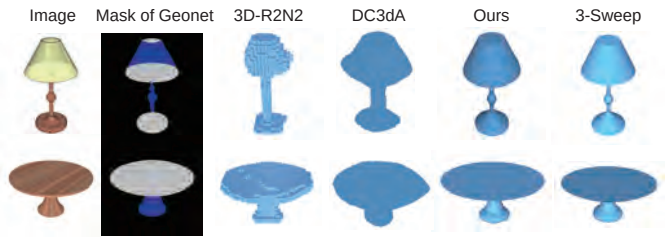


Fig. 11: The comparison with 3-sweep, 3D-R2N2 and the Densely Connected 3D Autoencoder (DC3dA) in Li *et al*. [61].

Timing. The training of the networks is performed on a server with 4 NVIDIA GeForce GTX Titan X GPUs, an Intel i7-6700K CPU, and 64GB RAM. It takes three days to train the Mask R-CNN and one day to train the DCN on our dataset of 8183 images. It takes $1s$ for GeoNet to segment one image and less than 1 second to reconstruct objects from the masks including stages of instance labeling, profile fitting, and 3D sweeping with

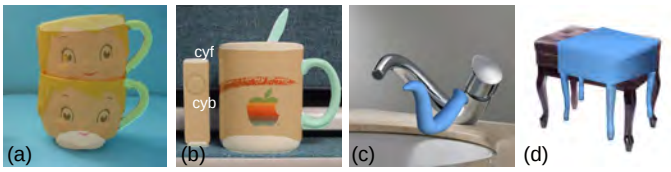


Fig. 12: The failure cases of our approach.

multi-thread acceleration (the individual profile optimization can be performed in parallel).

Limitations. Our method has limitations. As shown in Fig. 12, the network is not able to infer the regions of instances which are cluttered or under occlusion. Priors such as symmetry and physical validity can be enforced to alleviate the problem as in [62] [63]. Next, the network may give wrong class labels when the 2D projection of the shape is vague. As shown in Fig. 12, the remote control is mistaken for a generalized cylinder by the network. For complex objects, our method is currently not able to accurately reconstruct parts which deviate much from the training set or cannot be approximated by GC-GCs such as the parts of the table shown in Fig. 12. In this example, it is also noted that our method may fail to predict correct alignments between the parts. This is because in our experiments, individual parts are constructed in parallel whereas their semantic relations such as coplanar or co-axial may need further rectification utilizing methods of e.g., [64]. In the future, it would be interesting to incorporate such semantics in the network design. Finally, our method cannot handle cases where the axis of the object does not lie on a spatial plane. Thus the object can not have spiral axis such as a spring. To infer such spatially varying curved trajectory requires additional assumptions [65]. We leave this for future work.

7 Conclusion

This paper presents a fully automatic method for extracting 3D editable objects from a single photograph. Our framework uses Mask R-CNN as a basis to build a network which is capable of improving the instance segmentation results. In the subsequent modeling stage, we simultaneously optimize for the camera pose and the 3D object profile and estimate the 3D body shape by a sweeping algorithm.

Our framework is capable of reconstructing primitive objects constituted by generalized cuboids and generalized cylinders. Unlike previous 3D reconstruction methods which reconstruct either 3D point clouds, voxels, or surface meshes, our model recovers high-quality semantic parts and their relations, which naturally enables plausible edits of the image objects. Qualitative and quantitative results have demonstrated the effectiveness of our method. In the future, we plan to explore possibilities of building a more generic and end-to-end framework to reconstruct high-quality primitive 3D shapes from single images or videos.

Acknowledgments

This work was supported in part by the National Key Research & Development Program of China (2016YFB1001403), the National Natural Science Foundation of China No. 61502306, No. 61772462, No. U1736217, No. U1609215, Microsoft Research Asia, the Programs of ScienceandTechnologyCommissionofShanghaiMunicipality (17JC1403800), Shanghai Academic/Technology Research Leader (17XD1402900), and the China Young 1000 Talents Program.

References

[1] S. Zhou, H. Fu, L. Liu, D. Cohen-Or, and X. Han, "Parametric reshaping of human bodies in images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 126:1–126:10, 2010.

[2] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra, "Interactive images: Cuboid proxies for smart image manipulation," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 99:1–99:11, 2012.

[3] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh, "3d object manipulation in a single photograph using stock 3d models," *ACM Trans. Graph.*, vol. 33, no. 4, p. 127, 2014.

[4] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung, "Make it stand: Balancing shapes for 3d fabrication," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 81:1–81:10, 2013.

[5] J. Dumas, J. Hergel, and S. Lefebvre, "Bridging the gap: Automated steady scaffoldings for 3d printing," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 98:1–98:10, 2014.

[6] "Apple arkit," *Apple Inc.*, 2017.

[7] R. Arora, R. H. Kazi, F. Anderson, T. Grossman, K. Singh, and G. Fitzmaurice, "Experimental evaluation of sketching on surfaces in vr," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. ACM, 2017.

[8] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-sweep: Extracting editable objects from a single photo," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 195:1–195:10, 2013.

[9] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 11–20.

[10] N. Jiang, P. Tan, and L.-F. Cheong, "Symmetric architecture modeling with a single image," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 113:1–113:8, 2009.

[11] R. A. Brooks, R. Creiner, and T. O. Binford, "The acronym model-based vision system," in *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, ser. IJCAI'79, vol. 1, 1979, pp. 105–113.

[12] A. Gupta, A. A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *European Conference on Computer Vision*. Springer, 2010, pp. 482–496.

[13] 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," in *European Conference on Computer Vision*. Springer, 2016, pp. 628–644.

[14] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[20] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[21] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *European Conference on Computer Vision*, 2016, pp. 534–549.

[22] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] G. Bertasius, J. Shi, and L. Torresani, "Semantic segmentation with boundary neural fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3602–3610.

[24] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5168–5177.

[25] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*, 2016, pp. 75–91.

[26] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *IEEE International Conference on Computer Vision (ICCV)*, 2005, pp. 654–661.

[27] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems*, 2006, pp. 1161–1168.

[28] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[29] A. Mousavian, H. Pirsiavash, and J. Košecká, "Joint semantic segmentation and depth estimation with deep convolutional networks," in *International Conference on 3D Vision (3DV)*, 2016, pp. 611–619.

[30] I. D. Reid and A. Zisserman, "Goal-directed video metrology," in *Proceedings of the 4th European Conference on Computer Vision*, ser. ECCV '96, 1996, pp. 647–658.

[31] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 123–148, 2000.

[32] M. Wilczkowiak, P. Sturm, and E. Boyer, "Using geometric constraints through parallelepipeds for calibration and 3d modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 194–207, 2005.

[33] A. R. François, G. G. Medioni, and R. Waupotitsch, "Reconstructing mirror symmetric scenes from a single view using 2-view stereo geometry," in *International Conference on Pattern Recognition*, vol. 4. IEEE, 2002, pp. 12–16.

[34] W. Hong, A. Y. Yang, K. Huang, and Y. Ma, "On symmetry and multiple-view geometry: Structure, pose, and calibration from a single image," *International Journal of Computer Vision*, vol. 60, no. 3, pp. 241–265, 2004.

[35] Y. Zhang, W. Xu, Y. Tong, and K. Zhou, "Online structure analysis for real-time indoor scene reconstruction," *ACM Trans. Graph.*, vol. 34, no. 5, pp. 159:1–159:13, 2015.

[36] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Depth perception from a single still image," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, ser. AAAI'08. AAAI Press, 2008, pp. 1571–1576.

[37] Q. Huang, H. Wang, and V. Koltun, "Single-view reconstruction via joint analysis of image and shape collections," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 87:1–87:10, 2015.

[38] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models," in *IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 3762–3769.

[39] K. Rematas, C. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars, "Novel views of objects from a single image," *arXiv preprint arXiv:1602.00328*, 2016.

[40] A. Bansal, B. Russell, and A. Gupta, "Marr revisited: 2d-3d alignment via surface normal prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5965–5974.

[41] J. Xiao, B. Russell, and A. Torralba, "Localizing 3d cuboids in single-view images," in *Advances in Neural Information Processing Systems*, 2012, pp. 746–754.

[42] M. Hejrati and D. Ramanan, "Categorizing cubes: Revisiting pose normalization," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[43] D. Dwibedi, T. Malisiewicz, V. Badrinarayanan, and A. Rabinovich, "Deep cuboid detection: Beyond 2d bounding boxes," *arXiv preprint arXiv:1611.10010*, 2016.

[44] D. Eigen, C. Puhrsch, and R. Fergus, *Depth map prediction from a single image using a multi-scale deep network*, 2014, vol. 3, pp. 2366–2374.

[45] P. Sala and S. Dickinson, "3-d volumetric shape abstraction from a single 2-d image," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–9.

[46] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *Proc. CVPR*, vol. 2, 2017.

[47] X. Chen, J. Tang, and C. Li, "Progressive 3d shape abstraction via hierarchical csg tree," in *Second International Workshop on Pattern Recognition*, vol. 10443, 2017, p. 1044315.

[48] P. Sala and S. Dickinson, "Contour grouping and abstraction using simple part models," in *ECCV*. Springer, 2010, pp. 603–616.

[49] A. A. Requicha and H. B. Voelcker, "Solid modeling: a historical summary and contemporary assessment," *IEEE Computer Graphics and Applications*, no. 2, pp. 9–24, 1982.

[50] Y. Shiroma, Y. Kakazu, and N. Okino, "A generalized sweeping method for sgc modeling," in *Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications*. ACM, 1991, pp. 149–157.

[51] A. Angelidis, M.-P. Cani, G. Wyvill, and S. King, "Swirling-sweepers: Constant-volume modeling," *Graphical Models*, vol. 68, no. 4, pp. 324–332, 2006.

[52] P. Hu, H. Cai, and F. Bu, "D-sweep: Using profile snapping for 3d object extraction from single image," in *International Symposium on Smart Graphics*. Springer, 2014, pp. 39–50.

[53] M.-M. Cheng, "Curve structure extraction for cartoon images," in *Proceedings of the 5th Joint Conference on Harmonious Human Machine Environment*, 2009, pp. 13–25.

[54] 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.

[55] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.

[56] E. R. Dougherty, "An introduction to morphological image processing," *Spie Optical Engineering*, vol. tt9, 1992.

[57] A. S. Montero and J. Lang, "Skeleton pruning by contour approximation and the integer medial axis transform," *Computers & Graphics*, vol. 36, no. 5, pp. 477–487, 2012.

[58] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[60] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.

[61] L. Yi, H. Su, L. Shao, M. Savva, H. Huang, Y. Zhou, B. Graham, M. Engelcke, R. Klokov, V. Lempitsky *et al.*, "Large-scale 3d shape reconstruction and segmentation from shapenet core55," *arXiv preprint arXiv:1710.06104*, 2017.

[62] R. Guo and D. Hoiem, "Support surface prediction in indoor scenes," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ser. ICCV '13, 2013, pp. 2144–2151.

[63] T. Shao, A. Monszpart, Y. Zheng, B. Koo, W. Xu, K. Zhou, and N. J. Mitra, "Imagining the unseen: Stability-based cuboid arrangements for scene understanding," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 209:1–209:11, 2014.

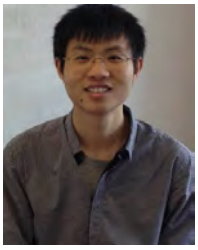
[64] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "Globfit: Consistently fitting primitives by discovering global relations," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 52:1–52:12, 2011.

[65] S.-H. Bae, R. Balakrishnan, and K. Singh, "Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models," in *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '08, 2008, pp. 151–160.

Xin Chen is a Ph.D. student at the School of Information Science and Technology(SIST), ShanghaiTech University. He obtained his B.S. from the School of Science at Qingdao University of Technology . His research interests include human reconstruction, image-based modeling and deep learning.

Yuwei Li is a Ph.D. student at the School of Information Science and Technology(SIST), ShanghaiTech University. She obtained her B.E. from the School of Computer Engineering and Science at Shanghai University. Her research interests include 3D reconstruction, deep learning, and human-computer interaction.

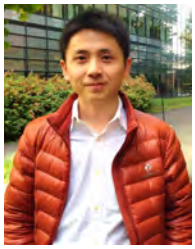
Xi Luo received her B.S. degree in Communication Engineering from School of Mechanical and Electrical and Information Engineering, Shandong University, Weihai, China, in 2016. Now she is a Ph.D. student in computer science, affiliated with ShanghaiTech University, Shanghai, China. Her current research interests are 3D reconstruction, virtual fitting, and human-computer interaction.



Tianjia Shao is currently an Assistant Professor at the School of Computing, University of Leeds. Before that, he was an assistant researcher in the State Key Lab of CAD&CG, Zhejiang University. He received his PhD in Computer Science from Institute for Advanced Study, and his B.S. from the Department of Automation, both in Tsinghua University. His research interests include RGBD image processing, indoor scene modeling, structure analysis and 3D model retrieval.

Jingyi Yu is a Professor in the School of Information Science and Technology, ShanghaiTech University. Before that, he was a full Professor at the Department of Computer and Information Sciences at the University of Delaware. His research covers a range of areas in computer vision including sensors and early vision, computational photography, image-based modeling and rendering, illumination and reflectance, stereo matching, shape-from-X, saliency and segmentation, motion tracking, and medical image analysis. His research has been supported by the NSF, NIH, and DoD. He was a recipient of the NSF CAREER Award in 2009, the Air Force Young Investigator Award in 2010, and the UD College of Engineering Outstanding Junior Faculty Award in 2013.

Kun Zhou is a Cheung Kong Professor in the Computer Science Department of Zhejiang University, and the Director of the State Key Lab of CAD&CG. Prior to joining Zhejiang University in 2008, Dr. Zhou was a Leader Researcher of the Internet Graphics Group at Microsoft Research Asia. He received his B.S. degree and Ph.D. degree in computer science from Zhejiang University in 1997 and 2002, respectively. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality. He currently serves on the editorialadvisory boards of ACM Transactions on Graphics and IEEE Spectrum. He is a Fellow of IEEE.



Youyi Zheng is a Researcher (PI) at the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University. He obtained his PhD from the Department of Computer Science and Engineering at Hong Kong University of Science & Technology, and his M.Sc. and B.Sc. degrees in Mathematics, both from Zhejiang University. His research interests include geometric modeling, imaging, and human-computer interaction.