Localization and Completion for 3D Object Interactions

Xi Zhao, Ruizhen Hu, Haisong Liu, Taku Komura and Xinyu Yang

***Abstract*—Finding where and what objects to put into an existing scene is a common task for scene synthesis and robot/character motion planning. Existing frameworks require development of hand-crafted features suitable for the task, or full volumetric analysis that could be memory intensive and imprecise. In this paper, we propose a data-driven framework to discover a suitable location and then place the appropriate objects in a scene. Our approach is inspired by computer vision techniques for localizing objects in images: using an all directional depth image (ADD-image) that encodes the 360-degree field of view from samples in the scene, our system regresses the images to the positions where the new object can be located. Given several candidate areas around the host object in the scene, our system predicts the partner object whose geometry fits well to the host object. Our approach is highly parallel and memory efficient, and is especially suitable for handling interactions between large and small objects. We show examples where the system can hang bags on hooks, fit chairs in front of desks, put objects into shelves, insert flowers into vases, and put hangers onto laundry rack.**

***Index Terms*—scene synthesis, ADD-image, localization, interaction completion**

I. Introduction

Predicting the type and location of objects that can be added into an existing scene is a process that can be useful for automatic scene generation and character/robot motion planning. One approach to automatically synthesize 3D scenes is to first locate large objects such as bed, sofa and desk into the room and then progressively add smaller objects based on a hierarchy designed or learned from examples. Such scenes can be useful for contents such as 3D computer games and films. Also, given a target location to place an object, motion planning approaches can be applied to plan the motion of the character/robots to bring an object to the target location.

For locating the right place to put objects, classic methods rely on simple object labels and relative displacement vectors [1], [2] between object centroids, that do not generalize to arbitrary objects or complex relationships. A template based method [3] makes use of hand-crafted features to analyze the geometry of the object geometry or the open space around the object and fit objects into a template model. Such an approach suffers from low precision of the hand-crafted features and difficulty in handling objects whose shape differs from the objects used to prepare the template.

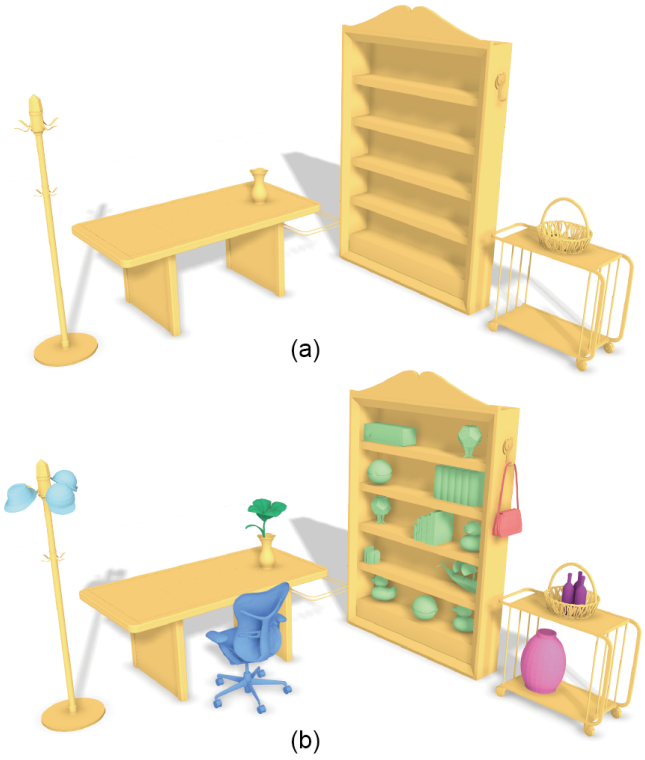


Fig. 1. Given a scene with separated furniture objects as in (a), our method can fill each furniture with objects (as shown in (b)) by localizing and completing the possible interactions that happen to it.

Another stream of research based on deep 3D convolution are recently applied to complete scenes partially captured by 3D scanners such as Kinect, or predicting the partner objects that could be hosted by existing objects in the scene. Completing scenes or predicting locations by 3D volumetric representations can suffer from high-memory consumption due to non-uniform scales of objects. When adding small objects in to scenes composed of large objects, the resolution of the volume needs to be set high such that that the small objects do not suffer from rounding error.

In this paper, we propose a novel data-driven framework to automatically predict the type and location of objects that can be added into a scene (as shown in Fig. 1). Given existing objects that we call *interaction host*, our system identifies the potential areas around them where interactions may occur, and then retrieve and place proper objects that we call *interaction*

Xi Zhao, Haisong Liu and Xinyu Yang are with School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China.

Ruizhen Hu (corresponding author) is with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Email: ruizhen.hu@gmail.com

Taku Komura is with School of informatics, Edinburgh University, Edinburgh, UK.

partners into the scene. Both the interaction host and the partners form a small scene named *interaction unit*. The procedure can be divided into two steps: first discovering the positions to put different types of objects in the given scene, and then deciding the exact configuration of the object to be fit in. Our discovery process is inspired by the recent localization techniques proposed in computer vision. We define an all directional depth image (*ADD-image*) to describe the relationship between the sample and the surrounding geometry. Our system regresses ADD-images randomly sampled in the scene to the position and size of the region of interest (ROI) where the new object can be located. As this is an image-based approach, it is much more memory efficient than full 3D volumetric convolution approaches that requires full voxelization of the host object or the scene. Given several candidate areas around the host object in the scene, our system further predicts the geometry of the partner object that can fit well to the host object. Although we adopt a 3D convolutional framework for this process, it does not suffer from the rounding error as the size of the domain of the 3D convolutional volume is scaled to the ROI where the interaction happens. We finally, find an object model that partially fits to the volume predicted by the 3D convolutional framework.

Our approach is highly parallel, memory efficient, and suitable for handling indoor scenes where large and small objects co-exist. By learning from a set of existing scenes, our system can deal with novel objects with larger geometry variety. The localization allows existing popular 3D convolution based methods to handle complex interactions between objects of different sizes. We show examples where the system can hang bags on hooks, fit chairs in desks, put objects in to shelves and insert flowers into vases. We also evaluate the proposed method both qualitatively and quantitatively.

II. Related work

A. Interaction Representation

Interactions between scene elements are key for analyzing the context of 3D scenes. To be able to model the complex relation/interaction between objects, several different interaction representation have been proposed in previous works. Relative vectors used in Fisher et al. [4] can encode simple spatial relations, while Interaction Bisector Surface (IBS) introduced in Zhao et al. [5], which captures the spatial boundary between two objects, can provide more detailed and informative interaction representation with both geometric and topological features extracted on the IBS. Hu et al. [6] further combined IBS with Interaction Region (IR), which is used to describe the geometry of the surface region on the object corresponding to the interaction, to encode more geometric features on the objects. To guide the placement of new objects around the given object, Zhao et al. [3] define a new feature called Space Coverage Feature (SCF) to encode the relation between an openspace point and the given object. Pirk et al. [7] build a spatial and temporal representation of interactions named interaction landscapes. They compute the flow of particles of one interaction part with respect to the another part to describe the functionality of the latter. Such a representation is useful for describing how objects dynamically interact with one another, especially when one of the object has high flexibility such that its movements can be well described by particles. The target of our research is in synthesizing static complex interactions where the geometry plays an important role. It could potentially be combined with Pirk et al. [7] when animating dynamically changing complex relations.

In this paper, we use an all directional depth image instead of SCF feature to represent spatial relationships, and use networks to localize interaction type and location, which provide more accuracy and efficiency for the synthesis than the template based method in [3].

B. 3D Scene Synthesis

Researchers have been making efforts to improve the efficiency and generality of 3D scene synthesis methods, and most of the works are focused on indoor scenes consisting of furniture objects. Yu et al. [8] propose a system to automatically synthesize indoor scenes by learning the hierarchical and spatial relationships for various furniture objects from the given examples. The final layout is optimized by simulated annealing using a Metropolis-Hastings state search step. Fisher et al. [1] learn a probabilistic model based on the arrangement and occurrences of the objects consisting the scene. In their work, only simple relations, e.g., “support”, “on side” are considered, where a framework to handle complex relations is needed for applying their method to arbitrary scenes.

There is another stream of synthesis methods that consider human activities. For example, Qi et al. [9] model the objects, affordances and activities by a probabilistic grammar model named spatial And-Or graph(S-AOG), and new scenes can be synthesized by sampling the S-AOG. Fu et al. [10] build relations between objects not only by close proximity but also by the human activity. With scene relation graphs encoding such relations, a new layout mask is to be generated based on the user input, which can guide the synthesis of 3D scene. Most of these works focus on the modeling of the co-occurrence between objects and predicting their high level layout, and cannot handle cases such as objects in a shelf and flowers in a vase.

To capture more complex relations between objects for scene synthesis, Majerowicz et al. [11] learn object arrangments from images. They show an example where the system learns how to fill in a shelf given an example image of a shelf. Zhao et al. [3] define a scene template, which can also handle complex interactions, for a given scene exemplar and use it to guide the creation of scene variations with similar complex object-object relations. In this paper, instead of synthesizing new scenes based only on one single example, we use advanced learning techniques combined with geometric inference to learn from a set of scenes, so that a wider variation of scenes can be generated faster and more robustly.

In the most latest work, Hu et al. [12] generates the entire scene context at once to reflect the functionality of the given object and then subdivide the generated scene into regions with different interactions. While there is no guarantee for the completeness and accuracy of each individual interaction in [12],

our work focuses on more detailed interaction unit completion which could also be incorporated in [12] for producing better results. Our method is also closely related to Wang et al. [13], which includes steps of predicting the location, category and orientation of a new object. Instead of building a probability distribution of 2D object layout, our method works in 3D and predicts the interaction category and 3D coordinate of the object center. For the detailed configuration of the object, we predict the 3D guidance for placing existing models, rather than trying different configurations and picking the best one.

C. Network for Detection and Shape Completion

There is a rich literature on computer vision techniques that detect objects in images with rectangular boxes. Here we only list some of the most related works. Beside the famous R-CNN methods and its variants [14]–[16] which are based on region proposal, the end to end network such as Yolo [17] and SSD [18] make impressive improvement in terms of processing speed and prediction accuracy. Our work was inspired by [17] and [19], as predicting the 3D region that is suitable for a possible interaction is inherently a detection problem. We also use a regression network to predict the interaction type and 3D interaction region based on the spatial relationship representation.

3D Shape completion, which fills in the missing or occluded parts of a 3D shape, is attracting more attention recently. Networks with volumetric convolutions have be successfully used for 3D shape completion and synthesis. Wu et al. [20] proposed the first volumetric convolution network which is a DBN and complete the depth data by up-down sampling. Song et al. [21] propose an end-to-end network to complete the depth data and do the voxel-wise semantic labeling at the same time. Han et al. [22] uses a global structure network to guide the completion of local geometry. The completion is done progressively done from the boundary of the missing region. Dai et al. [23] propose a 3D encoder prediction network to predict and fill in missing data of a 3D model. We use the similar architecture to predict the interaction partner. Our approach is similar to their approach in sense it predicts the geometry of the partner object given the input geometry of the host object.

III. Overview

Given a 3D object as the interaction host (Fig. 2(a)), our goal is to construct a 3D scene by first identifying the region of interest (ROI) where interaction partners can locate, and then fitting partner objects into the scene (Fig. 2(d)). Our method consists of the following three steps.

Step 1: Localizing Valid Interaction Areas (see Section IV). The system first sample points in the open space around the host object, and computes an omni-directional depth image, referred to as All Directional Depth image or ADD-image, at each point. A regressor that we call the localization network, that maps the ADD-image to the interaction information, including the type, confidence, location and size of ROI that contains the volume where the interaction happens, is trained. The type here means the pairwise interaction type such as "desk-chair" or "hook-bag". During run-time, among all the ROIs obtained through regression, we choose the non-overlapping ROIs with high confidence as the candidates (shown as the transparent box in Fig. 2(b).)

Step 2: Predicting the Interaction Partner (see Section V). Based on the interaction type and ROI information predicted in the first step, the system predicts the partial geometry of the interaction partner. To achieve this goal, for each candidate ROI, the system predicts the rough geometry of the interaction partner by a 3D encoder-predictor network, which maps the signed distance function of the interaction host to that of the interaction partner within the ROI box. In Fig. 2(c), the voxels with signed distance function value close to 0 are shown.

Step 3: Forming the Interaction Unit (see Section VI). After predicting the geometry of the interaction partner where the interaction happens, the system retrieves 3D models from a given dataset by matching the geometry and the interaction labels. The retrieved object is then placed around the interaction host based on the predicted geometry to form an interaction unit. Finally, a post-processing step is done to avoid penetration and floating artifacts.

IV. Localizing Valid Interaction Areas

To avoid blindly matching candidate 3D objects to the given interaction host, our strategy is to identify all the possible interaction types and their corresponding locations around the interaction host, and then extract the area only with high confidence. We define the *interaction ROI* as a cube where the interaction may happen.

The goal of this step is to predict one or more ROIs with high confidence around the given interaction host. To accomplish this goal, a regressor that we call the localization network (see Section IV-C), that maps the ADD-images (see Section IV-A) at every point in the open space around the interaction host and the information about the nearby ROI, which includes the type of interaction, confidence, the location and size of interaction ROIs (see Section IV-B), is produced.

A. All Directional Depth Image

ADD-image of a point is an all directional depth image that encodes the 360-degree field of view of this point. It describes the geometry that surround this point. The process of computing ADD-image is shown in Fig. 3. A global coordinate is defined such that the y-axis is in the upright direction, and x-z plane is the ground plane of the scene. Given a point P in the openspace around the interaction host, we first define a sphere centered at P, where the north pole is in the direction of the global y-axis. We then uniformly sample n directions along the latitude and longitude of the sphere, and cast rays from P to all these directions. The depth d for rays, which is the distances from P to the first intersection point along each direction, is then computed. We normalize the depth d by a threshold to ignore the intersection too far away and scale the values to the range between 0 and 1. We set the threshold to 30 (the size of the desk in Fig. 2 is $34 \times 19 \times 25$), and the sample number n to 48 in our experiment.

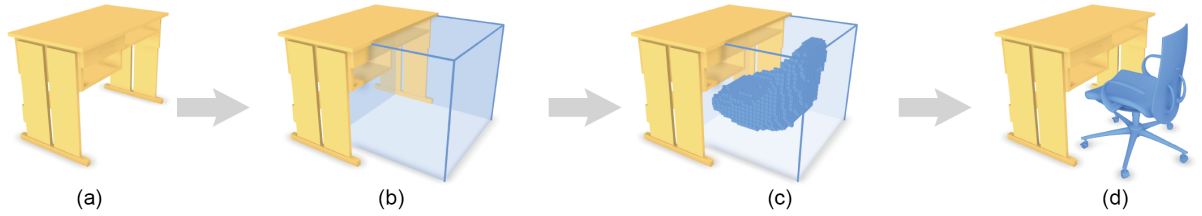


Fig. 2. Overview of our method. Given a novel host object (a), we first localize the possible interaction that may happens to the host by predicting the type, location and size of the ROI (show with the transparent boxes in (b)). Then within the ROI we predict the rough geometry of the interaction partner (shown with voxels in (c)) and then match objects to the predicted geometry to build the final scene (shown as in (d)).

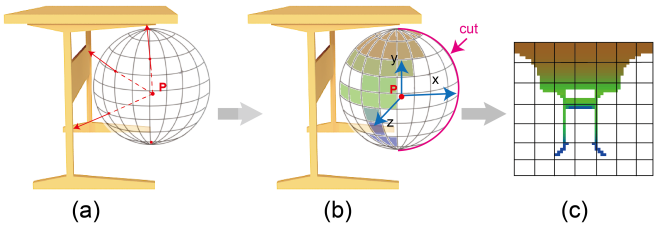


Fig. 3. ADD-image computation steps. Given a openspace point P, we compute the depth values of rays cast towards all directions uniformly sampled on a sphere (shown as (a)). We then build the local coordinate shown in blue in (b) and cut the sphere along the longitude that passes the x-y plane to get the final ADD-image (c).

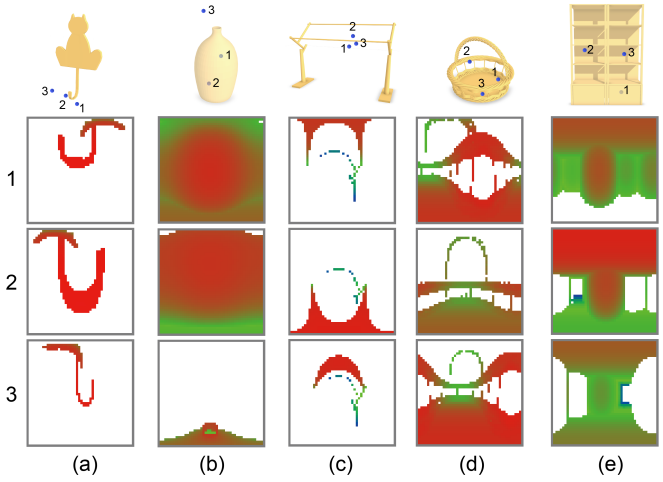


Fig. 4. We show three example ADD-images for different models (a) a hook, (b) a vase, (c) a laundry rack, (d) a basket and (e) a bookshelf. The id (1-3) shows the correspondence between the points around the model and the ADD-images.

To be able to perform convolutions on the depth information, we flatten the sphere to a 2D image, by cutting the sphere along the longitude that passes the x-y plane of a local coordinate of P (Fig. 3 (b)). To define the local coordinate, we first project all rays to the x-z surface of the global coordinate, and use the average of these projected rays as the x axis of the local coordinate, and the global y axis is used as the y axis of the local coordinate. Then local z axis can be easily determined by the local x and y axis. Examples of ADD-images computed around different host objects are shown in Fig. 4.

B. Preparing Training Data

To learn a mapping between the ADD-images and the ROI, we prepare a dataset of example interaction units where two objects are adjacent to one another, forming a minimal unit of a scene. A groundtruth ROI is computed for each interaction unit, and then training data is prepared, each of which is composed of the input ADD-image and the output data including the center location and size of the groundtruth ROI, the confidence value and the id of interaction type. The whole list of interaction types is shown in Fig. 8.

For each interaction unit, we extract a groundtruth interaction ROI to guide the network training. The ROI is defined at the area of the interaction host where the two objects are in close proximity. Taking the table-chair interaction unit in Fig. 5 as an example, when considering the table as the interaction host, the groundtruth ROI is shown as the red box. To extract such a ROI, we first compute the Interaction Bisector Surface (IBS) [5] between the table and the chair (shown as the grey line in Fig. 5), and then find the points sampled on chair (the interaction partner) that defines the IBS. The readers are referred to Zhao et al. [5] for the details of the sampling process. The groundtruth ROI is the minimum cube that contain 85% of these points.

For each training sample, we provide the confidence value that ranges between 0 and 1. Given a point p, where the sample ADD-image is produced, we compute the sample's confidence value as follows. First, we examine if p is on the same side of the IBS where the center of the ROI exists. If it is on the same side, this is a positive sample. Similar to YOLO [17], the confidence value is defined as the IOU of the ground truth ROI and the ROI computed by the localization network at the time of training. If p is in the opposite side of the IBS, this is a negative sample, and thus the confidence value is 0.

For each training sample, we sample same number of positive and negative examples for training: we randomly sample points until both the positive and negative samples reach the desired number. In our experiments, we collect 1000 positive and negative examples. We apply a constraint that a positive example is adopted only when the IOU between the ROI centered at p and the groundtruth ROI is larger than 0.2. This is to collect more positive samples closer to the ground truth, which helps to produce a better mapping. Using the table-chair scene (Fig. 5) as an example, it can be observed that the positive examples exist in the volume where the chair can be tucked in (shown as the light blue background in Fig. 5).

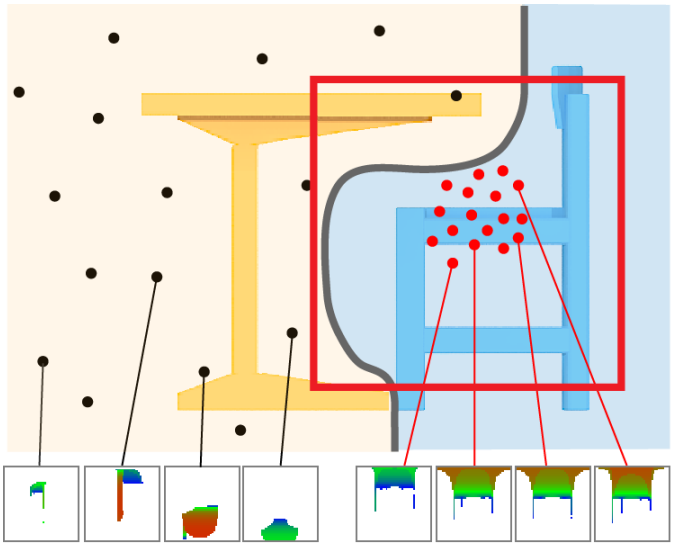


Fig. 5. This images shows an example of groundtruth ROI (the red box), positive samples(shown as the red dots) which correspond to the groundtruth ROI, and negative samples (shown as black dots) which correspond to no ROI. We also show ADD-images for some positive and negative samples.

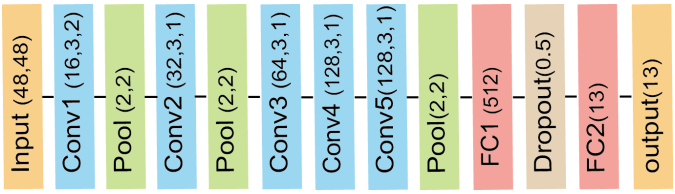


Fig. 6. The localization network we used to predict ROI information from ADD-images.

C. Predicting the ROI

We apply a CNN to regress the ADD-image to the ROI and the associated attributes. Our localization network has five convolution layers followed by two fully connected layers. The first two and the last convolution layers are followed by maxpooling layers. The full description of our network structure is shown in Fig. 6. The network is essentially a simpler version of the model proposed in [17]. The changes aimed to make the network to fit to the low resolution input ADD-image, which is 48×48, so as to reduce the number of parameters and the make the network faster to train.

The input of the network is the ADD-image of any point p in the open space of the given interaction host, and the the output of the network is a vector: $[t, c, x, y, z, w]$, where $t = [t_1, \cdots, t_n]$ indicates the probability of the ADD-image in interaction type t_i, c is the confidence, (x, y, z) is the position of ROI center in the local coordinate of p (defined in Section IV) and w is the size of the ROI.

For any training point p, t is a one-hot vector with only the entry corresponding to its interaction type being 1. Confidence value c is set to be the IOU between the ground truth ROI and the predicted ROI during training for all positive training points and 0 for all negative training points. Both (x, y, z) and w correspond to the groundtruth ROI of the interaction host and are normalized by a threshold to map to range $(0, 1]$.

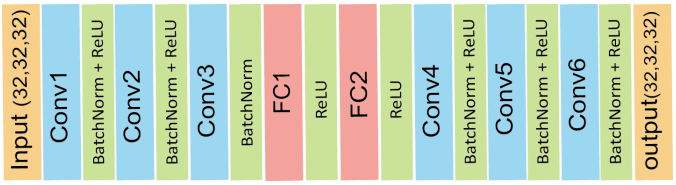


Fig. 7. The 3D encoder-predictor network [23] we used to estimate the geometry of possible interaction partner within the ROI.

We use a loss function similar to the one used in [17]:

$$\begin{aligned} &\mathbf{I}^{\text{roi}}\mathbf{w}_{\text{coord}}[(x-\hat{x})^2 + (y-\hat{y})^2 + (z-\hat{z})^2] \\ &+ \mathbf{I}^{\text{roi}}\mathbf{w}_{\text{size}}(w-\hat{w})^2 \\ &+ \mathbf{I}^{\text{roi}}\mathbf{w}_{\text{confidence}}(c-\hat{c})^2 \\ &+ \mathbf{I}^{\text{noroi}}\mathbf{w}_{\text{confidence}}(c-\hat{c})^2 \\ &+ \mathbf{I}^{\text{roi}}\mathbf{w}_{\text{class}}\sum_{i=0}^{n}(t_i-\hat{t}_i)^2 \end{aligned} \tag{1}$$

where $\mathbf{I}$ is indicator function. $\mathbf{I}^{\text{roi}}$ denotes the ADD-image corresponding to positive examples while $\mathbf{I}^{\text{noroi}}$ corresponds to negative examples.

During runtime, we randomly sample points around the given testing interaction host object and then predict one ROI for each point using the network.We then use Non-Maximum Suppression to find the best set of ROIs for the given object.

For each given host object, it is possible to predict ROIs with different interaction types. We estimate the main interaction type for a host as follows: among the top N ROIs with highest confidence, we count the number of ROIs predicted as each type, then the type which has a dominant number is considered as the "main" type.

V. Predicting the Interaction Partner

In this section, the goal is to predict the geometry of the interaction partner within the ROI predicted from Section IV. To accomplish this goal, we use a 3D encoder-predictor network (Fig. 7) to predict the Signed Distance Field (SDF) of the interaction partner and then collect the voxels whose SDF values are close to zero to represent the rough geometry of the interaction partner.

We use the network proposed in [23], which has three encoder layers, two fully connected layers and then three predictor layers. The detailed structure is shown in Fig. 7. We use no skip connections because we found the skip-connections have no obvious influence on the results. $L1$ loss is used for the training. The input is the SDF of the interaction host within the ROI and the output is the SDF of the interaction partner within the ROI. Both the input and output are $32 \times 32 \times 32$ voxel grids.

We now describe about preparing the training data for the 3D encoder-predictor network. Since the output of the trained localization network described in Section IV would be the input of this 3D encoder-predictor network, we first examine the distribution of the ROIs computed by the localization network in the previous step. We sample ADD-images around the groundtruth ROI, and predict the corresponding ROI center

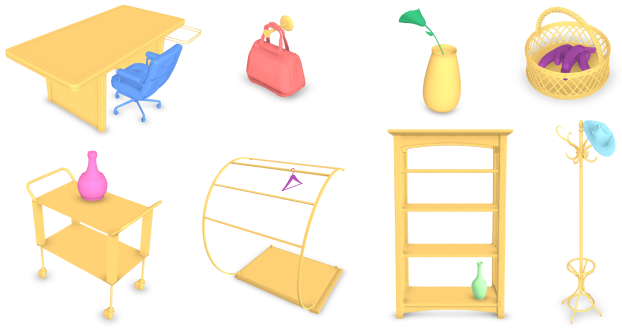


Fig. 8. There are 8 types of interactions in our dataset: desk-chair, hook-bag, vase-flower, basket-object, hangcart-object, laundry rack-hanger, bookshelfobject and stand-hat. Here we show an example of each type of interactions.

location (x, y, z) and size w using the localization network. The distributions of the center location and the size are computed and Gaussians are fit to each of them. We then sample from these distributions and produce cubes which is used to produce a training data of the 3D encoder-predictor network. We produce $32\times32\times32$ SDFs of the interaction host and the partner within these cubes. The SDFs of the interaction host and the interaction partner are used as the training data for the 3D encoder-predictor network.

VI. Synthesizing the Interaction Units

Once we have the rough geometry of the interaction partner around the the area where the interaction is happening (denoted as the predicted geometry), we match the existing object models to it to produce the final scene where an object is fit to the interaction host. Some predicted geometry is just a partial geometry of the interaction partner. For example, the predicted geometry of a bag that hangs on a hook normally just contains the handle of the bag (see Fig. 14 (b)). In order to fit the 3D model to the predicted rough geometry, which could sometimes be only the partial geometry, we use the geometric hashing [24]. Final adjustment is done for solving floating issues. If the matched objects collide with each other, we randomly remove some interaction partner objects until no collision exists.

VII. Experiments and Evaluation

A. Dataset

We use a dataset consisting of 8 different types of interactions, including desk-chair, hook-bag, vase-flower, basket-object, handcart-object, laundry rack-hanger, bookshelf-object, and stand-hat. The 3D model for building the dataset are from the database used in [12].

Each type of interaction contains around 40 different manually made scenes. As shown in Fig. 8, all the interaction scenes in our dataset contain two objects. The whole dataset can be found in the supplementary material.

In the rest of this section, we first evaluate the results of the localization network described in Section IV. We next evaluate the results of 3D encoder-predictor network described in Section V, as well as the final interaction in Section VI.

B. Evaluation of the Localization Network

In this section, we evaluate the performance of the localization network by comparing the accuracy with a template-based approach [3], where a objects are fit into template IBSs to synthesize scenes. To use the template method to predict the interaction type and ROI information, we match the host geometry with multiple templates, and find the type of the ROI with the highest similarity score. To apply the template method based on the same existing data, we match each novel host object with all the templates corresponding to the training data of our method. We do a 5-folds cross validation, and thus the number of templates used for each novel host object is around 250 $(40 \times (4/5) \times 8)$ in our experiments.

ROI Results. To predict the ROI, we randomly sample 5000 points around the host object and select the final ROIs by Non-Maximum Suppression with the IOU threshold 0.1.

We show the ROI detection results in Fig. 9. Different colors are used for each interaction type. Within each scene, the thicker boxes correspond to those with higher confidence values. Our method successfully predicts the interaction type, ROI size and location for most cases. Although the training data contains only pairwise interactions, our method can find multiple interactions within one single host. For example, in Fig. 9, multiple ROIs are found for long desk, double hooks, larger handcarts, bookshelves, stands and laundry racks.

For some hosts, our algorithm predicts more than one type of ROIs. For example, in Fig. 10, the desk, shelf and stand are considered to be able to support objects as a cart (shown as pink boxes in (a), (c) and (e)). Also both push handle of the cart, and parts of the stand are predicted as a bar to put hangers on (shown as purple boxes in (b) and (e)). We also show that the bottom of the desk can be considered as a shelf unit (green boxes in (a)), the corners of the shelf are predicted to be able to support hats (the two small blue boxes in (c)), and the bucket is considered as a vase to put flowers in (green box in (d)). This makes sense because objects may have similar local parts. For example the space between the lower and upper surfaces of the desk in (a) can be used to keep objects. More results for interaction localization can be found in the supplementary material.

To demonstrate that our method is not simply memorizing the dataset, we show the ROIs predicted for a list of gradually changing shapes in Fig. 11. Note that all of these models are different from those in the training set. We can see that the predicted ROIs change gradually to match the vase models. The ADD-images of these vases, which capture partial shape information from the viewpoint of open space points, also change gradually during the shape deformation. Example ADD-images for these vases can be found in the supplementary material.

ROI Type Accuracy. We compute the confusion matrix of the interaction type results for the points that labeled as positive examples. Results shown in Fig. 12. The average classification accuracy of our method is 0.94, while the template based method is only around 0.46.

The main mistake made by our method is that 20% of the basket-object relationship was predicted as the cart-object relationship. This mainly happens to the carts with a concave

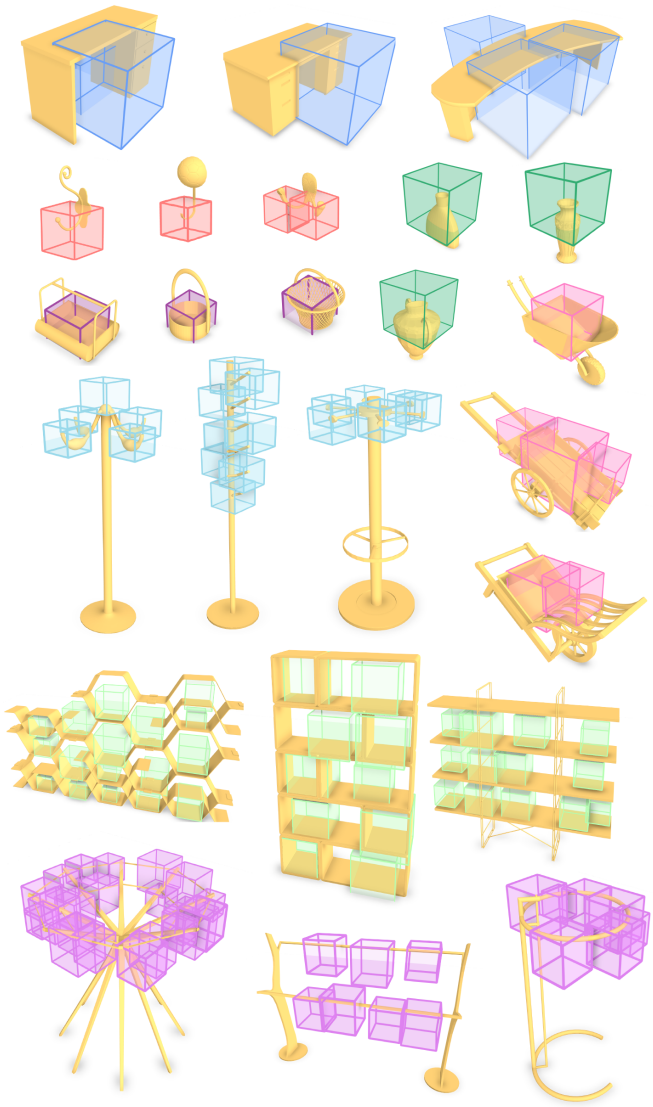


Fig. 9. Predicted ROIs for different types of host objects. Different colors represent different types of interactions.

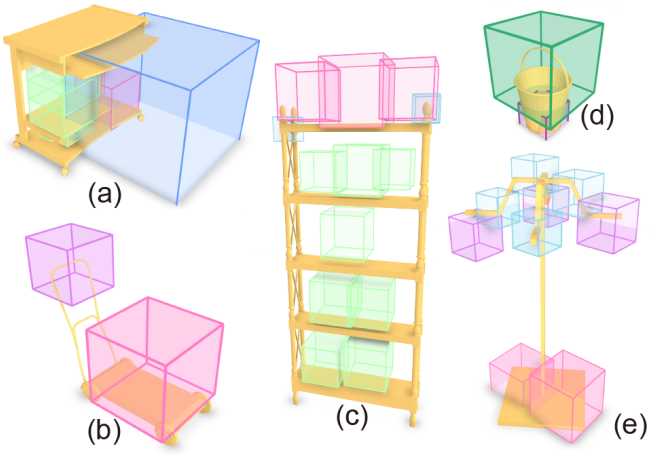


Fig. 10. Examples of multiple types of interactions found for one host object.

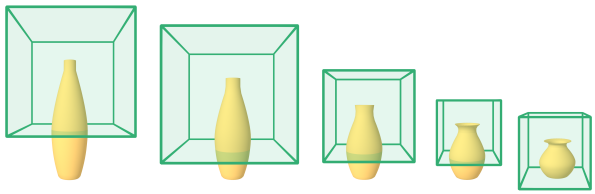


Fig. 11. The predicted ROIs for gradually changing vases.

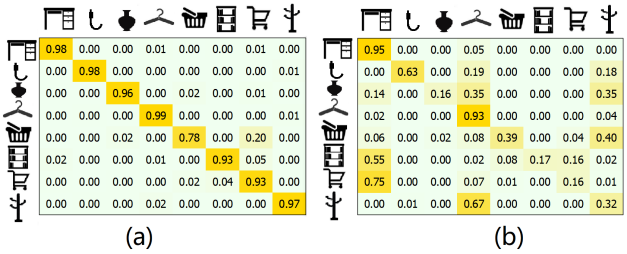


Fig. 12. Confusion matrix for the predicted interaction type by (a) our method and (b) template method [3]. The icons respresent the types of interactions, which are desk-chair, hook-bag, vase-flower, laundary rack-hanger, basket-object, bookshelf-objects, handcart-object, and stand-hats from left to right (top to down).

part which is quite similar to a basket. The other small portion of wrong classification also caused by the local similarity between different type of host objects.

The template method has a much lower classification accuracy compared to our method. This is mainly because the template method compares the histograms of the points classified by the relationship features between the template and the candidate window. Although the histogram can describe how good the template fits to the candidate window, it loses the spatial distribution of the feature points. As a result, the approach cannot evaluate the similarity of the local geometry well. For example, 67% of the stands and 19% of the hooks were predicted as laundry racks, and 55% of the shelves were predicted as desks.

ROI Accuracy. To evaluate the localization accuracy of ROI, we compute the IOU between the predicted ROI and the groundtruth ROI. We draw the IOU vs. recall curve as shown in Fig. 13. For each interaction host, we first predict ROIs for all candidate locations, and then select the ROI with the highest confidence as the best ROI. For the template method, we use the ROI with the highest similarity to the template as the best ROI. The IOU for each input interaction host is then computed between the best ROI and the groundtruth ROI. Note that if the ROI with the highest confidence has the wrong type, then IOU is set to 0.

We show the IOU vs. Recall curves for each types of the interaction in Fig. 13. The solid lines are the results by our method while the dashed lines the template method. The curves by our method are obviously higher. It means our method predict ROIs with higher accuracy. From the curves we can see that our method got more than 50% of the results has IOU larger than 0.4, while the template method only got less than 10% recall with the same IOU threshold.

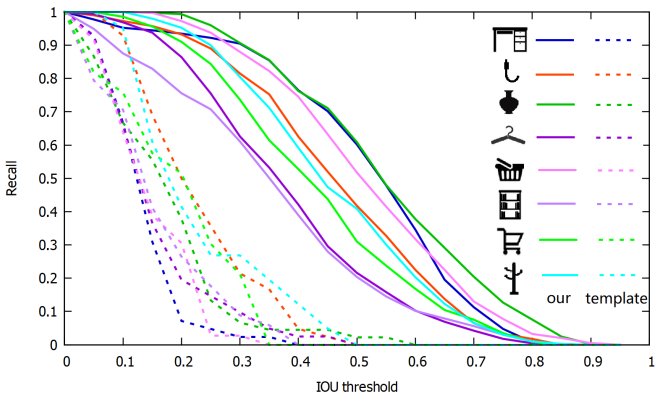


Fig. 13. Evaluation of the ROI precision. The solid lines shows the IOU-recall curves of each interaction type by our method and the dashed lines shows the curves by the template method in [3].

Timing. The template method suffers from high computation costs when synthesizing new scenes based on a set of existing data, because it needs to match all templates with the novel host objects to recognize the interaction type. In our experiment, the template method takes around 1 second for predicting each ROI information within each window when the comparisons with the templates are done parallely on a i7-2600 CPU. Although our method takes around 6 hours to train the model with around 2,000,000 training samples, it only take around 0.005s to predict each ROI.

Summary. Our localization approach can accurately compute the ROI and also classify the interaction type well. It outperforms the template method both in terms of accuracy and performance.

C. Evaluation of the 3D Encoder-Predictor Network and the Final Results

We first show the results of the 3D geometry predicted by the 3D Encoder-Predictor Network as well as the final results by fitting 3D models into the computed SDF. We then show the results of a user study and discuss about the computational time.

Predicted SDF and the Final Scenes. We show the predicted SDF by our method in Fig. 14 and Fig. 15.

In Fig. 14, we show examples of the predicted SDF for each interaction type (the first column), and three scenes synthesized by matching different objects to the SDF. We can see that the predicted SDF is a representative shape of each type of interaction. For example, the SDF predicted around the desk is a shape that is composed of the chair surface and chair back, which is a general geometry that can be applied to fit different types of chairs in front of the desk. The SDF predicted for the hook has a handle shape, which can be applied to fit different bags onto the hook. Other examples also show that the SDF well represents the geometry of the interaction partner, and can match to different objects to produce a wide variation of results.

Fig. 15 shows that the shape of the predicted SDF can adapt to the geometry of the interaction hosts. This is the main advantage of our method comparing to the template method. The template method only matches a rigid template to the novel geometry and it can be difficult to discover a template that can match all types of novel objects. For example, if the template is computed from a vase with wide bottleneck, then this template is quite likely to be oversize for a very narrow vase. Similar situations can also happen with bookshelves, hooks, etc. when the template is too large for a novel object. Even when the template can fit to the novel object well, the synthesized results may lack varietion because the template size is fixed. This leads to the interaction partners in similar sizes. To increase the variation of results, more templates with different size are needed, but this will increase the computation time for matching.

On the contrary, our system can learn from a number of examples, and interpolate between them to adapt to a wide variation of interaction hosts, and produce the appropriate SDF to represent the geometry of the partner that fits well into the host. For example, as shown in Fig. 15, the SDF is narrow for a narrow vase, thick for a vase with a wider bottleneck (a) and the hanger geometry can be predicted for long, short or curved bars (b). We also show similar cases for baskets in (c) and shelve units in (d).

More final results for each type of interaction in Fig. 16 and the supplementary material.

User Study. To evaluate our final results, we conduct an online survey in terms of the plausibility of the scenes produced by our method. For each of the 8 scene types, we randomly select 3 scenes from our final results and 2 scenes that are manually made. The Images of the synthesized scenes are mixed with the manually made scenes for each type in random order, and presented to 50 participants that has no graphics or design related background. Participants were shown 40 images in total and asked about the plausibility of the scene on a 5 point Likert scale (1 = Completely random, implausible scene, 3 = Somewhat plausible scene, 5 = Very plausible scene).

Fig. 17 shows the distributions of the ratings for each type of interactions. The manually created scenes are rated higher for each type except the hook-bag interaction. An independent-samples t-test [25] is conducted to check the difference between the ratings that are given to the manually made scenes and our results. The p-values for each type of interactions are listed in Table I. For significance level of 0.01, no significant statistical difference is found in the ratings except for basket-objects and shelf-objects scenes. This means except these two types, there is a more than 1% chance that there is no real difference between the two sets of ratings.

For the basket-objects and shelf-objects scenes that have lower p-values, we show two example scenes and their corresponding scores in Fig. 18. In Fig. 18(a), the left scene receives a lower average score than the right scene. According to the interview after the online survey, some users feel the left scene is not so good because the apple is not in the middle of the basket. Regarding the shelf-object scenes in Fig. 18(b), the left scene receives a lower score because some users feel it is unnatural that the same bottle appears twice in the shelf and the trophy is not facing front. The objects like the trophy which is symmetric or partially symmetric may have a conventional

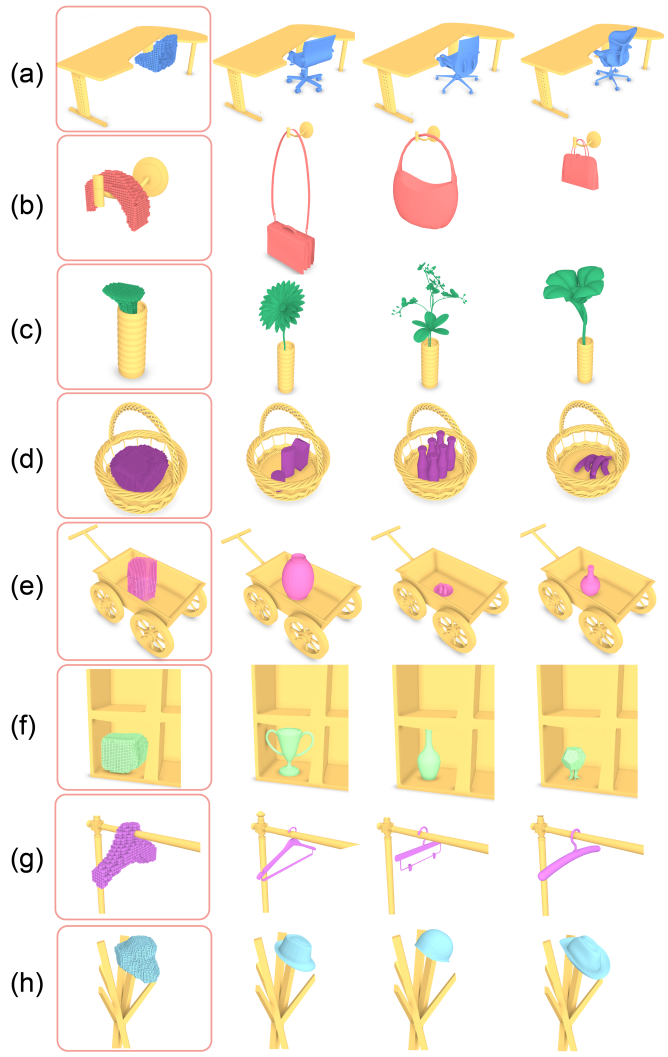


Fig. 14. Examples of predicted SDF (first image of each row) of interaction partner and matching results with three different models.

TABLE I
THE P-VALUES FOR EACH TYPE OF INTERACTIONS WHICH ARE USED FOR EVALUATING THE DIFFERENCE BETWEEN THE RATINGS GIVEN TO BOTH OUR RESULTS AND THE MANUAL MADE SCENES.

type	desk-chair	hook-bag	vase-flower	rack-hanger
p-value	0.2364	0.04621	0.466	0.3841
type	basket-objects	shelf-objects	cart-objects	stand-hats
p-value	0.007529	0.000158	0.01907	0.557

front side, but our method does not directly consider such orientation information in the last matching step, thus leads to results like the left scene of Fig. 18(b). All the images used for user study and the given score can be found in the supplementary material.

Timing. The training for the encoder-predictor network is done for each interaction type separately. It costs around 10 hours to train the model with 25,000 training samples and 100 epochs. During testing, it takes around 0.5 seconds to predict the SDF within each ROI. We use one GeForce GTX 1080 Ti graphics card for the experiment. For building final interactions, it takes up to 1 second to match each existing object to the predicted SDF.

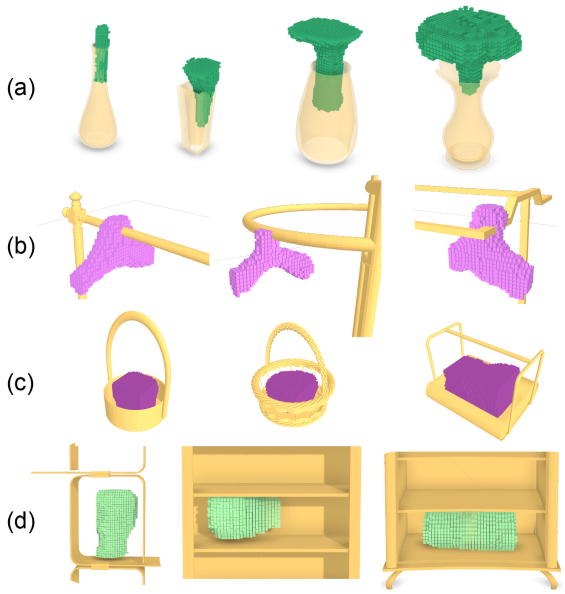


Fig. 15. Here we show how the predicted SDF adapt with different host objects.

VIII. DISCUSSION AND CONCLUSION

In this paper, we present an algorithm that can complete interactions by combining the strength of geometry analysis and network prediction. The main advantage of this work is that, unlike the previous template based method, our system is not constrained by a rigid template which may fail when applied to novel host objects with different geometry. The method can localize the interaction faster and with higher precision, and also be able to predict the interaction partner geometry that adapts to different geometry of the given host.

Our system can be useful for constructing novel scenes with multiple objects: the designer can roughly design the scenes by placing large objects in the scene, and let the system add other objects associated to them (see Fig. 1). This process can reduce the burden of the designer as they will only need to focus on high level design of the scene. The pairwise relations are dominant for most of the scenes, and our method can cover the majority of situations.

There are some limitations worth investigating further. As we mainly focus on the synthesis of pairwise interaction units, the relationship between multiple objects are not considered. When fitting an object into a large scene with multiple objects, it may be needed to consider its relation not only with one interaction host but with others. For example, in Fig. 1, a large static chair may result in inconvenience when taking things out from the bottom of the shelf. To cope with such a problem and handle larger scale scenes, the system needs to consider the relationship between multiple objects based on not only the geometry, but also the functionality and semantic information.

The second limitation of our method is that it does not produce new interaction partners. The candidate interaction

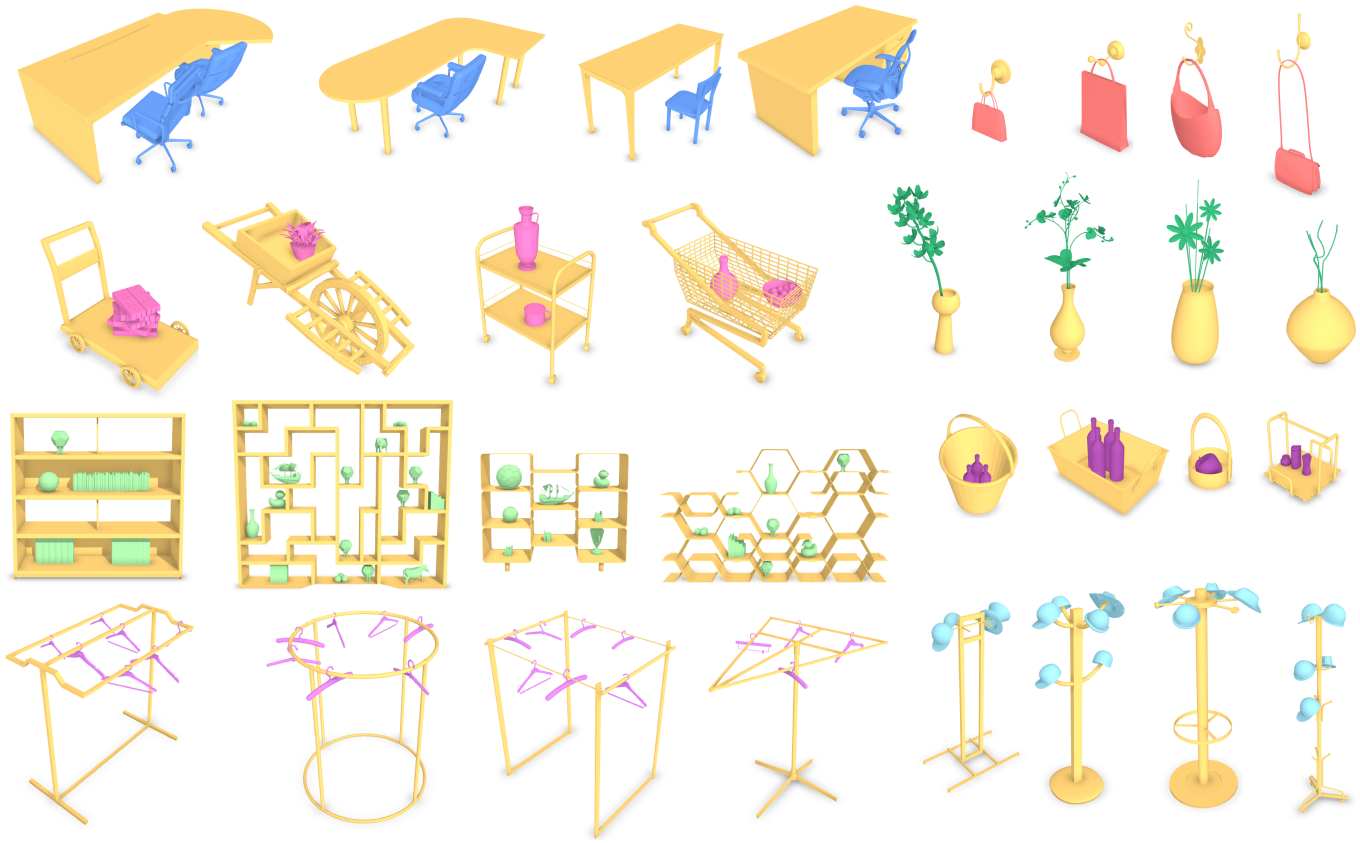


Fig. 16. Examples of our synthesized interactions.

partner objects may not fit well to a novel host and in some cases it might be ideal to produce new objects that uniquely fit to the host. It would be interesting to design more advanced generative networks for our second prediction step and directly create multiple partner models that fit well to the host and are compatible in terms of styles.

The third limitation is that our system simply conducts fitting based on the geometry and intrinsic information such as the front direction is not considered when fitting the interaction partner. This can affect the plausibility as shown in the left image of Fig. 18(b). For considering the front direction, we can either use the predefined front direction and encode this information into the interaction representation, or use learning based methods to predict the correct relative direction, so as to improve the quality of the final results.

ACKNOWLEDGMENT

The authors would like to thank Huibin Li, He Wang and Zhiqiang Tian who provide comments and suggestions at the early stage of this work. This work was supported in part by the China Postdoctoral Science Foundation (2015M582664), the National Natural Science Foundation for Young Scholars of China (61602366), the National Natural Science Foundation of China (61602311, 61872250) and Shenzhen Innovation Program (JCYJ20170302153208613).

REFERENCES

[1] M. Fisher, D. Ritchie, M. Savva, T. A. Funkhouser, and P. Hanrahan, "Example-based synthesis of 3d object arrangements," *ACM TOG*, vol. 31, no. 6, p. 135, 2012.
[2] K. Chen, Y.-K. Lai, Y.-X. Wu, R. Martin, and S.-M. Hu, "Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information," *ACM TOG*, vol. 33, no. 6, 2014.
[3] X. Zhao, R. Hu, P. Guerrero, N. Mitra, and T. Komura, "Relationship templates for creating scene variations," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 207:1–207:13, Nov. 2016.
[4] M. Fisher, M. Savva, and P. Hanrahan, "Characterizing structural relationships in scenes using graph kernels," *ACM TOG*, vol. 30, no. 4, p. 34, 2011.
[5] X. Zhao, H. Wang, and T. Komura, "Indexing 3d scenes using the interaction bisector surface," *ACM TOG*, vol. 33, no. 5, 2014.
[6] R. Hu, C. Zhu, O. van Kaick, L. Liu, A. Shamir, and H. Zhang, "Interaction context (icon): Towards a geometric functionality descriptor," *ACM TOG*, vol. 34, no. 4, pp. 83:1–83:12, 2015.
[7] S. Pirk, V. Krs, K. Hu, S. D. Rajasekaran, H. Kang, Y. Yoshiyasu, B. Benes, and L. J. Guibas, "Understanding and exploiting object interaction landscapes," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 1, Jul. 2017.
[8] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher, "Make it home: automatic optimization of furniture arrangement," *ACM TOG*, vol. 30, no. 4, p. 86, 2011.
[9] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-Centric Indoor Scene Synthesis Using Stochastic Grammar," p. 10.
[10] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and H. Fu, "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–13, Nov. 2017.
[11] L. Majerowicz, A. Shamir, A. Sheffer, and H. H. Hoos, "Filling your shelves: Synthesizing diverse style-preserving artifact arrangements," *IEEE TVCG*, vol. 20, no. 11, pp. 1507–1518, 2014.
[12] R. Hu, Z. Yan, J. Zhang, O. van Kaick, A. Shamir, H. Zhang, and H. Huang, "Predictive and generative neural networks for object functionality," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1:1–1:13, Aug. 2018.

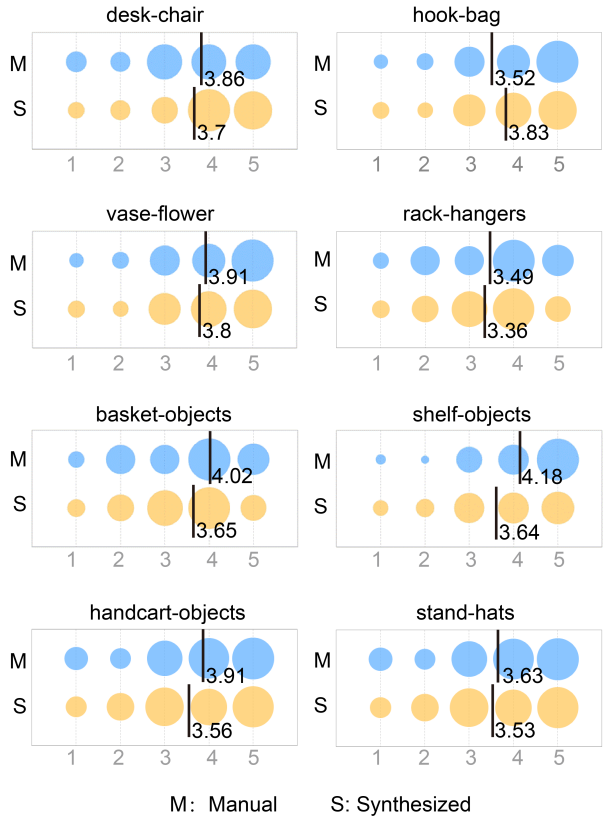


Fig. 17. Results of our online survey for evaluating the plausibility of our results (orange) vs the manually made scenes (blue). We show the distribution of the ratings for each interaction type by bubble chart. The bubbles correspondent to the marks of 1 to 5 from left to right. The size of each bubble represents the percentage of the corresponding rating. In each row, the vertical line shows the average score of the corresponding type of scenes.

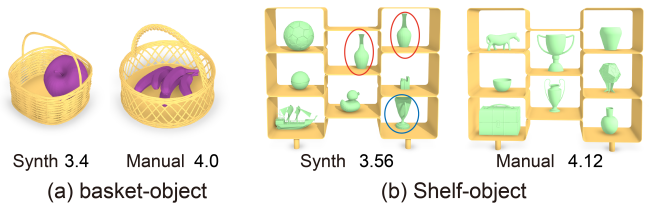
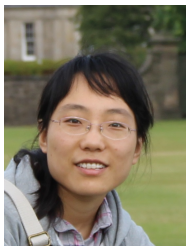


Fig. 18. Example scenes from the user study that our results get lower average score than manually made ones. The number below each scene is the average of the ratings given by the users. In (b), the red circles highlight the identical objects that appear in the shelf and the blue circle shows the trophy that is not facing front.

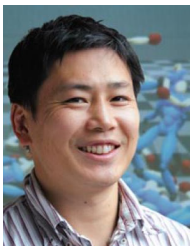
[13] K. Wang, M. Savva, A. X. Chang, and D. Ritchie, "Deep convolutional priors for indoor scene synthesis," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–14, Jul. 2018.
[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv:1311.2524 [cs]*, Nov. 2013, arXiv: 1311.2524.
[15] R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, p. 1137, 2017.
[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," pp. 21–37, 2015.
[19] J. Redmon and A. Angelova, "Real-Time Grasp Detection Using Convolutional Neural Networks," vol. 2015, pp. 1316–1322, 2014.
[20] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.
[21] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 190–198.
[22] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference," *arXiv:1709.07599 [cs]*, Sep. 2017, arXiv: 1709.07599.
[23] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 3, 2017.
[24] Y. Lamdan and H. J. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," in *ICCV*, vol. 88, 1988, pp. 238–249.
[25] J. A. Rice, *Mathematical Statistics and Data Analysis.*, 3rd ed. Belmont, CA: Duxbury Press., 2006.
[26] M. Savva, A. X. Chang, G. Bernstein, C. D. Manning, and P. Hanrahan, "On being the right scale: Sizing large collections of 3D models," in *SIGGRAPH Asia 2014 Workshop on Indoor Scene Understanding: Where Graphics meets Vision*, 2014.



Xi Zhao is a Lecturer in the department of computer science, Xian Jiaotong University, China. She received her Ph.D from School of Informatics, Edinburgh University and M.Sc from School of Computer Science and Engineering, Southeast University. Her research interests include shape analysis, geometry processing and interaction analysis.

Ruizhen Hu is an Assistant Professor at the College of Computer Science & Software Engineering, Shenzhen University, China. She received her Bachelor and Ph.D. degrees from the Department of Mathematics, Zhejiang University in 2010 and 2015. Her research interests include shape analysis, geometry processing and fabrication.

Haisong Liu is a third year undergraduate at the Xian Jiaotong University, China. He majors in Computer Science. In 2016 he won a gold medal in the ACM-ICPC China Shaanxi Provincial Contest. In 2017 he won a silver medal in the ACM-ICPC Asia Regional Contest Beijing Site. His research interests include computer graphics, machine learning, and parallel programming.



Taku Komura is a Reader (Associate Professor) at the Institute of Perception, Action and Behaviour, School of Informatics , Edinburgh University . He is also a Royal Society Industry Fellow. He received his B.Sc,M.Sc. and D.Sc in Information Science from the University of Tokyo. His research interests include character animation, computer graphics and interactive techniques.

Xinyu Yang received his Bachelor, Master and Ph.D degrees from Xian Jiaotong University in 1995, 1997 and 2001. He is currently a Professor in the department of computer science in Xian Jiaotong University. His research interests include multimedia modeling and mining, big data privacy protection.