Cross-modal Attribute Transfer for Rescaling 3D Models

Lin Shao* Angel X. Chang†* Hao Su*‡ Manolis Savva†* Leonidas Guibas*
*Stanford University †Princeton University ‡University of California San Diego

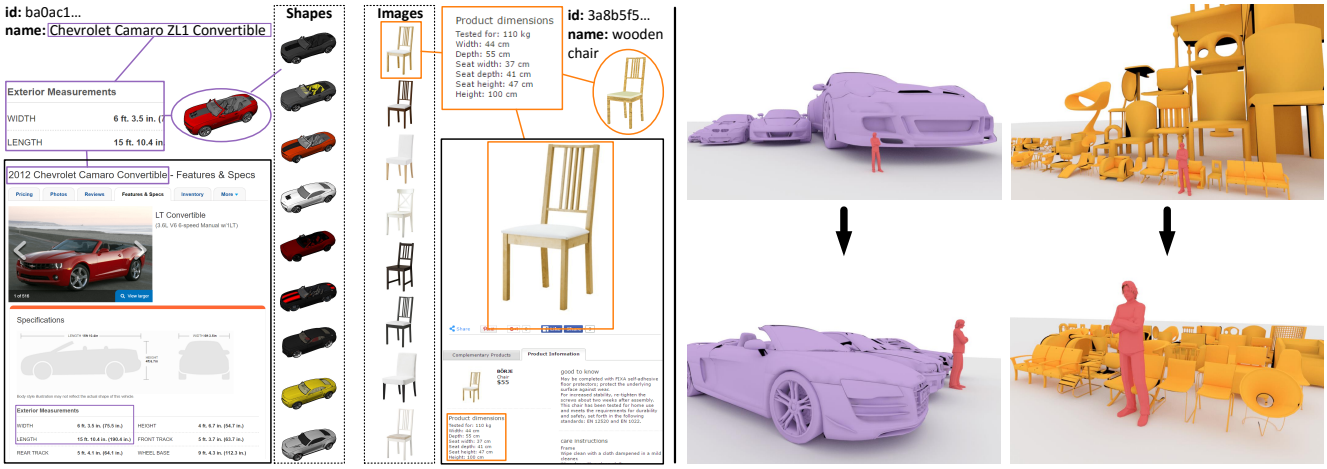

Figure 1: **Left**: we connect 3D models to webpages with physical attributes. Our approach transfers real-world dimensions by cross-modal linking through text or images (purple and orange links correspondingly). **Right**: we transfer real-world dimensions to a 3D model dataset and rectify physically implausible model scales.

Abstract

We present an algorithm for transferring physical attributes between webpages and 3D shapes. We crawl product catalogues and other webpages with structured metadata containing physical attributes such as dimensions and weights. Then we transfer physical attributes between shapes and real-world objects using a joint embedding of images and 3D shapes and a view-based weighting and aspect ratio filtering scheme for instance-level linking of 3D models and real-world counterpart objects. We evaluate our approach on a large-scale dataset of unscaled 3D models, and show that we outperform prior work on rescaling 3D models that considers only category-level size priors.

1. Introduction

3D model repositories such as TurboSquid[1] and the Trimble 3D Warehouse[2] have led to a proliferation of 3D data. Recent efforts in constructing large-scale annotated 3D model datasets have also significantly expanded the amount of 3D data available for research [2]. At the same time, the 3D model data in existing repositories is lacking many critical semantic attributes. Meaningful part segmentations, material annotations, real-world sizes, volumes and weights are some physically grounded attributes that are critical for reasoning in many computational tasks, but are absent or unreliable in existing 3D model datasets.

In contrast, physically grounded attributes such as the sizes, materials and weights of objects are some of the most important properties in commercial product information and websites. A promising link between semantic knowledge in the web, and 3D data is through 2D images, which are ubiquitous in websites.

We leverage recent advances in joint embedding of 3D shapes and images, to connect 3D object models with product info and images from websites creating a network of shapes and product info entries. Our contribution lies in using this network to tackle the problem of cross-modal attribute transfer between 3D models and other modalities. We demonstrate how a propagation algorithm can be used to enrich large-scale 3D shape collections with physical object attributes extracted from the web. Though our approach is applicable for transferring physical attributes in general, we focus on physical dimensions. We make our data and code public for the benefit of the community.

Contributions. We propose an algorithm for transferring

[1] https://www.turbosquid.com
[2] https://3dwarehouse.sketchup.com/

semantic attributes from product webpages describing real objects to 3D models of those objects. Our algorithm leverages a view-based loss scheme to robustly match 2D images and 3D shapes within local neighborhoods of a jointly embedded space. We collect a dataset of product information from the web and link it to 3D models at the instance level. We evaluate our method on this dataset by comparison to baselines for predicting the real world dimensions of 3D models. We demonstrate applications in object size and weight prediction from 2D images, 3D shape retrieval, and 3D scene design.

2. Related work

The use of 3D computer graphics models for computer vision tasks has become widespread. In particular, rendering of such models for training data has been shown to be useful in a variety of contexts including optical flow [6], object pose prediction [17, 7], and semantic segmentation [8, 18]. However, very few 3D CAD models in research datasets have physically accurate dimensions, making it hard to compose them into 3D scenes with plausible relative sizes. Our goal is to enrich such 3D models by connecting them to corresponding product info webpages.

A related line of recent work addresses amodal size prediction or amodal segmentation in RGB or RGB-D [10, 12, 5, 19]. These methods note the challenge of the amodal size prediction due to the difficulty of obtaining training data, since cropped or occluded objects in images need to be manually annotated. By providing dimensions for 3D models, we aim to make training data generation for these approaches easier.

Though there is much work in 3D shape analysis, it typically addresses 3D object models in isolation, with few attempts at connecting 3D models to other modalities such as text. An example is 3D Wikipedia [14] which connects references to parts of real-world environments in Wikipedia articles to reconstructed geometry of those real-world environments. We similarly establish a correspondence between geometry, and text or images but we do so for 3D models representing products to transfer physical attributes to the 3D models.

Recent advances in deep learning have enabled joint embedding of 3D shapes and images [13, 9], and joint embedding of object images and scene images [1]. Earlier work in natural language processing has extracted numerical attributes such as size and weight from web text [4]. However, connecting information from web text to transfer physical attributes between representations of objects in different modalities remains largely unexplored.

The closest prior work utilizes observations of 3D models in 3D scenes to propagate category-level size priors and plausibly rescale model collections [16]. However, their method mainly performs category-level size prediction and

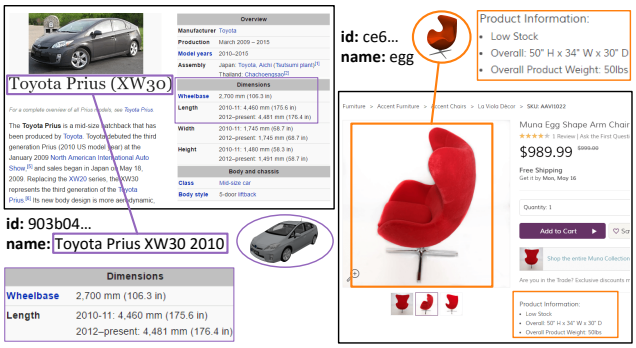

Figure 2: Product webpages linked to 3D models through text or images. **Left**: Toyota Prius XW30 Wikipedia page with dimension information matched to 3D model through name label. **Right**: chair product webpage matched to chair model through image.

can only do instance-level prediction for models observed in 3D scenes. We present an instance-level attribute transfer algorithm for 3D shapes of real-world products. Furthermore, we show that attributes can be transferred in both directions, enriching both the product images and corresponding 3D models.

3. Approach

Our goal is to connect physical attributes from websites describing real-world objects to 3D models of those particular objects. To do this we leverage two kinds of links: text and image links. The former relies on matching of text in a website describing the object with the text in the website from which a 3D model was retrieved to directly enrich the 3D model (see Figure 2 left). The latter relies on visual similarity of images found on product websites and rendered images of a 3D model (see Figure 2 middle). We describe the rationale for each type of link based on the properties of the 3D model and webpage data that we collect (see Section 4).

To enable transferring of physical attributes between product info websites and 3D models, we leverage recent work in joint embedding of shapes and images [13]. We adapt AlexNet (left) and modify the last fully connected layer (fc8) to predict the embedding features. We create a embedding space using a shape similarity metric and train CNNs to project real-world images and rendered 3D model images representing an object so that they are near each other (see Section 5.1).

After obtaining the embedding space we need an algorithm to transfer attributes between instances in the space. We build a network of local neighborhood around each instance to propagate attributes through nearest neighbor edges. Physical attributes that are associated with any instance in the network can be propagated to neighboring

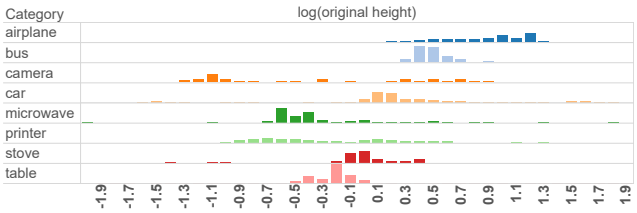

Figure 3: Several categories of 3D models in ShapeNetCore that exhibit unrealistically broad distributions of height values. The horizontal axis plots reported virtual unit height in log scale indicating that instances in each category span several orders of magnitude.

points using this transfer algorithm (see Section 5). The transfer process uses an aspect ratio filter to reduce the number of spurious matches (see Section 5.2) and handles variation in viewpoints by assigning higher weights to transfer pairs that have better matching viewpoints (see Section 5.3).

4. Data

To demonstrate our cross-modal attribute transfer algorithm we need both 3D shape data and webpage data describing the same real-world objects. There are far fewer 3D shapes than webpages describing objects so the choice of dataset domain is dictated largely by the available shape data. We leverage ShapeNet [2], a recently released large-scale repository which provides many 3D model instances of common object categories.

3D model data. We use the ShapeNetCore subset of ShapeNet which covers 55 object categories and contains about 51,000 models with consistent rigid alignments. One of our motivations is the observation that 3D models in this corpus—much like in other public repositories—are modeled in unknown or unreliable virtual unit scales (see Figure 3). In preliminary analysis we found that the categories in ShapeNet vary significantly in terms of the proportion of 3D model instances that correspond to real-world objects. For example, most car models have real-world counterparts and are named with the brand and make of car. In contrast, 3D models of tables are rarely obviously connected to real-world counterparts, and are therefore rarely named with brand and model information.

Due to this disparity in instance-level matching of 3D models to products we collected physical attributes from webpages with two pipelines: one directly retrieving webpages for matching 3D model instances (Table 1 left), and one crawling product catalogues for entire categories of objects (Table 1 right). In both cases, we sought webpages with structured physical attribute data for objects.

Webpage data. For the 3D model categories with many real-world instances, we run Google search queries given the name label provided for each ShapeNetCore model. If a Wikipedia entry or Google infobox entry is returned for the query, we extract any available dimension and weight values. About 80% of the searches were directly linked to Wikipedia entries in this manner, the remainder were manually linked to Google infobox values or values in other webpages. At the end of this process we have a set of seed 3D shapes matched to real-world counterpart objects with dimension attributes. For the categories with few recognizable instances, we crawl several furniture product catalogue websites to collect the webpage contents and product images[3]. These furniture websites contain product images along with product dimensions, weights and material descriptions. Since these webpages list the product information in semi-structured tables, we used patterns to identify and extract the dimensions and weights, taking care to normalize to consistent units (i.e. meters and kilograms), and to ensure consistent interpretation of width, length, and height. The total webpages for different categories are reported in Table 1. Note that the "cabinet" category refers to a category containing cabinets, cupboards, file cabinets and bookcases. With this data, our algorithm will then recover instance-level linking by matching product images to specific 3D models. Figure 2 illustrates the instance-level linking of 3D shapes to product webpages either directly through text (purple) as in the first pipeline, or through image-to-shape matching (orange).

	instance-level link			category-level link	
category	models	pages	category	models	pages
car	7497	1274	table	8443	5083
airplane	4045	1933	chair	6778	554
loudspeaker	1618	243	sofa	3173	1269
bus	939	94	cabinet	2336	4546
total	14099	3544	**total**	20730	11452

Table 1: Summary of 3D model and webpage data for our experiments. Our full data collection covered more categories (see supplemental material).

5. Algorithm

Here we describe the steps of our attribute transfer algorithm. As input, we take a set of shapes (3D models) $\mathcal{S}$ for which we want to estimate physical attributes. In addition, we have a set of items $\mathcal{I}$ collected from the web that contain the physical attributes we would like to transfer to the shapes. We start by identifying instance-level links between the models and the items to obtain a set of seed models $\mathcal{S}^+$, for which we can directly propagate the instance-level attributes. Next, we establish a joint embedding space using the shapes $\mathcal{S}$ and project images associated with the items into this space (see Section 5.1). Using this embedding, we

[3]We crawled www.ashleyfurniture.com, www.furniture.com, www.ikea.com, and www.wayfair.com.

establish a local neighborhood for each shape S_i and find the closest k annotated items or shapes as candidate sources for attribute transfer. Since the embedding space was not optimized for matching the attribute, we introduce a filtering step to refine match candidates. For the purposes of size transfer, we use an aspect ratio filter (Section 5.2). In addition, we introduce a re-ranking step to allow for more robust linking—in the image-to-shape transfer case this is a view-based weighting to account for viewpoint variation.

The flow of our cross-modal attribute transfer algorithm is summarized in Algorithm 1.

Algorithm 1: Cross-modal attribute transfer algorithm

input : Items $\mathcal{I} = \{I_a\}$ with annotated attributes
input : Initial set of shapes $\mathcal{S} = \{S_i\}$
`// Embedding construction`
Identify seed set $\mathcal{S}^+$ with known attributes by linking I_a to S_i
Establish embedding space D^- based on shapes $\{S_i\}$
Project 2D images of $\{I_a\}$ to embedding space D^-
`// Attribute prediction`
foreach *shape* S_i **in** $\mathcal{S}$ **do**
    Compute location P_i^- of S_i in D^-
    Find k closest annotated items $\{I_{a(i,j)}\}_{j=1}^k$
    Apply attribute match filter (AR filter)
    Re-rank items and select best matching I_i^*
    Transfer attribute from I_i^* to shape S_i
output Shapes $\{S_i\}$ with annotated attributes
:

Our algorithm implements a single hop transfer process. While diffusion processes or other iterative methods are interesting to consider for future work, it is not trivial to correlate rates of scalar diffusion with a similarity or distance metric in the embedded space. Furthermore, iterative processes can be prone to noise, drift, or regression to the mean. The single hop method we implement also has the advantage of being more attribute-agnostic.

In the following sections, we first describe the construction of the joint embedding space based on prior work (Section 5.1). Then we describe attribute transfer when corresponded seed shapes are available in which case shape-to-shape matching can be used for transfer (Section 5.2). Finally, we discuss the more challenging cross-modal attribute transfer when we have to establish links between images and shapes (Section 5.3).

5.1. Joint embedding of 3D models and images

We apply the method of Li et al. [13] to put shapes and images each object category into a joint embedding space.

5.1.1 Constructing the embedding space

Let $\mathcal{S} = \{S_i\}_{i=1}^n$ be the 3D model set. All our 3D models are assumed to be rigidly aligned with consistent up and front orientations. Every model S_i is rescaled and centered to fit tightly into a unit cube and is then rendered from m viewpoints to produce rendered images $\mathcal{I}_i = \{I_{i,v}\}_{v=1}^m$. In our experiments m is set to be 20 and viewpoints are evenly sampled around the up vector. We compute viewpoint feature vectors $H_{i,v}$ using Histogram of Gradients (HoG) [3]. These feature vectors consist of a three-level pyramid of HoG on images of resolution 120×120, 60×60 and 30×30 totaling 10,188 dimensions per viewpoint. These features are a baseline representation of shapes which can certainly be replaced by learned features. For every shape S_i, the full feature vector F_i is created by concatenating all m viewpoint feature vectors $H_{i,v}$ into $F_i = (H_{i,1}, H_{i,2}, ..., H_{i,m})$. The distance matrix D of $\mathcal{S}$ is generated by taking the L_2 distance between different shape feature vectors F_i.

The embedding space is based on non-linear multidimensional scaling (MDS) [11] with Sammon mapping [15] by minimizing the Sammon error during MDS:

$$E = \frac{1}{\sum_{i<j}\mathcal{D}(i,j)} \sum_{i<j} \frac{(\mathcal{D}(i,j) - D^-(i,j))^2}{\mathcal{D}(i,j)} \quad (1)$$

where $\mathcal{D}^-(i,j)$ is the distance matrix in the resulting embedding space. The result of this optimization is an embedding space of reduced dimensionality $\mathbb{R}^{128}$.

5.1.2 Embedding images

In order to project images into this space, we train a CNN to takes the images $\{I_i\}$ into corresponding points $\{P_i^-\}$ in the embedding space. To train the CNN, every shape S_i is rendered into a set of images $\{R_{i,k}\}$. These rendered images are associated with point $P_{S_i}^-$ of the shape in the embedding space. The rendered images vary in lighting and viewpoint as described by Li et al. [13].

The CNN encodes a function f which projects the images $\{R_{i,k}\}$ to $P_{S_i}^-$. The parameters θ of the CNN are optimized under the loss $L(\theta) = \sum_{i,k} \|f(R_{i,k};\theta) - P_{S_i}^-\|$. This CNN then embeds images into the space D^-.

5.1.3 Embedding shapes

To project new shapes into the joint embedding space we use an approach based on that of Li et al. [13] with the following modifications. Let S^* be a new shape model. A shape feature vector F^* is generated and the distance of S^* to every shape model S_i in the shape set $\{\mathcal{S}\}$ $d_{S^*,S_i} = \|F^* - F_i\|_2$ is computed. We want to find a corresponding point P_*^- in the embedding space. Let $d_{S^*,S_i}^- =$

$\|P_*^- - P_i^-\|$ be the corresponding distance in the embedding space. d_{min} is the minimum distance of d_{S^*,S_i} among all other shapes in $\mathcal{S}$. P_*^- is solved by L-BFGS minimization of the following objective function:

$$P_{S^*}^- = \underset{P^-}{\operatorname{argmin}} \sum_i^n \frac{(d_{S^*,S_i} - d_{S^*,S_i}^-)^2}{d_{S^*,S_i} - d_{min}/2} \tag{2}$$

Here n is set to be 400 in our experiments instead of the number of shapes in $\mathcal{S}$. We only take into account the closest 400 models as they are the most similar and distances to dissimilar models are less important. Since the target points are in $\mathbb{R}^{128}$ taking 400 neighbors as constraints is a reasonable compromise. The subtraction of the minimum distance in the denominator makes the range of the summation broader thus better bringing out contrasts between points. These modifications to the objective make the optimization more efficient and result in better projections based on our experiments.

5.2. Shape-to-shape attribute transfer

We use shape-to-shape transfer when we have a nonempty seed shape set $\mathcal{S}^+$ with known physical attributes. We then transfer these attributes to nearby models using the algorithm as described with no special re-ranking (i.e., the L_2 distance in embedding space directly determines the ranking of neighbors).

Aspect Ratio Filter. For every item and shape with length l, width w, and height h we compute the normalized aspect ratio values denoted by $(l/d, w/d, h/d)$ where d is the diagonal length. This aspect ratio filter is a simple threshold on the dot product of two normalized aspect ratio vectors. A shape and item are only allowed to match if they pass this aspect ratio filter threshold. This threshold is a user-defined parameter that balances transfer accuracy and transfer coverage. Higher values improve accuracy but reduce coverage. We set this threshold to values between 0.95 and 0.996 with higher values for shape categories with less geometry variation (e.g., cabinets and sofas), and lower values for categories with more variation (e.g., chairs).

5.3. Image-to-shape attribute transfer

Transferring attributes between images and shapes is a challenging task. When there is no clear match through text associated with the shape and the webpage we have to establish visual appearance links. This is where we leverage the common space provided by the joint embedding. A key difference compared to the case of shape-to-shape transfer is that images capture only one 2D view whereas shapes are 3D representations. Here, we discuss refinements to the general transfer algorithm in order to better handle viewpoint variation. These refinements allow for re-ranking the neighborhood of a shape in the embedding space to better match images with particular viewpoints.

Algorithm 2: Image-to-shape attribute transfer algorithm

input : Images $\mathcal{I} = \{I_a\}$ with annotated attributes
input : Initial set of shapes $\mathcal{S} = \{S_i\}$
`// Embedding construction`
Establish joint embedding space D^- based on shapes $\{S_i\}$
Project 2D images $\{I_a\}$ to joint embedding space and calculate HoG feature vectors $\{H_a\}$.
`// View loss weights`
Render every model S_i to images under l different viewpoints and calculate HoG feature vectors $\{H_{r(i,s)}\}_{s=1}^l$
Calculate view loss scores $\{w_s\}_{s=1}^l$
`// Attribute prediction`
foreach *shape* S_i **in** $\mathcal{S}$ **do**
  Compute location P_i^- of S_i in D^-
  Find top k nearest items $\{I_{a(i,j)}\}_{j=1}^k$
  Apply Aspect Ratio (AR) filter
  Select image I_i^* where $I_i^* = \underset{j=1:k}{\operatorname{argmin}}\{\underset{s=1:l}{\operatorname{argmin}}\, w_s \|H_{a(i,j)} - H_{r(i,s)}\|_1\}$
  Transfer the diagonal length of item I_i^* to shape S_i
output Shapes $\{S_i\}$ with annotated attributes
:

We first normalize every model S_i to a unit cube and render it into images $\{I_{r(i,s)}\}_{s=1}^l$ from evenly spaced viewpoints at 10° around the up vector at elevations of 5°, 10°, 15° and 20°. Then we compute feature vectors of the rendered images $\{I_{r(i,s)}\}_{s=1}^l$ and the webpage images $\{I_{a(i,j)}\}_{j=1}^k$ as described in Section 5.1.1. We denote these feature vectors by $\{H_{r(i,s)}\}_{s=1}^l$ and $\{H_{a(i,j)}\}_{j=1}^k$ respectively. We then compute the pairwise L_1 distances of the image feature vectors $\{H_{r(i,s)}\}_{s=1}^l$ and $\{H_{a(i,j)}\}_{j=1}^k$. Finally, we select the most similar image I_i^* through:

$$I_i^* = \underset{j=1:k}{\operatorname{argmin}}\{\underset{s=1:l}{\operatorname{argmin}}\, w_s \|H_{a(i,j)} - H_{r(i,s)}\|_1\} \tag{3}$$

Here, $\{w_s\}_{s=1}^l$ represent weights associated with each viewpoint which we refer to as the view-based loss. These weights capture the intuition that images from different viewpoints should be weighted differently when matching the same object. A summary of the refined transfer algorithm is given in Algorithm 2.

Handling viewpoint variation. In formulating the view-based loss we seek to assign higher weights to more informative viewpoints. More informative viewpoints are ones that better reflect shape geometry and thus capture the similarity ranking between shapes. Conditioning this notion of informativeness on particular attributes to be transported would better capture correlations between view and promi-

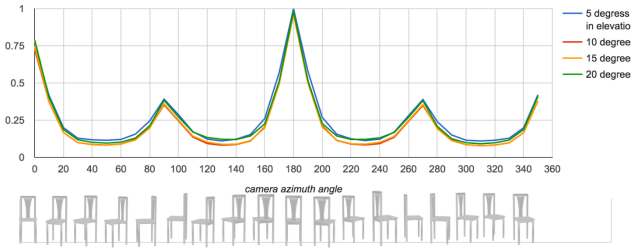

Figure 4: View-based loss scores for chairs. The horizontal axis plots camera azimuth angle starting in front and moving clockwise around—the view is visualized below in rendered images. The vertical axis plots the loss score. Colors represent camera elevations.

nence of an attribute (e.g., width vs height). We take a simple attribute-agnostic approach.

We use the rendered images $\{I_{r(i,s)}\}_{s=1}^{l}$ to create feature vectors $\{H_{(i,s)}\}_{s=1}^{l}$ for each 3D shape $\mathcal{S} = \{S_i\}$. We define the shape descriptor F_{S_i} to be a concatenation of the image feature vectors $(H_{(i,1)}, H_{(i,2)}, ..., H_{(i,l)})$ as in Section 5.1.1. We construct a distance matrix using the L_1 distance $d_s(S_i, S_j) = \|F_{(i)} - F_{(j)}\|_1$ on these descriptors. Based on the distance matrix, we compute a ranking matrix R_s where $R_s(i,k)$ is the index of the k-th nearest shape to shape S_i.

For a given viewpoint v, we compute the distance matrix $D_v(S_i, S_j) = \|H_{(i,v)} - H_{(j,v)}\|_1$. This distance $d_v(S_i, S_j)$ represents differences between shape S_i and shape S_j in rendered images under viewpoint v. A ranking matrix R_v associated with viewpoint v is also computed based on d_v. $R_v(i,k)$ is the index of the k-th nearest shape to shape S_i based on the distance matrix D_v. $R_v(i,j)$ returns the ranking number of shape S_j to shape S_i.

The loss score for viewpoint v is computed based on the difference between R_s and R_v. For every shape S_i we find the top m nearest shapes based on R_s. Then we find these top m nearest shapes' ranking numbers inR_v. These ranking numbers are compared divided by the distances between these top m nearest shapes and shape S_i:

$$\text{LossScore(v)} = \sum_{S_i} \sum_{k=1}^{m} \frac{|k - R_v(i, R_s(i,k))|}{d_s(S_i, S_{R_s(i,k)})} \qquad (4)$$

In our experiments, we set $m = 50$. The computed view loss scores are then divided by the maximum loss score among all viewpoints to obtain a normalized $[0, 1]$ value. High loss values indicate more 3D geometry information is lost from a given view.

Figure 4 plots the view loss scores for chairs. Chairs exhibit bilateral symmetry and that the most informative camera angle in the front-right viewpoints is $50°$ azimuth and $10 - 15°$ elevation.

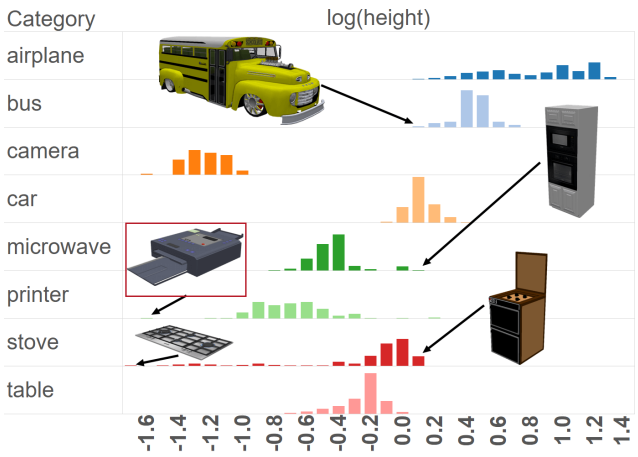

Figure 5: Heights predicted using our approach in log scale for several categories of 3D models (cf. Figure 3). The height distributions are now narrower and more realistic. Some outliers such as the short stove top surface, and the tall microwave embedded in a cabinet are reasonably sized. The short printer is an example of an error case due to a particularly long diagonal length.

6. Evaluation

Figures 1 and 5 show qualitatively the impact of our size predictions (note the failure case in the printer category, and see supplemental for more results). Before rescaling we see that many objects in a category have drastically different dimensions. For example, chairs can be much shorter or much taller than a person.

To quantitatively evaluate our algorithm we run a series of experiments measuring the accuracy of predicted dimensions against known ground truth dimensions. Each experiment addresses particular attribute transfer source and target models. Experiment A evaluates *shape-to-shape* transfer when attributed 3D models are available as sources for dimension propagation. Experiment B evaluates *image-to-shape* transfer which we use to establish links in categories with few 3D models of known dimensions—in this case product images are the sources of dimension information and shapes in the embedding space are the targets. Experiment C evaluates image-to-shape transfer to a set of new 3D models that first have to be projected into the embedding space. In all three experiments, we compare against the category-based estimates used by prior work [16], and also isolate the effect of the aspect ratio filter and view-based loss components of our algorithm.

Experimental setup. We first create a set of 3D models with known ground truth dimensions by manually verifying correct instance-level matching to product webpages in our corpus. Some of these ground truth models are part of the ShapeNetCore dataset used for constructing the embedding

Method	airplane	car	loudspeaker	bus	**All**
Prior	62.6	8.5	55.3	22.6	43.5
AR	67.8	13.6	70.0	33.8	48.0
JE	**6.9**	8.5	37.5	10.0	12.9
JE+AR	7.0	**8.5**	**36.1**	**10.0**	**11.8**
# models	364	229	43	17	653

Table 2: Mean relative diagonal length error for Experiment A (shape-to-shape transfer).

space, while others are external models that were previously unobserved and have to be projected into the space. For the latter set we collected 638 models of chairs, tables, sofas and cabinets from the 3D Warehouse[4].

In both cases we apply our algorithm to transfer dimensions to each ground truth model from its local neighborhood in the embedding. We then compare the transferred sizes to the ground truth values. We calculate a relative diagonal size error $|L_{gt}^{dia} - L_{pred}^{dia}|/L_{gt}^{dia}$ where L_{gt}^{dia} and L_{pred}^{dia} are the diagonal lengths of the ground truth and predicted dimensions correspondingly.

We compare our algorithm against two baseline methods. The Prior baseline simply assigns the mean diagonal length value of the object category to each instance of the category. We compute the mean diagonal length of all objects with size annotations within each category. This method is equivalent to the category-based priors of previous work. The aspect ratio (AR) baseline embeds 3D model and product instances in an $\mathbb{R}^3$ space constructed from the three normalized aspect ratio values. This baseline isolates the benefit of the joint embedding (JE) space over a simpler alternative.

Experiment A: shape-to-shape transfer. This experiment evaluates the scenario when a seed set of 3D shapes with known dimensions is the transfer source and the targets are the remaining shapes used during embedding space construction. We first split the ShapeNetCore models with verified dimensions annotated into a 70% training set and a 30% test set. We carry out the embedding stage of our algorithm using the models in the training set. Then we carry out the prediction step on the test set and compare the predicted dimensions with the known ground truth dimensions. Table 2 reports the percentage relative error averages.

As we expected, the JE and JE+AR methods significantly outperform the Prior and AR baselines for most categories. The car category exhibits less overall size variance than the other categories, leading to the Prior method performing particularly well in that case. The aspect ratio filter component in JE+AR does not significantly improve results in this experiment. Comparing JE and JE+AR, the AR filter does not significantly improve the transfer error. This is not particularly surprising as models are compared using LFD-HoG global features which already take global shape differences into consideration (see Section 5.1.1).

[4] https://3dwarehouse.sketchup.com

Method	chair	table	sofa	cabinet	**All**
Prior	21.3	21.2	17.2	26.1	20.5
AR	16.0	17.2	**11.3**	18.1	14.7
JE	14.2	17.3	14.8	22.5	16.6
JE+VL	14.2	13.4	16.6	19.9	16.1
JE+AR	11.2	16.0	11.8	21.3	14.3
JE+AR+VL	**11.0**	**12.2**	11.7	**17.8**	**12.8**
# models	123	115	243	111	592

Table 3: Mean relative diagonal length error for Experiment B (image-to-shape transfer).

Method	chair	table	sofa	cabinet	**All**
Prior	29.2	21.3	12.6	24.7	24.2
AR	22.8	46.5	17.8	34.5	32.8
JE	14.4	16.5	14.9	21.6	15.3
JE+VL	14.7	13.6	14.8	39.8	21.9
JE+AR	11.4	14.6	12.4	21.6	15.4
JE+AR+VL	**11.4**	**13.6**	**9.9**	**19.9**	**14.5**
# models	209	186	51	192	638

Table 4: Mean relative diagonal length error for Experiment C (new model projection and transfer).

Experiment B: image-to-shape transfer. In this experiment we address the categories where there are no seed shapes with known sizes. Webpage product images with known dimensions are used as the transfer sources and the targets are shapes already in the embedded space. For the test set, we verify the dimensions of 592 chair, table, sofa and cabinet models in ShapeNetCore by matching them to crawled product webpages. Table 3 reports the mean relative diagonal error values for all methods.

Again, the results show that overall our embedding transfer approach outperforms the simpler baselines. The sofa category is particularly well suited to the AR baseline (and JE+AR filtering) as it consists of a discrete set of clusters with characteristic aspect ratios (one-and-a-half "loveseat", two seat, three seat, four seat etc.). In contrast, the performance on the table and cabinet categories is much more sensitive to the view loss component.

Experiment C: projecting new models. For this experiment, we use an external test set of 638 models with known dimensions that were not part of the embedding space construction. We first project these models using the method described in Section 5.1.3 and then proceed with the same pipeline as before. This experiment demonstrates the practically useful scenario where a newly seen 3D model has

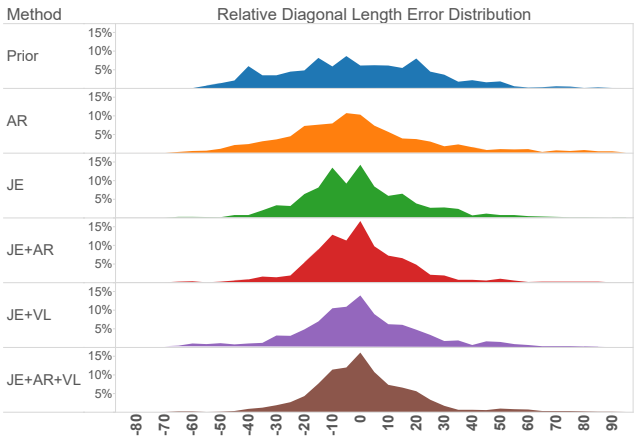

Figure 6: Distribution of relative diagonal length errors for size predictions by each method averaged over all evaluation categories. The category-based prior (Prior) and aspect ratio space (AR) methods have broad distributions indicating predictions with high relative error. Our approach using joint embedding (JE) and its additional component have increasingly narrower distributions indicating more accurate predictions. The total area under 10% error (reflecting expectation for under 10% error) for each method is as follows. Prior: 21.0%, AR: 28.4%, JE: 31.7%, JE+AR: 37.5%, JE+VL: 33.8%, JE+AR+VL: 38.7%.

to be handled by a pre-trained embedding space. Table 4 reports mean errors for all methods.

The benefit of our approach is clear in this setting. Note that the AR baseline in particular does not generalize well for new models. In contrast, JE and its variants perform at similar error levels as the previous experiment.

6.1. Overall results

The results of our evaluation indicate that our transfer algorithm predicts real-world dimensions for 3D models more accurately than prior work and simpler baseline approaches. In particular, experiment C handling previously unobserved 3D models demonstrates that simple category-based priors (Prior) and aspect ratio–based transfer (AR) do not generalize well and are unlikely to be robust in practical prediction scenarios. This conclusion is reflected when visualizing the distribution of relative diagonal errors for each of the methods averaged over all evaluation categories (see Figure 6).

The ablative comparison of the components of our algorithm shows that aspect ratio filtering (JE+AR) improves transfer accuracy in general but the effect is limited in shape-to-shape transfer. This is likely due to the fact that the LFD-HoG global shape descriptors we use for constructing our embedding space already take global shape aspect ratio into consideration. The view-based loss component

Figure 7: **Left**: Chair image from Amazon product catalogue. Predicted dimensions are $0.660\,\mathrm{m} \times 0.610\,\mathrm{m} \times 1.041\,\mathrm{m}$ (length $\times$ width $\times$ height) and predicted weight is $14.1\,\mathrm{kg}$ (cf. specified product dimensions of $0.685\,\mathrm{m} \times 0.673\,\mathrm{m} \times 1.067\,\mathrm{m}$ and $18.1\,\mathrm{kg}$. **Right**: Table image from Amazon. Predicted dimensions are $0.548\,\mathrm{m} \times 1.097\,\mathrm{m} \times 0.680\,\mathrm{m}$ and $17.7\,\mathrm{kg}$. The product size and weight are $0.406\,\mathrm{m} \times 1.097\,\mathrm{m} \times 0.680\,\mathrm{m}$ and $15.2\,\mathrm{kg}$.

(JE+VL) also improves transfer accuracy in general though it is sensitive to particular categories. The improvement is complementary to JE+AR as demonstrated by the further improvement when the two are combined in JE+AR+VL.

Predicting size and weight for images. We demonstrate an application of our algorithm for size and weight prediction from object images. Additional applications are provided in the supplemental material. Figure 7 shows product images of a chair and a table retrieved from Amazon (not in our training or testing data) and their predicted size and weight values with our algorithm by matching to the nearest neighbor 3D model in our embedding. The predicted values closely match the actual values. As a quantitative evaluation, we predicted sizes for 15 chair product images, obtaining a mean diagonal error of 8.3% (compared to error of 12.7% using the baseline category-level prior).

7. Conclusion

In this paper, we proposed an algorithm for transferring physical attributes from product webpages describing real objects to 3D models of those objects. We collected a large dataset of product information from the web and linked it to 3D models at the instance level. We evaluated our algorithm by comparison to several baselines for predicting real world sizes of 3D models. We illustrated how our approach can be used to predict sizes for objects given 2D image inputs.

We showed that cross-modal transfer of physical attributes such as size and weight are possible through a joint embedding scheme linking 3D shapes and webpage metadata through 2D images. Prediction of real-world size for 3D models is critical for enabling generation of realistic 3D scene training data, and for simulation and interaction in the emerging VR/AR application areas. Physical attributes are also an important prerequisite for truly connecting the visual world to richer semantics.

Acknowledgments The authors acknowledge the support of NSF grants IIS-1528025, and DMS-1546206, MURI grant N00014-13-1-0341, and a Samsung GRO award.

References

[1] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. on Graphics (SIGGRAPH)*, 34(4), 2015. 2

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1, 3

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 4

[4] D. Davidov and A. Rappoport. Extraction and approximation of numerical attributes from the web. In *In Proc. of 48th ACL*. ACL, 2010. 2

[5] Z. Deng and L. J. Latecki. Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[6] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, T. Brox, et al. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766. IEEE, 2015. 2

[7] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4731–4740, 2015. 2

[8] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. *arXiv preprint arXiv:1511.07041*, 2015. 2

[9] M. Hueting, M. Ovsjanikov, and N. J. Mitra. CrossLink: joint understanding of image and 3D model collections through shape and camera pose variations. *ACM Transactions on Graphics (TOG)*, 34(6):233, 2015. 2

[10] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Amodal completion and size constancy in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 127–135, 2015. 2

[11] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964. 4

[12] K. Li and J. Malik. Amodal instance segmentation. In *European Conference on Computer Vision*, pages 677–693. Springer, 2016. 2

[13] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via cnn image purification. *ACM Trans. Graph.*, 2015. 2, 4

[14] B. C. Russell, R. Martin-Brualla, D. J. Butler, S. M. Seitz, and L. Zettlemoyer. 3D Wikipedia: Using online text to automatically label and navigate reconstructed geometry. *ACM Transactions on Graphics (SIGGRAPH Asia 2013)*, 32(6), November 2013. 2

[15] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, (5):401–409, 1969. 4

[16] 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. 2, 6

[17] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3D model views. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[18] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[19] Y. Zhu, Y. Tian, D. Mexatas, and P. Dollár. Semantic amodal segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2