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

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 TurboSquid1 and the Trimble 3D Warehouse2 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

1https://www.turbosquid.com
2https://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.](image)
We use the ShapeNetCore subset of 3D model data. 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.

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. 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).

### 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

---

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 \( I = \{I_n\} \) with annotated attributes  
**input** : Initial set of shapes \( S = \{S_i\} \)  
// Embedding construction  
1. Identify seed set \( S^+ \) with known attributes by linking \( I_n \) to \( S_i \)  
2. Establish embedding space \( D^- \) based on shapes \( \{S_i\} \)  
3. Project 2D images of \( \{I_i\} \) to embedding space \( D^- \)  
// Attribute prediction  
4. foreach shape \( S_i \) in \( S \) do  
5. Compute location \( P^*_i \) of \( S_i \) in \( D^- \)  
6. Find \( k \) closest annotated items \( \{I_{a(i,j)}\} \)  
7. Apply attribute match filter (AR filter)  
8. Re-rank items and select best matching \( I^*_i \)  
9. 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 corresponding 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.
$\|P^*_s - P_i^*\|$ be the corresponding distance in the embedding space. $d_{\text{min}}$ is the minimum distance of $d_{S^*,S_i}$ among all other shapes in $S$. $P^*_s$ is solved by L-BFGS minimization of the following objective function:

$$P^*_s = \arg\min_{P^*} \sum_i (d_{S^*,S_i} - d_{\hat{S}^*,S_i})^2 / d_{S^*,S_i} - d_{\text{min}}/2$$ \hspace{1cm} (2)

Here $n$ is set to be 400 in our experiments instead of the number of shapes in $S$. 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 $S^*$ 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.

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 $10^\circ$ around the up vector at elevations of $5^\circ$, $10^\circ$, $15^\circ$ and $20^\circ$. 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:

$$I^*_i = \arg\min_{j=1:k} \{ \arg\min_{s=1:L} w_s \|H_{a(i,j)} - H_{r(i,s)}\|_1 \}$$ \hspace{1cm} (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-

---

**Algorithm 2: Image-to-shape attribute transfer algorithm**

```
input : Images $I = \{I_a\}$ with annotated attributes
input : Initial set of shapes $S = \{S_i\}$

// Embedding construction
1 Establish joint embedding space $D^-$ based on shapes $\{S_i\}$
2 Project 2D images $\{I_a\}$ to joint embedding space and calculate HoG feature vectors $\{H_a\}$.

// View loss weights
3 Render every model $S_i$ to images under $l$ different viewpoints and calculate HoG feature vectors $\{H_{r(i,s)}\}_{s=1}^L$
4 Calculate view loss scores $\{w_s\}_{s=1}^L$

// Attribute prediction
5 foreach shape $S_i$ in $S$ do
6 Compute location $P^*_i$ of $S_i$ in $D^-$
7 Find top $k$ nearest items $\{I_{a(i,j)}\}_{j=1}^k$
8 Apply Aspect Ratio (AR) filter
9 Select image $I^*_i$ where
   \[ I^*_i = \arg\min_{j=1:k} \{ \arg\min_{s=1:L} w_s \|H_{a(i,j)} - H_{r(i,s)}\|_1 \} \]
10 Transfer the diagonal length of item $I^*_i$ to shape $S_i$
```

output Shapes $\{S_i\}$ with annotated attributes:
distance \( d \) represents differences between shape \( S \). Based on the distance matrix, we compute a ranking matrix \( R \) as in Section 5.1.1. We construct a distance matrix using the shape descriptor \( F \) based on the distance matrix \( D \), defined as \( d_{s}(S_i, S_j) = ||F(i) - F(j)||_1 \) on these descriptors. Based on the distance matrix, we compute a ranking matrix \( R \) 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 in \( R_v \). These ranking numbers are compared divided by the distances between these top \( m \) nearest shapes and shape \( S_i \):

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

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 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.

### 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.
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. 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 \( \frac{L_{\text{gt}} - L_{\text{pred}}}{L_{\text{gt}}} \) where \( L_{\text{gt}} \) and \( L_{\text{pred}} \) 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).

**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

<table>
<thead>
<tr>
<th>Method</th>
<th>airplane</th>
<th>car</th>
<th>loudspeaker</th>
<th>bus</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>62.6</td>
<td>8.5</td>
<td>55.3</td>
<td>22.6</td>
<td>43.5</td>
</tr>
<tr>
<td>AR</td>
<td>67.8</td>
<td>13.6</td>
<td>70.0</td>
<td>33.8</td>
<td>48.0</td>
</tr>
<tr>
<td>JE</td>
<td>6.9</td>
<td>8.5</td>
<td>37.5</td>
<td>10.0</td>
<td>12.9</td>
</tr>
<tr>
<td>JE+AR</td>
<td>7.0</td>
<td>8.5</td>
<td>36.1</td>
<td>10.0</td>
<td>11.8</td>
</tr>
</tbody>
</table>

# models 364 229 43 17 653

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

<table>
<thead>
<tr>
<th>Method</th>
<th>chair</th>
<th>table</th>
<th>sofa</th>
<th>cabinet</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>21.3</td>
<td>21.2</td>
<td>17.2</td>
<td>26.1</td>
<td>20.5</td>
</tr>
<tr>
<td>AR</td>
<td>16.0</td>
<td>17.2</td>
<td>11.3</td>
<td>18.1</td>
<td>14.7</td>
</tr>
<tr>
<td>JE</td>
<td>14.2</td>
<td>17.3</td>
<td>14.8</td>
<td>22.5</td>
<td>16.6</td>
</tr>
<tr>
<td>JE+VL</td>
<td>14.2</td>
<td>13.4</td>
<td>16.6</td>
<td>19.9</td>
<td>16.1</td>
</tr>
<tr>
<td>JE+AR</td>
<td>11.2</td>
<td>16.0</td>
<td>11.8</td>
<td>21.3</td>
<td>14.3</td>
</tr>
<tr>
<td>JE+AR+VL</td>
<td>11.0</td>
<td>12.2</td>
<td>11.7</td>
<td>17.8</td>
<td>12.8</td>
</tr>
</tbody>
</table>

# models 123 115 243 111 592

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

<table>
<thead>
<tr>
<th>Method</th>
<th>chair</th>
<th>table</th>
<th>sofa</th>
<th>cabinet</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>29.2</td>
<td>21.3</td>
<td>12.6</td>
<td>24.7</td>
<td>24.2</td>
</tr>
<tr>
<td>AR</td>
<td>22.8</td>
<td>46.5</td>
<td>17.8</td>
<td>34.5</td>
<td>32.8</td>
</tr>
<tr>
<td>JE</td>
<td>14.4</td>
<td>16.5</td>
<td>14.9</td>
<td>21.6</td>
<td>15.3</td>
</tr>
<tr>
<td>JE+VL</td>
<td>14.7</td>
<td>13.6</td>
<td>14.8</td>
<td>39.8</td>
<td>21.9</td>
</tr>
<tr>
<td>JE+AR</td>
<td>11.4</td>
<td>14.6</td>
<td>12.4</td>
<td>21.6</td>
<td>15.4</td>
</tr>
<tr>
<td>JE+AR+VL</td>
<td>11.4</td>
<td>13.6</td>
<td>9.9</td>
<td>19.9</td>
<td>14.5</td>
</tr>
</tbody>
</table>

# models 209 186 51 192 638

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

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

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

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 (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.

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


