Learning Material-Aware Local Descriptors for 3D Shapes

Hubert Lin[1] Melinos Averkiou[2] Evangelos Kalogerakis[3] Balazs Kovacs[4]
Siddhant Ranade[5] Vladimir G. Kim[6] Siddhartha Chaudhuri[6,7] Kavita Bala[1]
[1]Cornell Univ. [2]Univ. of Cyprus [3]UMass Amherst [4]Zoox [5]Univ. of Utah [6]Adobe [7]IIT Bombay

Abstract

Material understanding is critical for design, geometric modeling, and analysis of functional objects. We enable material-aware 3D shape analysis by employing a projective convolutional neural network architecture to learn material-aware descriptors from view-based representations of 3D points for point-wise material classification or material-aware retrieval. Unfortunately, only a small fraction of shapes in 3D repositories are labeled with physical materials, posing a challenge for learning methods. To address this challenge, we crowdsource a dataset of 3080 *3D shapes with part-wise material labels. We focus on furniture models which exhibit interesting structure and material variability. In addition, we also contribute a high-quality expert-labeled benchmark of* 115 *shapes from Herman-Miller and IKEA for evaluation. We further apply a mesh-aware conditional random field, which incorporates rotational and reflective symmetries, to smooth our local material predictions across neighboring surface patches. We demonstrate the effectiveness of our learned descriptors for automatic texturing, material-aware retrieval, and physical simulation.*

1. Introduction

Materials and geometry are two essential attributes of objects that define their function and appearance. The shape of a metal chair is quite different from that of a wooden one for reasons of robustness, ergonomics, and manufacturability. While recent work has studied the analysis and synthesis of 3D shapes [32, 48], no prior work directly addresses the inference of *physical* (as opposed to appearance-driven) materials from geometry.

Jointly reasoning about materials and geometry can enable important applications. Many large online repositories of 3D shapes have been developed [9], but these lack tags that associate object parts with physical materials which hampers natural queries based on materials, e.g., predicting which materials are commonly used for object parts, retrieving objects composed of similar materials, and simulating how objects behave under real-world physics. Robotic perception often needs to reason about materials: a glass tumbler must be manipulated more gently than a steel one, and a hedge is a softer emergency collision zone for a self-driving car than a brick wall. The color channel may be unreliable (e.g., at night), and the primary input is geometric depth data from LiDAR, time-of-flight, or structured light scanners. Interactive design tools can suggest a feasible assignment of materials for fabricating a modeled shape or indicate when a choice of materials would be unsuitable.

A key challenge in these applications is to reason about plausible material assignments *from geometry alone*, since color textures are often (in model repositories) or always (in night-vision robotics or interactive design) absent or unreliable. Further, material choices are guided by functional, aesthetic, and manufacturing considerations. This suggests that material assignments cannot be inferred simply from physical simulations, but require real-world knowledge.

In this paper, we address these challenges with a novel method to compute *material-aware descriptors* of 3D points directly from geometry. First, we crowdsource per-part material labels for 3D shapes. Second, we train a projective convolutional network [18] to learn an embedding of geometric patches to a material-aware descriptor space. Third, we curate a benchmark of shapes with expert-labeled material annotations on which our material descriptors are evaluated.

Learning surface point descriptors for 3D shape data has been explored in previous approaches for 3D shape segmentation [34] and correspondences [50, 18]. However, there are challenges with such an approach for our task. First, 3D shape datasets are limited in size. While the largest image dataset with material labels comprises ∼437K images [6], there is *no* shape dataset with material labels. Second, many 3D shapes in repositories have missing, non-photorealistic, or inaccurate textures (e.g., a single color for the whole shape). Material metadata is rarely entered by designers for 3D models. Therefore, it is difficult to automatically infer material labels. Third, gathering material annotations is a laborious task, especially for workers who do not have a strong association of untextured models with corresponding real-world objects. We address these challenges by designing a crowdsourcing task that enables effective collection of material annotations.

Our contributions are the following:

- The *first large database* of 3D shapes with per-part physical material labels, comprising a smaller expert-labeled benchmark set and a larger crowdsourced train-

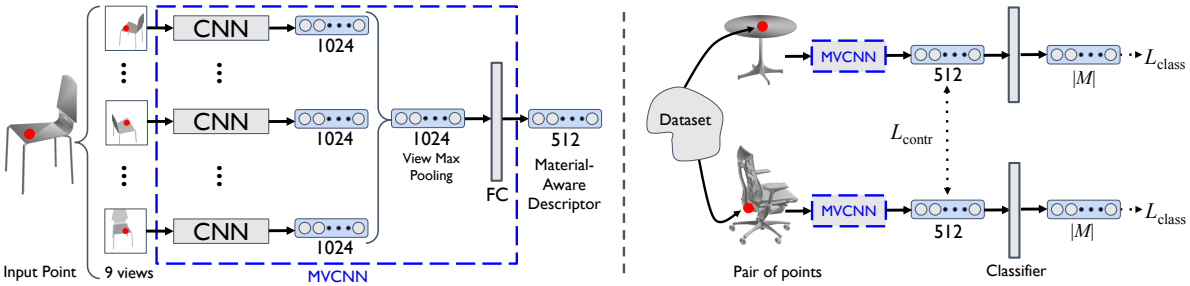


Figure 1: Left: Our MVCNN takes as input nine renderings of a point from multiple views and scales, passes them through a CNN which produces 1024-dimensional descriptors, which are then max-pooled over views and passed through a fully connected layer which gives the final 512-dimensional descriptor. Right: The training pipeline for pairs of points. The network is trained in a Siamese fashion.

ing set, and a *crowdsourcing strategy* for material labeling of 3D shapes.

- A new deep learning approach for extracting *material-aware local descriptors* of surface points of untextured 3D shapes, along with a symmetry-aware CRF to make material predictions more coherent.
- Prototype *material-aware applications* that use our descriptors for automatic texturing, part retrieval, and physical simulation.

2. Previous work

We review work on material prediction for shapes and images, as well as deep neural networks for shapes.

Material prediction for shapes. To make material assignment accessible to inexperienced users Jain *et al*. [19] proposed a method for automatically predicting reflective properties of objects. This method relies on a database of 3D shapes with known reflective properties, and requires data to be segmented into single-material parts. It uses light-field shape descriptors [10] to represent part geometry and a graphical model to encode object or scene structure. Wang *et al*. [44] proposed a method for transferring textures from object images to 3D models. This approach relies on aligning a single 3D model to a user-specified image and then uses inter-shape correspondence to assign the texture to more shapes. Chen *et al*. [12] used color and texture statistics in images of indoor scenes to texture a 3D scene. They require the scene to be segmented into semantic parts and labeled. All these methods focus on visual attributes and appeal of the final renderings. In our work we focus on classifying physical materials, and do not assume any additional user input such as an image that matches a 3D shape, or a semantic segmentation. Savva *et al*. [37] construct a large repository of 3D shapes with a variety of annotations, including category-level priors over material labels (e.g., "chairs are on average 42% fabric, 40% wood, 12% leather, 4% plastic and 2% metal") obtained from annotated image datasets [5]. The priors are not specific to individual shapes. Chen *et al*. [11] gather natural language descriptions for 3D shapes that sometimes include material labels ("This is a brown wooden chair"), but there is no fine-grained region labeling that can be used for training. Yang *et al*. [48] propose a data-driven algorithm to reshape shape components to a target fabrication material. We aim to produce component-independent material descriptors that can be used for a variety of tasks such as classification, and we consider materials beyond wood and metal.

Material prediction for images. Photographs are a rich source of the appearance of objects. Image-based material acquisition and measurement has been an active area for decades; a comprehensive study of image-based measurement techniques can be found in [46]. Material prediction "in the wild", i.e., in uncontrolled non-laboratory settings, has recently gained more interest fueled by the availability of datasets like the Flickr Materials Database [27, 38], Describable Textures Dataset [13], OpenSurfaces [5], and Materials in Context [6]. Past techniques identified features such as gradient distributions along image edges [27], but recently deep learning has set new records for material recognition ([13, 6]). In our work, we focus on renderings of untextured shapes rather than photographs of real world scenes.

Deep neural networks for shape analysis. A variety of neural network architectures have been proposed for both global (classification) and local (segmentation, correspondences) reasoning about 3D shapes. The variety of models is in large part due to the fact that unlike 2D images, which are almost always stored as raster grids of pixels, there is no single standard representation for 3D shapes. Hence, neural networks based on polygon meshes [31, 7], 2D renderings [41, 20, 18], local descriptors after spectral alignment [49], unordered point sets [45, 25, 34, 35, 40], canonicalized meshes [30], dense voxel grids [47, 14, 33], voxel octrees [36, 43], and collections of surface patches [15], have been developed. Bronstein *et al*. [8] provide an excellent survey of spectral, patch and graph-based approaches. Furthermore, methods such as [26] have been proposed for dense shape correspondences. Our goal is to learn features that reflect physical material composition, rather than representations that reflect geometric or semantic similarity. Our specific architecture derives from projective, or multi-view

convolutional networks for local shape analysis [20, 18], which are good at identifying fine-resolution features (e.g., feature curves on shapes, hard/smooth edges) that are useful to discriminate between material classes. However, our approach is conceptually agnostic to the network used to process shapes, and other volumetric, spectral, or graph-based approaches could be used instead.

3. Data Collection

We collected a crowd-sourced training set of 3080 3D shapes annotated with per-component material labels. We also created a benchmark of 115 3D shapes with verified material annotations to serve as ground-truth. Both datasets will be made publicly available.

3.1. 3D Training Shapes

Our 3D training shapes originate from the ShapeNet v2 repository [9]. We picked shapes from three categories with interesting material and structural variability: 6778 chairs, 8436 tables and 1571 cabinets. To crowd-source reliable material annotations for these shapes, we further pruned the original shapes as follows.

First, observing that workers are error-prone on textureless shapes, we removed shapes that did not include any texture references. These account for 32.2% of the original shapes. Second, to avoid relying on crowd workers for tedious manual material-based mesh segmentation, we only included shapes with pre-existing components (i.e., groups in their OBJ format). We also removed over-segmented meshes (> 20 components), since these tended to have tiny parts that are too laborious to label. Meshes without any, or with too many components accounted for an additional 17.1% of the original shapes. Third, to remove low-quality meshes that often resulted in rendering artifacts and further material ambiguity, we pruned shapes with fewer than 500 triangles/vertices (another 30.8% of the dataset). Finally, after removing duplicates, the remaining shapes were 1453 chairs, 1460 tables, and 218 cabinets, summing to a total of 3131 shapes to be labelled.

To gather material annotations for the components of these shapes, we created questionnaires released through the Amazon Mechanical Turk (MTurk) service. Each questionnaire had 20 queries (see supplementary for interface). Four different rendered views covering the front, sides and back of the textured 3D shape were shown. At the foot of the page, a single shape component was highlighted. Each query highlighted a different component. Workers were asked to select a label from a set of materials $\mathcal{M}$ for the highlighted component. The set of materials was $\mathcal{M} = \{$*wood*, *plastic*, *metal*, *glass*, *fabric (including leather)*$\}$. We selected this set to cover materials commonly found in furniture available in ShapeNet. We deliberately did not allow workers to select multiple materials to ensure they picked the most plausible material given the textured component rendering. We also provided a "null" option, with associated text "cannot tell / none of the above" so users could flag components whose material they found impossible to guess. Our original list of materials also included *stone*, but workers chose this option only for a small fraction of components (∼0.5%), and thus we excluded it from training and testing. Preliminary versions of the questionnaire showed that users often had trouble telling apart *metal* from *plastic* components. We believe the reason was that *metal* and *plastic* components sometimes have similar texture and color. Thus, we provided an additional option "*metal* or *plastic*". We note that our training procedure utilized this multi-material option, which is still partially informative as it excludes other materials.

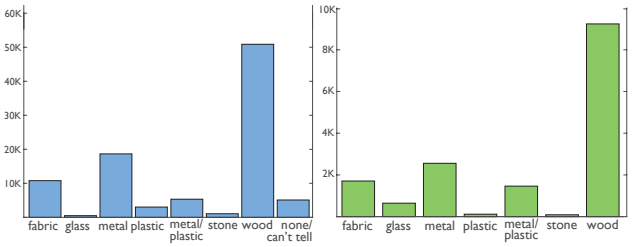


Figure 2: (a) Distribution of answers in our questionnaires (b) Distribution of material labels assigned to the components of our training dataset.

Out of the 20 queries in each questionnaire, 3 of them were "sentinels" i.e., components whose correct material was clearly visible, unambiguous and confirmed by us. We used these sentinels to check worker reliability. Users who incorrectly labelled any sentinel or selected "null" for more than half of the questions were ignored. In total, 7370 workers participated in our data collection out of which 23.7% were deemed as "unreliable". All workers were compensated with $0.25 for completing a questionnaire. On average, 5.2 minutes were spent per questionnaire.

Each component received 5 answers (votes). The distribution of votes is shown in Figure 2(a). If 4 or 5 out of 5 votes for a component agreed, we considered this a consensus vote. 15835 components received such consensus, of which 547 components had consensus on the "null" option. Thus, 15288 components (out of 19096 i.e., 80.1%) acquired material labels. We further checked and included 635 components with transparent textures and confirmed they were all glass. In total, we collected 15923 labeled components in 3080 shapes. The distribution of material labels is shown in Figure 2(b). For training, we kept only shapes with a majority of components labeled (2134 shapes).

3.2. 3D Benchmark Shapes

The 3D benchmark shapes originated from Herman Miller's online catalog [2] and 3D Warehouse [1]. All shapes were stored as meshes and chosen because they had explicit references to product names and descriptions from a corresponding manufacturer: IKEA [3] or Herman Miller. This dataset has 40 chairs, 47 tables and 28 cabinets. Expert annotators assigned material labels to all shape components

through direct visual reference to corresponding manufacturers' product images as well as information from the textual product descriptions. Such annotation is not scalable, hence this dataset is relatively small and used purely for evaluation. See supplementary for distribution of labeled parts.

4. Network Architecture and Training

Our method trains a convolutional network that embeds surface points of 3D shapes in a high-dimensional descriptor space. To perform this embedding, our network learns "material-aware" descriptors for points through a multi-task optimization procedure.

4.1. Network architecture

To learn material-aware descriptors, we use the architecture visualized in Figure 1. The network follows a multi-view architecture [20, 18]. Other architectures could also be considered, e.g. volumetric [47, 50], spectral [31, 7, 8], or point-based [34, 40].

We follow Huang *et al.*'s [18] multi-view architecture. We render 9 images around each surface point s with a Phong shader and a single directional light along the viewing axis. The rendered images depict local surface neighborhoods around each point from distances of 0.25, 0.5 and 1.0 times the shape's bounding sphere radius. The camera up vectors are aligned with the shape's upright axis, as we assume shapes to be consistently upright-oriented. The viewpoints are selected to maximize surface coverage and avoid self-occlusion [18]. In Huang *et al.*'s architecture [18], the images per point are processed through AlexNet branches [23] Because view-based representations for 3D shapes are somewhat similar to 2D images, we chose to use GoogLeNet [42] instead, which achieved strong results for 2D material recognition [6]. Alternatives like VGG [39] yielded no notable differences. We tried rendering 36 views as in Huang *et al.*'s work, but since ShapeNet shapes are upright-oriented, we found that 9 upright-oriented views were equivalent.

In our GoogLeNet-based MVCNN, we aggregate the 1024D output from the 7x7 pooling layer after "inception 5b" for each of our 9 views with a max view-pooling layer [41]. This aggregated feature is reduced to a 512D descriptor. A subsequent classification layer and sigmoid layer compute classification scores. For training, all parameters are initialized with the trained model from [6], except for the dimensionality reduction layer and classification layer whose parameters are initialized randomly from a Gaussian distribution with mean 0 and standard deviation 0.01.

Structured material predictions. Figure 3 visualizes the per-point material label predictions for a characteristic input mesh. Note that self-occlusions and non-discriminative views can cause erroneous predictions. Further, symmetric parts (e.g., left and right chair legs) lack consistency in material predictions, since long-range dependencies between surface points are not explicitly considered in our network. Finally, network material predictions are limited only to surface points, and not throughout the whole shape.

To address these challenges, the last part of our architecture incorporates a structured probabilistic model, namely a Conditional Random Field [24] (CRF). The CRF models both local and long-range dependencies in the material predictions across the input surface represented as a polygon mesh, and also projects the point-based predictions onto the input mesh. We treat the material predictions on the surface as binary random variables. There are $|\mathcal{M}|$ such variables per input polygon, each indicating the presence/absence of a particular material. Note that this formulation accommodates multi-material predictions.

Our CRF incorporates: (a) unary factors that evaluate the probability of polygons to be labeled according to predicted point material labels, (b) pairwise factors that promote the same material label for adjacent polygons with low dihedral angle, (c) pairwise factors that promote the same material label for polygons whose geodesic distance is small, (d) pairwise factors that promote the same material label for polygons related under symmetry. Specifically, given all surface random variables $\mathbf{C}_s$ for an input shape s, the joint distribution is expressed as follows:

$$P(\mathbf{C}_s) = \frac{1}{Z_s} \prod_{m,f} \phi_{\text{unary}}(C_{m,f}) \prod_{m,f,f' \in Adj} \phi_{\text{adj}}(C_{m,f}, C_{m,f'}) \prod_{m,\,f,f'} \phi_{\text{dist}}(C_{m,f}, C_{m,f'}) \prod_{m,\,f,f'} \phi_{\text{sym}}(C_{m,f}, C_{m,f'})$$

where $C_{m,f}$ is the binary variable indicating if face f is labeled with material m, and Z_s is a normalization constant. The unary factor sets the label probabilities of the surface point nearest to face f according to the network output. The pairwise factors $\phi_{\text{adj}}(C_{m,f}, C_{m,f'})$ encode pairwise interactions between adjacent faces, following previous CRFs for mesh segmentation [20]. Specifically, we define a factor favoring the same material label prediction for neighboring polygons (f, f') with similar normals. Given the angle $\omega_{f,f'}$ between their normals ($\omega_{f,f'}$ is divided by π to map it between $[0, 1]$), the factor is defined as follows:

$$\phi_{adj}(C_{m,f} = l, C_{m,f'} = l') = \begin{cases} \exp\left(-w_{m,a} \cdot w_{m,l,l'} \cdot \omega_{f,f'}^2\right), & l = l' \\ \exp\left(-w_{m,a} \cdot w_{m,l,l'} \cdot (1 - \omega_{f,f'}^2)\right), & l \neq l' \end{cases}$$

where l and l' represent the $\{0, 1\}$ binary labels for adjacent faces $\{f, f'\}$, $w_{m,a}$ and $w_{m,l,l'}$ are learned factor- and material-dependent weights. The factors $\phi_{\text{adj}}(C_{m,f}, C_{m,f'})$ favor similar labels for polygons f, f' which are spatially close (according to geodesic distance $d_{f,f'}$) and also belong to the same connected component:

$$\phi_{dist}(C_{m,f} = l, C_{m,f'} = l') = \begin{cases} \exp\left(-w_{m,d} \cdot w_{m,l,l'} \cdot d_{f,f'}^2\right), & l = l' \\ \exp\left(-w_{m,d} \cdot w_{m,l,l'} \cdot (1 - d_{f,f'}^2)\right), & l \neq l' \end{cases}$$

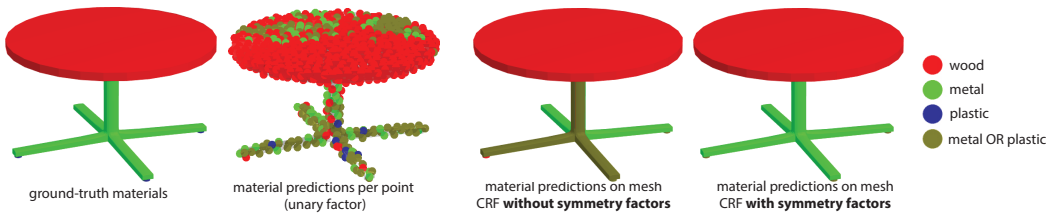


Figure 3: From left to right: Ground Truth, Per-Point Prediction from network, Post-CRF without symmetry factors, Post-CRF with symmetry factors. Note that the material labeling of our CRF with symmetry is almost in total agreement with ground-truth materials except for the plastic table leg caps (bottom of the legs) which are plastic, but are labeled as "metal or plastic" from our CRF.

where the weights $w_{m,d}$ and $w_{m,l,l'}$ are learned factor- and material-dependent parameters, and $d_{f,f'}$ represents the geodesic distance between f and f', normalized to $[0,1]$.

Finally, our CRF incorporates symmetry-aware factors. We note that such symmetry-aware factors were not considered before in other CRF-based mesh segmentation approaches. Specifically, our factors $\phi_{\text{symm}}(C_{m,f}, C_{m,f'})$ favor similar labels for polygons f, f' which are related under a symmetry. We detect rotational and reflective symmetries between components by matching surface patches through ICP, extracting their mapping transformations, and grouping them together when they undergo a similar transformation following Lun *et al*. [28, 29]. The symmetry-aware factors are expressed as:

$$\phi_{symm}(C_{m,f}=l, C_{m,f'}=l') = \begin{cases} \exp\left(-w_{m,s}\cdot w_{m,l,l'}\cdot s^2_{f,f'}\right), & l=l' \\ \exp\left(-w_{m,s}\cdot w_{m,l,l'}\cdot (1-s^2_{f,f'})\right), & l\neq l' \end{cases}$$

where the weights $w_{m,s}$ and $w_{m,l,l'}$ are learned factor- and label-dependent parameters, and $s_{f,f'}$ expresses the Euclidean distance between face centers after applying the detected symmetry.

Exact inference in this probabilistic model is intractable. Thus we use mean-field inference to approximate the most likely joint assignment to all random variables (Algorithm 11.7 of [22]). Figure 3 shows material predictions over the input mesh after performing inference in the CRF with and without symmetry factors.

4.2. Training

To train the network, we sample 150 evenly-distributed surface points from each of our 3D training shapes. Points lacking a material label, or externally invisible, are discarded. The remaining points are subsampled to 75 per shape to fit memory constraints. The network is trained end-to-end with a multi-task loss function that includes a multi-class binary cross-entropy loss for material classification and a contrastive loss [16] to align 3D points in descriptor space [31] according to their underlying material (Figure 1 (right)). Specifically, given: (i) a set of training surface points $\mathcal{S}$ from 3D shapes, (ii) a "positive" set $\mathcal{P}$ consisting of surface point pairs labeled with the same material label, (iii) a "negative" set $\mathcal{N}$ consisting of surface point pairs that do not share any material labels, (iv) binary indicator values $t_{m,p}$ per training point $p \in \mathcal{S}$ and label $m \in \mathcal{M}$ (equal to 1 when p is labeled with label m, 0 otherwise), the network parameters $\mathbf{w}$ are trained according to the following multi-task loss function:

$$\mathbf{L}(\mathbf{w}) = \lambda_{\text{class}}\mathbf{L}_{\text{class}}(\mathbf{w}) + \lambda_{\text{contr}}\mathbf{L}_{\text{contr}}(\mathbf{w})$$

The loss function is composed of the following terms:

$$\mathbf{L}_{\text{class}}(\mathbf{w}) = \sum_{p\in\mathcal{S}}\sum_{m\in\mathcal{M}} [t_{m,p}\log P(C_{m,p}=1 \mid \mathbf{f}_p, \mathbf{w}) + (1-t_{m,p})\log(1-P(C_{m,p}=1 \mid \mathbf{f}_p, \mathbf{w}))]$$

$$\mathbf{L}_{\text{contr}}(\mathbf{w}) = [\sum_{p,q\in\mathcal{P}} D^2(\mathbf{f}_p, \mathbf{f}_q) + \sum_{p,q\in\mathcal{N}} \max(M - D(\mathbf{f}_p, \mathbf{f}_q), 0)^2]$$

where $P(C_{m,p}=1 \mid \mathbf{f}_p, \mathbf{w})$ represents the probability of our network to assign the material m to the surface point p according to its descriptor $\mathbf{f}_p$, $D^2(\mathbf{f}_p, \mathbf{f}_q)$ measures squared Euclidean distances between the normalized image and surface point descriptors, and M is a margin typically used in constrastive loss (we set it to $\sqrt{0.2} - 0.2$). The loss terms have weights $\lambda_{\text{class}} = 0.016$ & $\lambda_{\text{contr}} = 1.0$ which were selected empirically to balance the terms to have same order of magnitude during training time. We will refer to the network optimized with this loss as "Multitask". We also experiment with a variant that utilizes solely classification loss. In this case $\lambda_{\text{class}} = 1.0$ & $\lambda_{\text{contr}} = 0.0$. We will refer to this network as "Classification". Note that in both Multitask and Classification, the classification layer is trained with an effective loss weight of $\lambda_{\text{class}} = 1.0$. For Multitask, the learning rate multiplier of the classification layer is increased to compensate for $\lambda_{\text{class}} = 0.016$.

Multitask training is performed with Adam [21] with learning rate 0.001, $\beta_1 = 0.9, \beta_2 = 0.999$. The network is trained in cycles of $100K$ iterations. We choose a stopping point when losses converge on a validation set, which occurs by the end of the second cycle. In order to optimize for the contrastive loss, the network is trained in a Siamese fashion with two branches that share weights (see Fig. 1 right). Classification training is performed through stochastic gradient descent with momentum. The initial learning rate

is set to 0.001 and momentum is set to 0.9. The learning rate policy is polynomial decay with power 0.5. L_2 weight decay is set to 0.0002. We train Classification for two cycles of $100K$ iterations. In the second cycle, the initial learning rate is reduced to 0.0001 and momentum is reduced to 0.4. Note that variants of both optimization procedures were tried for both loss functions and that we only report the optimal settings here. We also note that we tried contrastive-only loss but it did not perform as well as the variants here.

During training, point pairs are sampled from $\mathcal{P}$ and $\mathcal{N}$ with a 1:4 ratio with the intuition that learning to separate different classes in descriptor space is more difficult than grouping the same class together. To balance material classes, we explicitly cycle through all material pair combinations when sampling pairs. For example, if we sample a negative *wood-glass* pair, subsequent negative pairs will not be *wood-glass* until all other combinations have been sampled. Because it is possible for points to have multiple ground truth labels (e.g. *metal or plastic*), we ensure that negative pairs do not share any ground truth labels. For example, if we try to sample a *plastic-metal* pair and we draw a *metal or plastic* point paired with a *metal* point, this pair would be discarded and re-sampled until a true negative *plastic-metal* pair is drawn. On a Pascal Titan X GPU, training with batchsize 5 takes about 12 hours per $100K$ iterations.

CRF training. The CRF module is trained to maximize the log-likelihood of the material labelings in our training meshes [22] on average:

$$L = \frac{1}{|\mathcal{T}|} \sum_{s \in \mathcal{T}} \log P(\mathbf{C}_s = \mathbf{T}_s)$$

where $\mathbf{T}_s$ are ground-truth binary material labels per polygon in the training shape s from our training shape set $\mathcal{T}$. We use gradient descent and initialize the CRF weights to 1 [20]. Training takes $\sim$8 hours on a Xeon E5-2630 v4 processor.

5. Results

5.1. Evaluation

We evaluate our approach in a series of experiments. For our test set, we sample 5K evenly-distributed surface points from each of 115 benchmark test shapes. We discard externally invisible points, and evenly subsample the rest to 1024 points per shape. Our final test set consists of 117K points. See supplementary for distribution of points.

Material-aware descriptors. Mean precision for k nearest neighbor retrievals is computed by averaging the number of neighbors that share a ground truth label with the query point over the number of retrieved points (k). Nearest neighbors are retrieved in descriptor space from a class-balanced subset of training points. Mean precision by class is computed by computing the mean precision for subsets of the test set containing only test points that belong to the class of interest. Table 1 summarizes the mean precision at varying values of k. Both Classification and Multitask variations achieve similar mean class precision at all values of k. Furthermore, note that the Multitask variation achieves better precision than the Classification variation over all values of k in every class except for wood. We believe this is likely because the contrastive loss component of multi-task loss encourages distance between clusters of dissimilar classes while classification-only loss encourages the clusters to be separable without necessarily being far apart. Therefore it is less likely for Multitask descriptors to have nearby neighbors from a different cluster.

	Mean	Wood	Glass	Metal	Fabric	Plastic
Classif.						
k=1	55.7	76.4	34.3	65.0	56.1	46.7
k=30	56.9	75.3	41.1	64.9	55.3	47.6
k=100	57.3	75.1	43.0	64.9	55.5	48.0
Multitask						
k=1	56.2	62.2	40.8	68.6	58.0	51.2
k=30	56.2	61.0	42.6	68.9	57.4	51.1
k=100	56.6	60.7	44.7	68.7	57.4	51.5

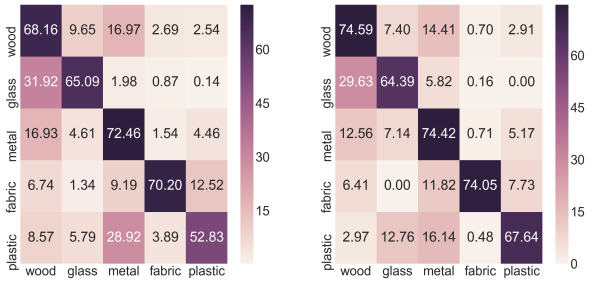
Table 1: Precision (%) at k nearest neighbors in descriptor space.

Material prediction. To demonstrate that our descriptors are useful for material classification, we evaluate the learned classifier on our test shapes. We measure the top-1 accuracy per material label. The top-1 accuracy of a label prediction for a given point is 1 if the point has that label according to ground-truth, and 0 otherwise. If the point has multiple ground-truth labels, the accuracy is averaged over them. The top-1 accuracy for a material label is computed by averaging this measure over all surface points. These numbers are summarized in Table 2.

We note that both Classification and Multitask variations produce similar mean class top-1 accuracies. However, the Classification variation exhibits a larger variance in its top-1 class accuracies, with better wood accuracy in exchange for worse glass and fabric accuracies compared to the Multitask variation. After applying the CRF, both variations have improved top-1 accuracies for all classes except for glass. Glass prediction accuracy remains almost the same for the Multitask variation, but drops drastically for Classification. We suspect that this occurs because glass parts sometimes share similar geometry with wooden parts in furniture (for example, flat tabletops or flat cabinet doors may be made of either glass or wood). In this case, several point-wise predictions will compete for both glass and wood. If more of these predictions are wood rather than glass, it is likely the CRF will smooth out the predictions towards wood, which will result in performance drop for glass predictions. Fig. 4 shows top-1 prediction confusion matrices. Wood points are often predicted correctly, yet sometimes are confused with metal. Glass points are often confused with wood points. Fabric is occasionally confused with plastic or wood. These confusions often happen for chairs with thin leather backs or thin seat cushions. Plastic is occasionally confused with

Network	Mean	Wood	Glass	Metal	Fabric	Plastic
MINC-bw	38	1.2	38	65	20	65
Classif.	65	82	53	72	62	55
C+CRF	66	85	36	77	66	65
Multitask	66	68	65	72	70	53
MT+CRF	71	75	64	74	74	68

Table 2: Top-1 material classification accuracy (%). Baseline MINC-bw is MINC [6] trained on greyscale photos and tested on untextured 3D renderings.



(a) Multitask (b) Multitask + CRF

Figure 4: Confusion matrices for Top-1 classification predictions. Rows are ground truths and columns are predictions.

metal. This is due to parts that are thin rounded cylinders often used in both metal and plastic-made components. Furthermore, the proportion of *plastic* labels to *"metal or plastic"* labels is low in our training dataset, which makes the learning less reliable in the case of plastic. In both variations, there is a bias towards wood predictions. This is likely due to the abundance of wooden parts in real-world furniture designs reflected in our datasets. However, the bias is less pronounced in the Multitask variation. Thus we believe that the Multitask variation is better for a more balanced generalization performance across classes.

Effect of Number of Views. To study the effect of the number of views, we train the MVCNN with 3 views (1 viewpoint, 3 distances) and compare to our results above with 9 views (3 viewpoints, 3 distances): see Table 3. Multiple viewpoints are advantageous.

5.2. Material-aware Applications

We illustrate the utility of the material-aware descriptors learned by our method in some prototype applications.

Texturing. Given the material-aware segmentation produced by our method, we can automatically texture a 3D mesh based on the predicted material of its faces. Such a tool can be used to automate texturing for new shapes or for collectively texturing existing shape collections. If the mesh does not have UV coordinates, we generate them automatically by simultaneous multi-plane unwrapping. Then, we apply a texture to each mesh face according to the physical material predicted by the material-aware segmentation. We have designed representative textures for each of the physical materials predicted by our method (wood, plastic, metal, glass, fabric). Resulting renderings for a few of the meshes from our test set can be seen in Figure 5.

Network	Mean	Wood	Glass	Metal	Fabric	Plastic
C 3view	59	81	41	71	60	40
C 9view	65	82	53	72	62	55
MT 3view	56	45	71	85	65	15
MT 9view	66	68	65	72	70	53

Table 3: Top-1 classification accuracy (%) for 3 views vs 9 views.

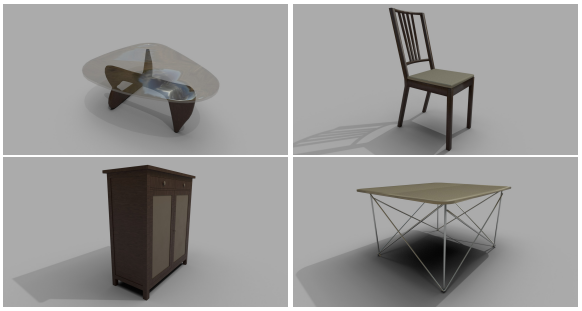


Figure 5: After material-aware segmentation of these representative shapes from our test set, shaders and textures corresponding to *wood*, *glass*, *fabric*, *plastic* and *metal* were applied to their faces.

Retrieval of 3D parts. Given a query 3D shape part from our test set, we can search for geometrically similar 3D parts in the training dataset. However, retrieval based on a geometric descriptor can return parts with inconsistent materials (see Figure 6(a)) whereas a designer might want to find geometrically similar parts with consistent materials (e.g. to replace the query part or its texture with a compatible database part) In Figure 6(b), we show retrieval results when we use both a geometric descriptor along with a simple material compatibility check. Our pipeline is used to obtain the material label for the untextured query part. Then, we retrieve geometrically similar parts from our training set whose crowdsourced material label agrees with the predicted one. For our prototype, we used the multi-view CNN of Su *et al*. [41] to compute geometric descriptors of parts.

Physical Simulation. Our material prediction pipeline allows us to perform simulation-based analysis of raw geometric shapes without any manual annotation of density, elasticity or other physical properties. This kind of visualization can be useful in interactive design applications to assist designers as they create models. In Figure 7, we show a prototype application which takes as input an unannotated polygon mesh, and simulates the effect of a downward force on it assuming the ground contact points are fixed. The material properties of the shape are predicted using a lookup table which maps material labels predicted by our method to representative density and elasticity values. We use the Vega toolkit [4] to select the force application region and deform the mesh under a downward impulse of 4800 N·s evenly distributed over this area. For this prototype, we ignore fracture effects and internal cavities, and assume the

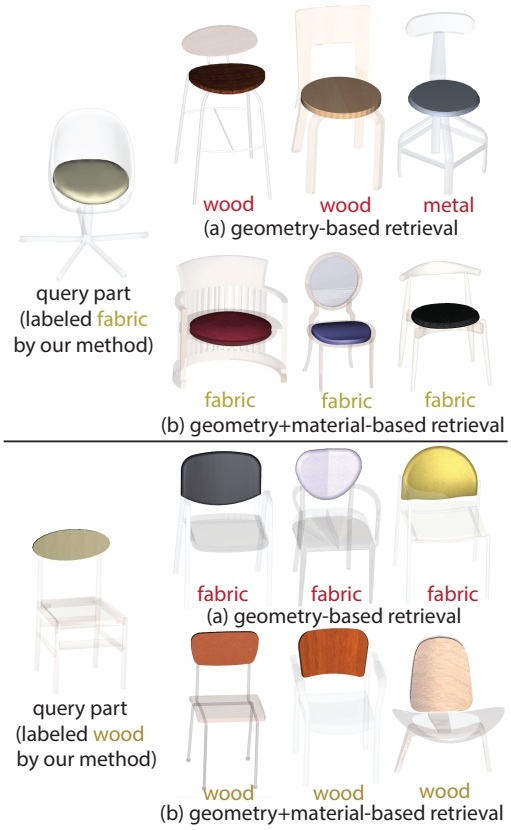


Figure 6: Given a query (untextured) part of a 3D shape and its material label predicted by our method, we can search for geometrically similar parts by considering (a) geometric descriptors alone, or (b) geometric descriptors together with material labels. Note that we discard any retrieved parts that are near duplicates of the query parts to promote more diversity in the retrieved results.

material is perfectly elastic. An implicit Newmark integrator performs finite element analysis over a voxelized (100^3) version of the shape. The renderings in Figure 7 show both the local surface strain (area distortion) as well as the induced deformation of shapes with different material compositions.

6. Conclusion

We presented a supervised learning pipeline to compute material-aware local descriptors for untextured 3D shapes, and developed the first crowdsourced dataset of 3D shapes with per-part physical material labels. Our learning method employs a projective convolution network in a Siamese setup, and material predictions inferred from this pipeline are smoothed by a symmetry-aware conditional random field. Our dataset uses a carefully designed crowdsourcing strategy to gather reasonably reliable labels for thousands of shapes, and an expert labeling procedure to generate ground truth labels for a smaller benchmark set used for evaluation. We demonstrated prototype applications leveraging the learned descriptors, and are placing the dataset in the public domain to drive future research in material-aware geometry processing.

(a) Wood: $\rho = 900$ kg/m^3, $E = 12.6$ GPa, $\nu = 0.45$

(b) Metal: $\rho = 8050$ kg/m^3, $E = 200$ GPa, $\nu = 0.3$

Figure 7: A downward impulse of 4800 N·s distributed over a chair seat (pink arrow) induces deformation (ignoring fracture and cavities). The left images show surface strain (area distortion), with blue = low and red = high. The corresponding deformation is visualized in the right images, with beige indicating the original shape and blue the overlaid deformed result. Wood, with much lower density (ρ) and Young's modulus (E), and higher Poisson's ratio (ν), is more strongly deformed than the stiffer metal (steel).

Our work is a first step and has several limitations. Our experiments have studied only a small set of materials, with tolerably discriminative geometric differences between their typical parts. Our projective architecture depends on rendered images and can hence process only visible parts of shapes. Also, our CRF-based smoothing is only a weak regularizer and cannot correct gross inaccuracies in the unary predictions. Addressing these limitations would be promising avenues for future work.

We believe that the joint analysis of physical materials and object geometry is an exciting and little-explored direction for shape analysis and design. Recent work on functional shape analysis [17] has been driven by priors based on physical simulation, mechanical compatibility or human interaction. Material-aware analysis presents a rich orthogonal direction that directly influences the function and fabricability of shapes. It would be interesting to combine annotations from 2D and 3D datasets to learn better material prediction models. It would also be interesting to reason about parametrized or fine-grained materials, such as different types of wood or metal, with varying physical properties. As a driving application, interactive modeling tools that provide continuous material-aware feedback on the shape being modeled could significantly aid real-world design tasks. Finally, there is significant scope for developing "material palettes" – probabilistic models of material co-use that take into account many intersectional design factors such as function, aesthetics, manufacturability and cost.

Acknowledgements. We acknowledge support from NSF (CHS-1617861, CHS-1422441, CHS-1617333). We thank Olga Vesselova for her input to the user study.

References

[1] 3D Warehouse, Trimble Inc. https://3dwarehouse.sketchup.com. Accessed: 2017. 3

[2] Herman Miller, Inc. https://www.hermanmiller.com. Accessed: 2017. 3

[3] IKEA. http://www.ikea.com. Accessed: 2017. 3

[4] J. Barbič, F. S. Sin, and D. Schroeder. Vega FEM Library. http://www.jernejbarbic.com/vega, 2012. 7

[5] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Opensurfaces: A richly annotated catalog of surface appearance. *ACM Trans. Graph.*, 32(4):111:1–111:17, 2013. 2

[6] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *Proc. CVPR*, 2015. 1, 2, 4, 7

[7] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum*, 34(5):13–23, 2015. 2, 4

[8] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 2, 4

[9] 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

[10] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. 22(3):223–232, 2003. 2

[11] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018. 2

[12] K. Chen, K. Xu, Y. Yu, T.-Y. Wang, and S.-M. Hu. Magic decorator: Automatic material suggestion for indoor digital scenes. *ACM Trans. Graph.*, 34(6):232:1–232:11, 2015. 2

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proc. CVPR*, pages 3606–3613, 2014. 2

[14] R. Girdhar, D. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[15] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018. 2

[16] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, pages 1735–1742. IEEE, 2006. 5

[17] R. Hu, M. Savva, and O. van Kaick. Functionality representations and applications for shape analysis. *Comp. Graph. For. (Eurographics State-of-The-Art Report)*, 2018. 8

[18] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer. Learning local shape descriptors with view-based convolutional neural networks. *ACM Trans. Graph.*, 37:6:1–6:14, 2018. 1, 2, 3, 4

[19] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel. Material Memex: Automatic material suggestions for 3D objects. *ACM Trans. Graph.*, 31(6):143:1–143:8, 2012. 2

[20] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3D shape segmentation with projective convolutional networks. In *Proc. CVPR*, 2017. 2, 3, 4, 6

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. 5

[22] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009. 5, 6

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 4

[24] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. ICML*, 2001. 4

[25] Y. Li, R. Bu, M. Sun, and B. Chen. PointCNN. *arXiv preprint arXiv:1801.07791*, 2018. 2

[26] O. Litany, T. Remez, E. Rodola, A. M. Bronstein, and M. M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proc. ICCV*, volume 2, page 8, 2017. 2

[27] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Proc. CVPR*, pages 239–246, 2010. 2

[28] Z. Lun, E. Kalogerakis, and A. Sheffer. Elements of style: Learning perceptual shape style similarity. *ACM Trans. Graph.*, 34(4), 2015. 5

[29] Z. Lun, E. Kalogerakis, R. Wang, and A. Sheffer. Functionality preserving shape style transfer. *ACM Trans. Graph.*, 35(6), 2016. 5

[30] H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V. G. Kim, and Y. Lipman. Convolutional neural networks on surfaces via seamless toric covers. *Trans. Graph.*, 36(4), 2017. 2

[31] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proc. ICCV workshops*, pages 37–45, 2015. 2, 4, 5

[32] N. J. Mitra, M. Wand, H. Zhang, D. Cohen-Or, and M. Bokeloh. Structure-aware shape processing. *Eurographics State of the Art Reports*, 2013. 1

[33] S. Muralikrishnan, V. G. Kim, and S. Chaudhuri. Tags2Parts: Discovering semantic regions from shape tags. In *Proc. CVPR*, 2018. 2

[34] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, 2017. 1, 2, 4

[35] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[36] G. Riegler, A. O. Ulusoy, and A. Geiger. OctNet: Learning deep 3D representations at high resolution. In *CVPR*, 2017. 2

[37] M. Savva, A. X. Chang, and P. Hanrahan. Semantically-enriched 3D models for common-sense knowledge. *CVPR Workshop on Functionality, Physics, Intentionality and Causality*, 2015. 2

[38] L. Sharan, C. Liu, R. Rosenholtz, and E. Adelson. Recognizing materials using perceptually inspired features. *IJCV*, 103:348–371, 2013. 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[40] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proc. CVPR*, 2018. 2, 4

[41] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proc. ICCV*, 2015. 2, 4, 7

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. In *Proc. CVPR*, 2015. 4

[43] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *Trans. Graph.*, 36(4), 2017. 2

[44] T. Y. Wang, H. Su, Q. Huang, J. Huang, L. Guibas, and N. J. Mitra. Unsupervised texture transfer from images to model collections. *ACM Trans. Graph.*, 35(6):177:1–177:13, 2016. 2

[45] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic Graph CNN for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 2

[46] T. Weyrich, J. Lawrence, H. P. A. Lensch, S. Rusinkiewicz, and T. Zickler. Principles of appearance acquisition and representation. *Found. Trends. Comput. Graph. Vis.*, 4(2), 2009. 2

[47] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proc. CVPR*, 2015. 2, 4

[48] Y.-L. Yang, J. Wang, and N. J. Mitra. Reforming shapes for material-aware fabrication. In *Computer Graphics Forum*, volume 34, pages 53–64. Wiley Online Library, 2015. 1, 2

[49] L. Yi, H. Su, X. Guo, and L. Guibas. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. 2017. 2

[50] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proc. CVPR*, 2017. 1, 4