

# 3D Shape Attributes

David F. Fouhey, Abhinav Gupta  
Robotics Institute  
Carnegie Mellon University

Andrew Zisserman  
Dept. of Engineering Science  
University of Oxford

## Abstract

*In this paper we investigate 3D attributes as a means to understand the shape of an object in a single image. To this end, we make a number of contributions: (i) we introduce and define a set of 3D Shape attributes, including planarity, symmetry and occupied space; (ii) we show that such properties can be successfully inferred from a single image using a Convolutional Neural Network (CNN); (iii) we introduce a 143K image dataset of sculptures with 2197 works over 242 artists for training and evaluating the CNN; (iv) we show that the 3D attributes trained on this dataset generalize to images of other (non-sculpture) object classes; and furthermore (v) we show that the CNN also provides a shape embedding that can be used to match previously unseen sculptures largely independent of viewpoint.*

## 1. Introduction

Suppose you saw the sculpture in Fig. 1(a) on vacation and wanted to call your friend and tell her what you saw. How might you describe it so she would know that you were referring to the one in Fig. 1(a) and not the one in (b)? What you would not do is describe the sculpture pixel by pixel. Instead you would probably give a high level description in terms of overall shape, holes, curvature, sharpness/smoothness, etc. This is in stark contrast to most contemporary 3D understanding algorithms that in the first instance produce a metric map, i.e. a prediction of a local metric property like depth or surface normals at each pixel.

The objective of this paper is to infer such generic 3D shape properties directly from appearance. We term these properties **3D shape attributes** and introduce a variety of specific examples, for instance planarity, thinness, point-contact, to concretely explore this concept. Although such attributes can be derived from an estimated depthmap in principle, in practice (as we will show with baselines) view dependence, insufficient resolution, errors and ambiguities in the reconstruction render this indirect approach inferior.

As with classical object attributes and relative attributes [12, 13, 35], 3D attributes offer a means of describing 3D object shape when confronted with something

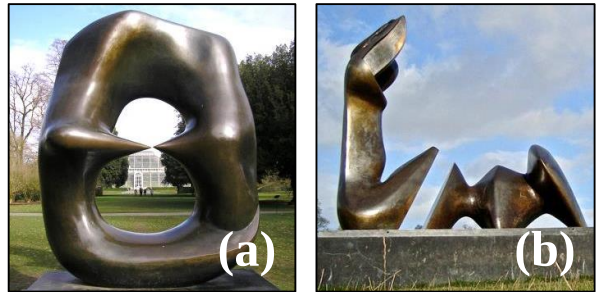


Figure 1. How would you describe the shape of (a) and contrast it with (b)? Probably not by quantifying the depth at each pixel, but instead characterizing the overall 3D shape: in (a) the object has a hole, does not have planar regions and is mostly smooth, has unequal aspect ratio, and touches the ground once. In contrast (b) has no hole and multiple parts that touch the ground multiple times. This paper proposes to infer shape in these terms.

entirely new – the *open world problem*. This is in contrast to a long line of work which is able to say something about 3D shape, or indeed recover it, from single images given a specific object class, e.g. faces [7], semantic category [23] or cuboidal room structure [18]. While there has been success in determining *how* to apply these constraints, the problem of *which* constraints to apply is much less explored, especially in the case of completely new objects. Used inappropriately, scene understanding methods produce either unconstrained results [10, 14] in which walls that should be flat bend arbitrarily or planar interpretations [15, 33] in which non-planar objects like lamps are flat. Shape attributes can act as a generic way of representing top-down properties for 3D understanding, sharing with classical attributes the advantage of both learning and application across multiple object classes.

There are two natural questions to ask: what 3D attributes should be inferred, and how to infer them? In Section 3, we introduce our attribute vocabulary, which draws inspiration from and revisits past work in both the computer and the human vision literature. We return to these ideas with modern computer vision tools. In particular, as we describe in Section 5, we use Convolutional Neural Networks (CNNs) to model this mapping and learn a model over all

of the attributes as well as a shape embedding.

The next important question is: what data to use to investigate these properties? We use photos of modern sculptures from Flickr, and describe a procedure for gathering a large and diverse dataset in Section 4. This data has many desirable properties: it has much greater variety in terms of shape compared to common-place objects; it is real and in the wild, so has all the challenging artifacts such as severe lighting and varying texture that may be missing in synthetic data. Additionally, the dataset is automatically organized into: *artists*, which lets us define a train/test split to generalize over artists; *works* (of art) irrespective of material or location, which lets us concentrate on shape, and *viewpoint clusters*, which lets us recognize sculptures from multiple views and aspects.

The experiments in Section 6 clearly show that we are indeed able to infer 3D attributes. However, we also ask the question of whether we are actually learning 3D properties, or instead a proxy property, such as the identity of the artist, which in turn enables these properties to be inferred. We have designed the experiments both to avoid this possibility and to probe this issue, and discuss this there.

## 2. Related Work

How do 2D images convey 3D properties of objects? This is one of the central questions in any discipline involving perception – from visual psychophysics to computer vision to art. Our approach draws on each of these fields, for instance in picking the particular attributes we investigate or probing our learned model.

One motivation for our investigation of shape attributes is a long history of work in the human perception community that aims to go beyond metric properties and address holistic shape in a view-independent way. Amongst many others, Koenderink and van Doorn [26] argued for a set of shape classes based on the *sign* of the principal curvatures and also that shape perception was not metric [27, 28], and Biederman [6] advocated shape classes based on non-accidental *qualitative* contour properties. We are also inspired by work on trying to use mental rotation [42, 44] to probe how humans represent shape; here, we use it to probe whether our models have learned something sensible.

A great deal of research in early computer vision sought to extract local or qualitative cues to shape, for instance from apparent contours [24], self-shadows and specularities [25, 47]. Recent computer vision approaches to this problem, however, have increasingly reduced 3D understanding to the task of inferring a viewpoint-dependent 3D depth or normal at each pixel [5, 10, 14]. These predictions are useful for many tasks but do not tell the whole story, as we argued in the introduction. This work aims to help fill this gap by revisiting these non-metric qualitative questions. Some exceptions to this trend include the qualitative labels explored in [16, 20] like porous, but these initial efforts had

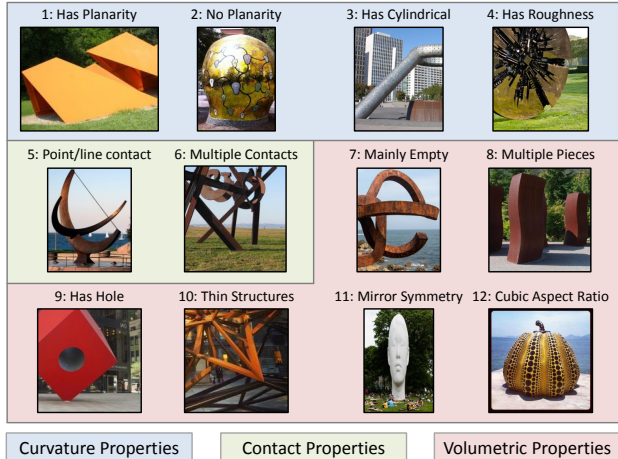


Figure 2. The 3D shape attributes investigated in this paper, and an illustration of each from our training set. Additional sample annotations are shown in Fig. 5.

limited scope in terms of data variety, vocabulary size, and quantity of images.

Our focus is 3D shape understanding, but we pose our investigation into these properties in the language of attributes [12, 13, 30, 31, 35] to emphasize their key properties of communicability and open-world generalization. The vast majority of attributes, however, have been semantic and there has never been, to our knowledge, a systematic attempt to connect attributes with 3D understanding or to study them with data specialized for 3D understanding. Our work is most related to the handful of coarse 3D properties in [12]. In addition to having a larger number of attributes and data designed for 3D understanding, our attributes are largely unaffected by viewpoint change.

## 3. 3D Attribute Vocabulary

Which 3D shape attributes should we model? We choose 12 attributes based on questions about three properties of historical interest to the vision community – curvature (how does the surface curve locally and globally?), ground contact (how does the shape touch the ground?), and volumetric occupancy (how does the shape take up space?).

Fig. 2 illustrates the 12 attributes, and sample annotations are shown in Fig. 5, with more in the supplementary material. We now briefly describe the attributes in terms of curvature, contact, and volumetric occupancy.

**Curvature:** We take inspiration from a line of work on shape categorization via curvature led by Koenderink and van Doorn (e.g., [26]). Most sculptures have a mix of convex and concave regions, so we analyze where curvature is zero in at least one direction and look for (1) piecewise planar sculptures, and (2) sculptures with *no* planar regions (many have a mix of planar/non-planar), (3) sculptures where one principal curvature vanishes (e.g., cylindrical), and (4) rough sculptures where locally the surface

changes rapidly.

*Contact:* Contact reasoning plays a strong role in understanding (e.g., [16, 19, 21]); we characterize ground contact via (5) point/line contact-vs-solid contact, and (6) whether it contacts the ground in multiple places.

*Volumetric:* Reasoning about free-space has long been a goal of 3D understanding [19, 32, 38]. We ask (7) the fraction of occupied space; (8) whether the sculpture has multiple pieces; (9) whether there are holes; (10) whether it has thin structures; (11) whether it has mirror symmetry; and (12) whether it has a cubic aspect ratio. These are, of course, not a complete set: we do not model, for example, enclosure properties or differentiate a hole from a mesh.

Note, of the 12 attributes, 10 are relatively unaffected by a geometric affine transformation of the image (or 3D space) – only the cubic aspect ratio and mirror symmetry attributes are measuring a global metric property.

## 4. Gathering a Dataset of 3D Shapes

In order to investigate these 3D attributes, we need a dataset of sculptures that has a diversity of shape so that different subsets of attributes apply. We also require images spanning many viewpoints of each sculpture in order to investigate the estimation of attributes against viewpoint.

Since artists often produce work in a similar style (Calder’s sculptures are mostly piecewise planar, Brancusi’s egg-shaped) we therefore need a variety of artists and multiple works/images of each. Previous sculpture datasets [2, 3] are not suitable for this task as they only contain a small number of artists and viewpoints.

Thus we gather a new dataset from Flickr. We adopt a five stage process to semi-automatically do this: (i) obtain a vocabulary of artists and works (for which many images will be available); (ii) cluster the works by viewpoint; (iii) clean up mistakes; (iv) query expand for more examples from Google images; and (v) label attributes. Note, organization by artist is not strictly necessary. However, artists are used subsequently to split the works into train and test datasets: as noted above, due to an artists’ style, shape attributes frequently correlate with an artist; Consequently artists in the train and test splits should be disjoint to avoid an overly optimistic generalization performance. The statistics for these stages are given in Tab. 1.

### 4.1. Generating a vocabulary of artists and works

Our goal is to generate a vocabulary of artists and works that is as broad as possible. We begin by producing a list of artists, combining manually generated lists with automatic ones, and then expand each artist to a list of their works.

The manual list consists of the artists exhibited at six sculpture parks picked from online top-10 lists, as well as those appearing in Wikipedia’s article on Modern Sculpture. An automatic list is generated from metadata from the 20 largest sculpture groups on Flickr: we analyze image titles

Table 1. Data statistics at each stage and the trainval/test splits.

| Stage         | Images   | Artists | Works    | View. Clusters |
|---------------|----------|---------|----------|----------------|
| Initial       | 352K     | 258     | 3412     | –              |
| View Clust.   | 213K     | 246     | 2277     | 16K            |
| Cleaned       | 97K      | 242     | 2197     | 9K             |
| Query Exp.    | 143K     | 242     | 2197     | 9K             |
| Trainval/Test | 109K/35K | 181/61  | 1655/532 | 7.2K/2.1K      |

for text indicating that a work is being ascribed to an artist, and take frequent bigrams and trigrams. The two lists are manually filtered to remove misspellings, painters and architects, a handful of mistakes, and artists with fewer than 250 results on Flickr. This yields 258 artists (95 from the manual list, and 163 from the automatic).

We now find a list of potential works for each artist using both Wikipedia and text analysis on Flickr. We query the sculptor’s page on Wikipedia, possibly manually disambiguating, and propose any italicized text in the main body of the article as a possible work. We also query Flickr for the artists’ works (e.g., Tony Smith Sculpture), and do n-gram analysis in titles and descriptions in front of phrases indicating attribution to the sculpture (e.g., “by Tony Smith”). In both cases, as in [37], stop-word lists were effective in filtering out noise. While Wikipedia has high precision, its recall is moderate at best and zero for most artists. Thus querying Flickr is crucial for obtaining high quality data. Finally, images are downloaded from Flickr for each work of each artist.

### 4.2. Building viewpoint clusters

Images from each work are partitioned into *viewpoint clusters*. These clusters are image sets that, for example, capture a different visual aspect of the work (e.g. from the front or side) or are acquired from a particular distance or scale (e.g. a close up). Fig. 3 shows example viewpoint clusters for several works.

There are two principal reasons for obtaining viewpoint clusters: (i) it enables recognition of a work from different viewpoints to be evaluated; and (ii) it makes label annotation more efficient as attributes are in general valid for all images of a cluster. Note, it might be thought that attributes could be labelled at the work level, but this is not always the case. For example, the hole in a Henry Moore sculpture or the ground contact of an Alexander Calder sculpture may not be visible in some viewpoint clusters, so those clusters will be labelled differently from the rest (i.e. no hole for the former, and unknown for the latter).

Clustering proceeds in a standard manner by defining a similarity matrix between image pairs, and using spectral clustering over the matrix. The pairwise similarity measure takes into account: (i) the number of correspondences (that there are a threshold number); (ii) the stability of these correspondences (using cyclic consistency as in [46]); and



Figure 3. The dataset consists of 143K images of sculptures that were gathered from Flickr and Google images. A representative sample is shown on the left. Note the great variety in shape, material, and style. More samples appear in the supplementary material. Our data has structure in terms of artist, work, and viewpoint cluster (shown numbered on the right). Each is important for investigating 3D shape attributes.

(iii) the viewpoint change (the rotation and aspect ratio change obtained from an affine transformation between the images). Computing correspondences requires some care though since sculptures often do not have texture (and thus SIFT like detections cannot be used). We follow [1] and first obtain a local boundary descriptor for the sculpture (by foreground-background segmentation and MCG [4] edges for the boundaries), and then obtain geometrically consistent correspondences using an affine fundamental matrix. Finally, a loose affine transformation is computed from the correspondences (loose because the sculpture may be non-planar, hence the earlier use of a fundamental matrix).

In general, this procedure produces clusters with high purity. The main failure is when an artist has several visually similar works (e.g. busts) that are confused in the meta-data used to download them. We also experimented with using GPS, but found the tags to be too coarse and noisy to define satisfactory viewpoint clusters.

### 4.3. Data Cleanup

The above processes are mainly automatic and consequently make some mistakes. A number of manual and semi-automatic post-processing steps are therefore applied to address the main failings. Note, we can quickly manipulate the dataset via viewpoint clusters as opposed to handling each and every image individually.

*Cluster filtering:* Each cluster is checked manually using three sample images to reject clearly impure clusters.

*Regrouping:* Some of the automatically generated works are ambiguous due to noisy meta-data: for instance “Reclining Figure” describes a number of Henry Moore sculptures. After clustering, these are reassigned to the correct works.

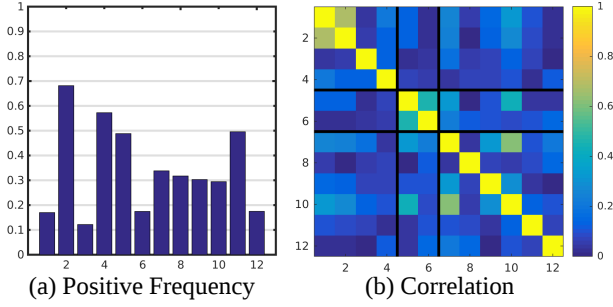
*Outlier image removal:* A 1-vs-rest SVM is trained for each work, using fc7 activations of a CNN [29] pretrained on ImageNet [39]. Each work’s images are sorted according to the SVM score, and the bottom images ( $\approx 10K$  across all works) flagged for verification.

### 4.4. Expansion Via Search Engines

Finally, we augment the dataset by querying Google. We perform queries with the artist and work name. Using the same CNN activation + SVM technique from the outlier removal stage, we re-sort the query results and add the top images after verification. This yields  $\approx 45K$  more images.

### 4.5. Attribute Labeling

The final step is to label the images with attributes. Here, the viewpoint clusters are crucial, as they enable the labeling of multiple images at once. Each viewpoint cluster is labeled with each attribute, or can be labeled as N/A in case the attribute cannot be determined from the image (e.g.,



Key: (1) Planar (2) No Planar (3) Cylindrical (4) Rough (5) Point Contact (6) Multiple Contact (7) Empty (8) Multiple Pieces (9) Holes (10) Thin (11) Symmetric (12) Cubic

Figure 4. (a) Frequency of each attribute (i.e., # positives / # labeled); (b) Correlation between attributes.

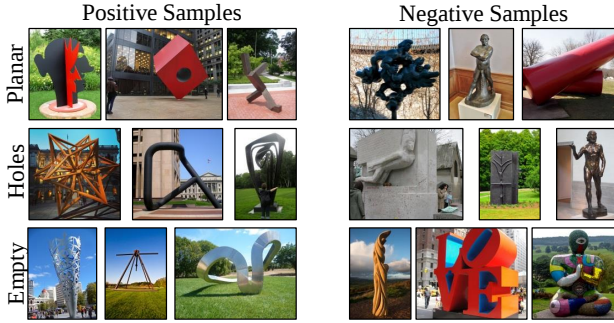


Figure 5. Sample positive and negative annotations from the dataset for the planar, has-holes, and empty attributes. More examples are included in the supplementary material.

contact properties for a hanging sculpture). One difficulty is determining a threshold: few sculptures are only planar and no sculpture is fully empty. We assume an attribute is satisfied if it is true for a substantial fraction of the sculpture, typically 80%. To give a sense of attribute frequency, we show the fraction of positives in Fig. 4(a).

The dataset is also diverse in terms of combinations of attributes and inter-attribute correlation. There are  $2^{12} = 4096$  possible combinations, of which 393 occur in our data. Most attributes are uncorrelated according to the correlation coefficient  $\phi$ , as seen in Fig. 4(b): mean correlation is  $\phi = 0.20$  and 72% of pairs have  $\phi < 0.2$ . The two strong correlations ( $\phi > 0.5$ ) are, unsurprisingly, (1) planarity and no planarity; and (2) emptiness and thinness.

## 5. Approach

We now describe the CNN architecture and loss functions that we use to learn the attribute predictors and shape embedding. We cast this as multi-task training and optimize directly for both. Specifically, the network is trained using a loss function over all attributes as well as an embedding loss that encourages instances of the same shape to have the same representation. The former lets us model the attributes that are currently labeled. The latter forces the network to learn a representation that can distinguish sculptures, im-

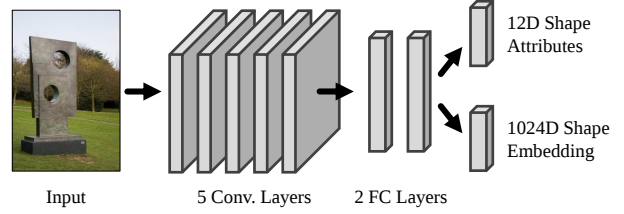


Figure 6. The multi-task network architecture, based on VGG-M. After shared layers, the network branches into layers specialized for attribute classification and shape embedding.

PLICITLY modeling aspects of shape not currently labeled.

**Network Architecture:** We adapt the VGG-M architecture proposed in [9]. We depict the overall architecture in Fig. 6: all layers are shared through the last fully connected layer, fc7. After fc7, the model splits into two branches, one for attributes, the other for embedding. The first is an affine map to 12D followed by independent sigmoids, producing 12 separate probabilities, one per attribute. The second projects fc7 to a 1024D embedding which is then normalized to unit norm.

We directly optimize the network for both outputs, which allows us to obtain strong performance on both tasks. The first loss models all the attributes with a cross-entropy loss summed over the valid attributes. Suppose there are  $N$  samples and  $L$  attributes, each of which can be 1 or 0 as well as  $\emptyset$  to indicate that the attribute is not labeled; the loss is

$$L(Y, P) = \sum_{i=1}^N \sum_{\substack{l=1 \\ Y_{i,l} \neq \emptyset}}^L Y_{i,l} \log(P_{i,l}) + (1 - Y_{i,l}) \log(1 - P_{i,l}), \quad (1)$$

for image  $i$  and label  $l$ , where we denote the label matrix as  $Y_{i,l} \in \{0, 1, \emptyset\}^{N,L}$  and the predicted probabilities as  $P_{i,l} \in [0, 1]^{N,L}$ . The second loss is an embedding loss over triplets as in [40, 41, 45]. Each triplet  $i$  consists of an anchor view of one object  $x_i^a$ , another view of the same object  $x_i^p$ , as well as a view of a different object  $x_i^n$ . The loss aims to ensure that two images of the same object are closer in feature space compared to another object by a margin:

$$\sum_{i=1} \max(D(x_i^a, x_i^p) - D(x_i^a, x_i^n) + \alpha, 0) \quad (2)$$

where  $D(\cdot, \cdot)$  is squared Euclidean distance. We generate triplets in a mini-batch and use soft-margin violators [40].

We see a number of advantages to multi-task learning. It yields a network that can both name attributes it knows about and model the 3D shape space implicitly. Additionally, we found it to improve learning stability, especially compared to individually modeling each attribute.

**Configurations:** We explore two configurations to validate that we are really learning about 3D shape. Unless otherwise specified, we use the system described above, *Full*. However, to probe what is being learned in one experiment, we also learn a network that only optimizes the attribute Loss (1), which we refer to as *Attribute-Only*.



Figure 7. Predictions for all attributes on test images. The system has never seen these sculptures or ones by the artists who made them, but generalizes successfully.

**Implementation Details:** *Optimization:* We use a standard stochastic gradient descent plus momentum approach with a batch size of 128. *Initialization:* We initialize the network using the model from [9] which was pre-trained on image classification [39]. *Parameters:* We use a learning rate of  $10^{-4}$  for the pre-trained layers, and  $10^{-4}$  and  $10^{-3}$  for classification and embedding layers respectively. *Augmentation:* At training time, we use random crops, flips, and color jitter. At test time, we sum-pool over multiple scales, crops and flips as in [9].

## 6. Experiments

We describe a set of experiments to investigate both the performance of the learnt 3D shape attribute classifiers, and what has been learnt. We aim to answer two basic questions: (1) how well can we predict 3D shape attributes from a single image? and (2) are we actually predicting 3D properties or a proxy property that correlates with attributes in an image? To address (1) we evaluate the performance on the Sculpture Images Test set, and also compare to alternative approaches that first predict a metric 3D representation and then derive 3D attributes from that (Sec. 6.1). We probe (2) in two ways. First, by evaluating the learnt representation for a different task – determining if two images from different viewpoints are of the same object or not (Sec. 6.2); and, second, by evaluating how well the 3D shape attributes trained on the Sculpture images generalize to non-sculpture data, in particular to predicting shape attributes on PASCAL VOC categories (Sec. 6.3).

### 6.1. Attribute Prediction

We first evaluate how well 3D shape attributes can be estimated from images. Here, we report results for our full network. Since our dataset is large enough, the attribute-only network does similarly. We compare the approach pro-

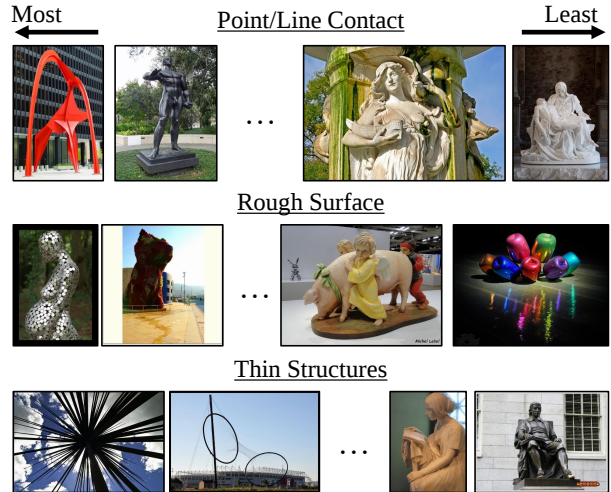


Figure 8. Test images sampled at the top, 95<sup>th</sup>, 5<sup>th</sup> percentiles and lowest percentile with respect to three attributes.

posed in this paper (which directly infers holistic attributes) to a number of baselines that are depth orientated, and start by computing a metric depth at every pixel.

**Baselines:** The baselines start by estimating a metric 3D map, and then attributes are extracted from this map. We use two recent methods for estimating depth from single images with code available: a CNN-based depth estimation technique [10] and an intrinsic images technique [5]. Since [5] expects a mask, we use the segmentation used for collecting the dataset (in Sec. 4.2). One question is: how do we convert these depthmaps into our attributes? Hand-designing a method is likely to produce poor results. We take a data-driven approach and treat it as a classification problem. We use two approaches that have produced strong performance in the past. The first is a linear SVM on top of kernel depth descriptors [8], which convert the depthmap into a high-dimensional vector incorporating depth configurations and image location. The second is the HHA scheme [17], which converts the depthmap into a representation amenable for fine-tuning a CNN; in this case, we learn the attribute network described in section 5.

**Evaluation Criteria:** Each method produces a prediction scoring how much the image has the attribute. We characterize the predictive ability of these scores with a receiver operator characteristic (ROC) over the Sculpture images test set. This enables comparison *across attributes* since the ROC is unaffected by class frequency. We summarize scores with the area under the curve.

**Results:** Fig. 7 shows predictions of all of the attributes on a few sculptures. To help visualize what has been learned, we show automatically sampled results in Fig. 8, sorted by the predicted presence of attributes. Additional results are given in the supplementary material.

We report quantitative results in Table 2. On an absolute basis, certain attributes, such as planarity and emptiness, are

Table 2. Area under the ROC curve. Higher is better. Our approach outperforms the baselines by a large margin and achieves strong performance on an absolute basis.

|             | Curvature                                                                         |                                                                                   |                                                                                   |                                                                                   | Contact                                                                           |                                                                                   | Occupancy                                                                         |                                                                                   |                                                                                     |                                                                                     |                                                                                     |                                                                                     |             |
|-------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------|
| Method      |  |  |  |  |  |  |  |  |  |  |  |  |             |
| [5] + [8]   | Plan                                                                              | ¬Plan                                                                             | Cyl                                                                               | Rough                                                                             | P/L                                                                               | Mult                                                                              | Emp                                                                               | Mult                                                                              | Hole                                                                                | Thin                                                                                | Sym                                                                                 | Cubic                                                                               | Mean        |
| [5] + [8]   | 64.1                                                                              | 63.4                                                                              | 51.2                                                                              | 61.3                                                                              | 61.1                                                                              | 61.6                                                                              | 66.5                                                                              | 52.8                                                                              | 56.0                                                                                | 63.5                                                                                | 56.2                                                                                | 55.7                                                                                | 59.4        |
| [10] + [8]  | 64.6                                                                              | 61.0                                                                              | 50.6                                                                              | 60.6                                                                              | 57.5                                                                              | 60.9                                                                              | 65.2                                                                              | 55.7                                                                              | 52.4                                                                                | 65.7                                                                                | 57.2                                                                                | 51.2                                                                                | 58.5        |
| [5] + [17]  | 70.0                                                                              | 64.4                                                                              | 53.1                                                                              | 63.9                                                                              | 63.6                                                                              | 64.8                                                                              | 73.7                                                                              | 56.4                                                                              | 54.1                                                                                | 69.7                                                                                | 60.2                                                                                | 56.2                                                                                | 62.5        |
| [10] + [17] | 67.5                                                                              | 61.8                                                                              | 51.9                                                                              | 64.8                                                                              | 58.5                                                                              | 64.8                                                                              | 71.5                                                                              | 57.8                                                                              | 52.4                                                                                | 67.7                                                                                | 59.4                                                                                | 56.1                                                                                | 61.2        |
| Proposed    | <b>82.8</b>                                                                       | <b>77.2</b>                                                                       | <b>56.9</b>                                                                       | <b>76.0</b>                                                                       | <b>74.4</b>                                                                       | <b>76.4</b>                                                                       | <b>87.0</b>                                                                       | <b>60.4</b>                                                                       | <b>69.3</b>                                                                         | <b>85.8</b>                                                                         | <b>60.8</b>                                                                         | <b>60.3</b>                                                                         | <b>72.3</b> |

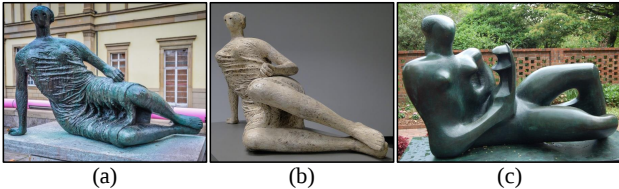


Figure 9. In mental rotation, the goal is to verify that (a) and (b) correspond and (a) and (c) do not. Roughness is a useful cue here.

easier than others to predict, as seen by their average performance; the harder ones include ones based on symmetry and aspect ratio, which may require a global comparison across the image, as opposed to aggregation of local judgments.

In relative terms, our approach outperforms the baselines, with especially large gains on planarity, emptiness, and thinness. Note that reconstructing thin structures is challenging even with stereo pairs; an approach based on depth-prediction is likely to fail at reconstruction, and thus on attribute prediction. Instead, our system directly recognizes that the object is thin (e.g., Fig. 7 bottom). Fig. 8 shows that frequently, the instances that least have an attribute are the negation of the attribute: for example, even though many other sculptures are not rough, the least rough objects are especially smooth.

The system’s mistakes primarily occur on images where it is uncertain: sorting the images by attribute prediction and re-evaluating on the top and bottom 25% of the images yields a substantial increase to 77.9% mean AUC; using the top and bottom 10% yields an increase to 82.6%.

Throughout, we fix our base representation to VGG-M [9]. Switching to VGG-16 [43] gives an additional boost: the mean increases from 72.3 to 74.4 and 1/3 of the attributes are predicted with AUCs of 80% or more.

## 6.2. Mental Rotation

If we have learned about 3D shape, our learnt representation ought to encode or *embed* 3D shape. But how do we characterize this embedding systematically? To answer this, we turn to the task of mental rotation [42, 44] which is the following: given two images, can we tell if they are different views of the same object or instead views of different

objects? This is a classification task on the two presented images: for instance, in Fig. 9, the task is to tell that (a) and (b) correspond, and that (a) and (c) do not.

Note, the design of the dataset has tried to ensure that sculpture shape is not correlated with location by ensuring that images of a particular work come from different locations (since multiple instances of a work are produced) and different materials (e.g., bronze and stone in Fig. 9).

We report four representations: (i) the 1024D embedding produced by our full network; (ii) the 4096D fc7 layer of the full network; (iii) the 4096D fc7 layer of the attribute-only network; (iv) the attribute probabilities themselves from the full network. If our attribute network is using actual 3D properties, then the attribute network’s activations ought to work well for the mental rotation task even though it was never trained for it explicitly. Additionally, the attributes themselves ought to perform well.

**Baselines:** We compare our approach to (i) the pretrained FC7 from the initialization of the network and to (ii) IFV [36] over the BOB descriptor [2] that was used to create the dataset and dense SIFT [34]. The pre-trained FC7 characterizes what has been learned; the IFV representations help characterize the effectiveness of the attribute predictions on their own. We use the cosine distance throughout.

**Evaluation Criteria:** We adopt the evaluation protocol of [22] which has gained wide acceptance in face verification: given two images, we use their distance as a prediction of whether they are images of the same object or not. Performance is measured by AUROC, evaluated over 100 million of the pairs, of which 0.9% are positives. Unlike [22], positives in the same viewpoint cluster are ignored: these are too easy decisions.

We further hone in on difficult examples by automatically finding and removing easy positives which can be identified with a bare minimum image representation. Specifically, we remove positive pairs with below-median distance in a 512-vocabulary bag-of-words over SIFT representation. This yields a more challenging dataset with 0.3% positives. As mentioned in Sec. 4 artists often produce work of a similar style, and the most challenging examples are often pairs of images from the same artist (which may or may not be of the same work). We call the standard setting *Easy*

Table 3. AUC for the mental-rotation task. Both variants of our approach substantially out-perform the baselines.

|      | Full Network |      | Attr. Only | Pretrained | IFV  |      |
|------|--------------|------|------------|------------|------|------|
|      | Emb.         | FC7  | Attr       | FC7        | [34] | [2]  |
| All  | <b>92.3</b>  | 90.7 | 81.9       | 89.8       | 88.9 | 78.0 |
| Hard | <b>86.9</b>  | 84.1 | 76.4       | 82.5       | 80.0 | 57.3 |

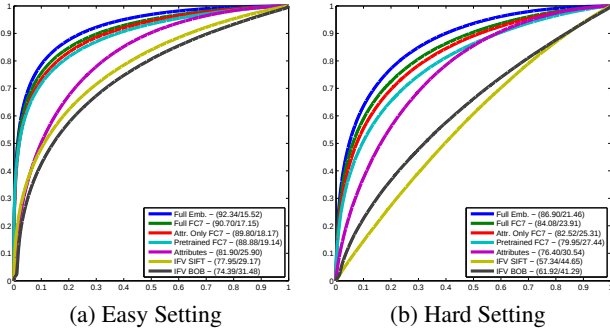


Figure 10. Mental rotation ROCs for easy and hard settings. In the legend, we report the AUC and EER for each method.

and the filtered setting with only hard positives *Hard*.

**Results:** Table 3 and Fig. 10 show results for both settings. By themselves, the 12D attributes produce strong performance, 3-4% better than IFV representations. The attribute-only network improves over pretraining (by 0.9% in easy, 2.5% in hard), suggesting that it has learned the shape properties needed for the task. The full system does best and substantially better than any baseline (by 3.4% in easy, 6.9% in hard). Relative performance compared to the initialization consistently *improves* for both the full system and the attribute-only system when going from Easy to Hard settings, providing further evidence that the system is indeed modeling 3D properties.

### 6.3. Object Characterization

Our evaluation has so far focused on sculptures, and one concern is that what we learn may not generalize to more everyday objects like trains or cats. We thus investigate our model’s beliefs about these objects by analyzing its activations on the PASCAL VOC dataset [11]. We feed the windows of the trainval set of VOC-2010 to our shape attribute model, and obtain a prediction of the probability of each attribute. We probe the representation by sorting class members by their activations (i.e., “which trains are planar?”) and sorting the classes by their mean activations.

**Per-image results:** The system forms sensible beliefs about the PASCAL objects, as we show in Fig. 11. Looking at intra-class activations, cats lying down are predicted to have single, non-point contact as compared to ones standing up; trains are generally planar, except for older cylindrical steam engines. Similarly, the non-planar dining tables are the result of occlusion by non-planar objects.

**Per-category results:** The system performs well at a

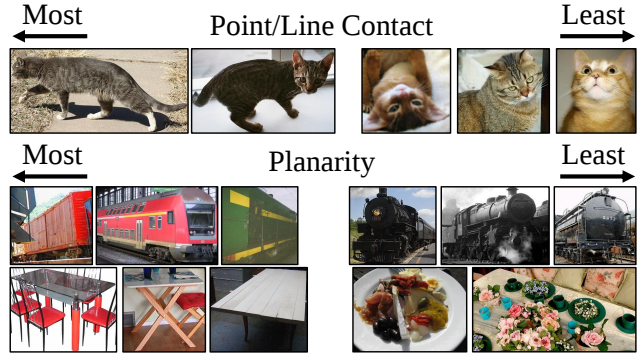


Figure 11. The top activations on PASCAL objects for Planarity and Point/Line Contact.

category-level as well. Note that averaging over windows characterizes how objects *appear* in PASCAL VOC, not how they are prototypically imagined: e.g., as seen in Fig. 11, the cats and dogs of PASCAL are frequently lying down or truncated. The top 3 categories by planarity are bus, TV Monitor, train; and the bottom 3 are cow, horse, sheep. For point/line contact: bus, aeroplane, car are at the top and cat, bottle, sofa are at the bottom. Finally, sheep, bird, and potted plant are the roughest categories in PASCAL and car, bus, and aeroplane the smoothest.

**Discriminating between classes:** It ought to be possible to distinguish between the VOC categories based on their 3D properties, and thus we verify that the predicted 3D shape attributes carry class-discriminative information. We represent each window with its 12 attribute probabilities and train a random forest classifier for two outcomes in a 10-fold cross-validation setting: a 20-way multiclass model and a one-vs-rest. The multiclass model achieves an accuracy of 65%, substantially above chance. The one-vs-rest model achieves an average AUROC of 89%, with vehicles performing best.

## 7. Summary and extensions

We have shown that 3D attributes can be inferred directly from images at quite high quality. These attributes open a number of possibilities of applications and extensions. One immediate application is to use this system to complement metric reconstruction: shape attributes can serve as a top-down cue for driving reconstruction that works even on unknown objects. Another area of investigation is explicitly formulating our problem in terms of relative attributes: many of our attributes (e.g., planarity) are better modeled in relative terms. Finally, we plan to investigate which cues (e.g., texture, edges) are being used to infer these attributes.

**Acknowledgments:** Financial support for this work was provided by the EPSRC Programme Grant Seebibyte EP/M013774/1, ONR MURI N000141612007, and a NDSEG fellowship to DF. The authors thank Omkar Parkhi and Xiaolong Wang for a number of helpful conversations, and NVIDIA for GPU donations.

## References

- [1] R. Arandjelović and A. Zisserman. Efficient image retrieval for 3D structures. In *BMVC*, 2010. 4
- [2] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *ICCV*, 2011. 3, 7, 8
- [3] R. Arandjelović and A. Zisserman. Name that sculpture. In *ACM ICMR*, 2012. 3
- [4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition*, 2014. 4
- [5] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015. 2, 6, 7
- [6] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987. 2
- [7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999. 1
- [8] L. Bo, X. Ren, and D. Fox. Depth Kernel Descriptors for Object Recognition. In *IROS*, 2011. 6, 7
- [9] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 5, 6, 7
- [10] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 1, 2, 6, 7
- [11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 8
- [12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2
- [13] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 1, 2
- [14] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3D primitives for single image understanding. In *ICCV*, 2013. 1, 2
- [15] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *ECCV*, 2014. 1
- [16] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 2, 3
- [17] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014. 6, 7
- [18] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 1
- [19] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2012. 3
- [20] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 2
- [21] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008. 3
- [22] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 7
- [23] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015. 1
- [24] J. J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984. 2
- [25] J. J. Koenderink. *Solid Shape*. 1990. 2
- [26] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8):557–564, 1992. 2
- [27] J. J. Koenderink and A. J. Van Doorn. Relief: Pictorial and otherwise. *Image and Vision Computing*, 13(5):321–334, 1995. 2
- [28] J. J. Koenderink, A. J. Van Doorn, and A. M. Kappers. Surface perception in pictures. *Perception & Psychophysics*, 52(5):487–496, 1992. 2
- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 4
- [30] N. Kumar, A. Berg, P. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 2
- [31] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009. 2
- [32] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 3
- [33] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 1
- [34] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 7, 8
- [35] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 1, 2
- [36] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 7
- [37] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *CVIR*, 2008. 3
- [38] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3D object shape from one depth image. In *CVPR*, 2015. 3
- [39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015. 4, 6
- [40] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 5
- [41] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2004. 5
- [42] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171:701–703, 1971. 2, 7
- [43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 7
- [44] M. J. Tarr and H. H. Bülthoff. Image-based object recognition in man, monkey and machine. *Cognition*, 67(12):1–20, 1998. 2, 7
- [45] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 5
- [46] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, 2015. 3
- [47] A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *Image and Vision Computing*, 7(1):38–42, 1989. 2