Learning Unsupervised Hierarchical Part Decomposition of 3D Objects from a Single RGB Image

Despoina Paschalidou$^{1,3,5}$  Luc van Gool$^{3,4,5}$  Andreas Geiger$^{1,2,5}$  

$^1$Max Planck Institute for Intelligent Systems Tübingen  
$^2$University of Tübingen  
$^3$Computer Vision Lab, ETH Zürich  
$^4$KU Leuven  
$^5$Max Planck ETH Center for Learning Systems  

{firstname.lastname}@tue.mpg.de  vangool@vision.ee.ethz.ch

Abstract

Humans perceive the 3D world as a set of distinct objects that are characterized by various low-level (geometry, reflectance) and high-level (connectivity, adjacency, symmetry) properties. Recent methods based on convolutional neural networks (CNNs) demonstrated impressive progress in 3D reconstruction, even when using a single 2D image as input. However, the majority of these methods focuses on recovering the local 3D geometry of an object without considering its part-based decomposition or relations between parts. We address this challenging problem by proposing a novel formulation that allows to jointly recover the geometry of a 3D object as a set of primitives as well as their latent hierarchical structure without part-level supervision.

Our model recovers the higher level structural decomposition of various objects in the form of a binary tree of primitives, where simple parts are represented with fewer primitives and more complex parts are modeled with more components. Our experiments on the ShapeNet and D-FAUST datasets demonstrate that considering the organization of parts indeed facilitates reasoning about 3D geometry.

1. Introduction

Within the first year of their life, humans develop a common-sense understanding of the physical behavior of the world [2]. This understanding relies heavily on the ability to properly reason about the arrangement of objects in a scene. Early works in cognitive science [21, 3, 27] stipulate that the human visual system perceives objects as a hierarchical decomposition of parts. Interestingly, while this seems to be a fairly easy task for the human brain, computer vision algorithms struggle to form such a high-level reasoning, particularly in the absence of supervision.

The structure of a scene is tightly related to the inherent hierarchical organization of its parts. At a coarse level, a scene can be decomposed into objects and at a finer level each object can be represented with parts and these parts with finer parts. Structure-aware representations go beyond part-level geometry and focus on global relationships between objects and object parts. In this work, we propose a structure-aware representation that considers part relationships (Fig. 1) and models object parts with multiple levels of abstraction, namely geometrically complex parts are modeled with more components and simple parts are modeled with fewer components. Such a multi-scale representation can be efficiently stored at the required level of detail, namely with less parameters (Fig. 2).
Recent breakthroughs in deep learning led to impressive progress in 3D shape extraction by learning a parametric function, implemented as a neural network, that maps an input image to a 3D shape represented as a mesh [32, 17, 25, 53, 58, 39], a pointcloud [13, 43, 1, 24, 49, 58], a voxel grid [5, 8, 14, 44, 45, 48, 55], 2.5D depth maps [26, 19, 42, 11] or an implicit surface [34, 7, 40, 46, 56, 55]. These approaches are mainly focused on reconstructing the geometry of an object, without taking into consideration its constituent parts. This results in non-interpretable reconstructions. To address the lack of interpretability, researchers shifted their attention to representations that employ shape primitives [51, 41, 31, 10]. While these methods yield meaningful semantic shape abstractions, part relationships do not explicitly manifest in their representations.

Instead of representing 3D objects as an unstructured collection of parts, we propose a novel neural network architecture that recovers the latent hierarchical layout of an object without structure supervision. In particular, we employ a neural network that learns to recursively partition an object into its constituent parts by building a latent space that encodes both the part-level hierarchy and the part geometries. The predicted hierarchical decomposition is represented as an unbalanced binary tree of primitives. More importantly, this is learned without any supervision neither on the object parts nor their structure. Instead, our model jointly infers these latent variables during training.

In summary, we make the following contributions: We jointly learn to predict part relationships and per-part geometry without any part-level supervision. The only supervision required for training our model is a watertight mesh of the 3D object. Our structure-aware representation yields semantic shape reconstructions that compare favorably to the state-of-the-art 3D reconstruction approach of [34], using significantly less parameters and without any additional post-processing. Moreover, our learned hierarchies have a semantic interpretation, as the same node in the learned tree is consistently used for representing the same object part. Experiments on the ShapeNet [6] and the Dynamic FAUST (D-FAUST) dataset [4] demonstrate the ability of our model to parse objects into structure-aware representations that are more expressive and geometrically accurate compared to approaches that only consider the 3D geometry of the object parts [51, 41, 15, 9]. Code and data is publicly available.

2. Related Work

We now discuss the most related primitive-based and structure-aware shape representations.

Supervised Structure-Aware Representations: Our work is related to methods that learn structure-aware shape representations that go beyond mere enumeration of object’s parts and recover the higher level structural decomposition of objects based on part-level relations [36]. Li et al. [30] represent 3D shapes using a symmetry hierarchy [54] and train a recursive neural network to predict its hierarchical structure. Their network learns a hierarchical organization of bounding boxes and then fills them with voxelized parts. Note that, this model considers supervision in terms of segmentation of objects into their primitive parts. Closely related to [30] is StructureNet [37] which leverages a graph neural network to represent shapes as n-ary graphs. StructureNet considers supervision both in terms of the primitive parameters and the hierarchies. Likewise, Hu et al. [22] propose a supervised model that recovers the 3D structure of a cable-stayed bridge as a binary parsing tree. In contrast our model is unsupervised, i.e., it does not require supervision neither on the primitive parts nor the part relations.

Physics-Based Structure-Aware Representations: The task of inferring higher-level relationships among parts has also been investigated in different settings. Xu et al. [57] recover the object parts, their hierarchical structure and each part’s dynamics by observing how objects are expected to move in the future. In particular, each part inherits the motion of its parent and the hierarchy emerges by minimizing the norm of these local displacement vectors. Kipf et al. [28] explore the use of variational autoencoders for learning the underlying interaction among various moving particles. Steenkiste et al. [52] extend the work of [16] on perceptual grouping of pixels and learn an interaction function that models whether objects interact with each other at multiple frames. For both [28, 52], the hierarchical structure emerges from interactions at multiple timestamps. In contrast to [57, 28, 52], our model does not relate hierarchies to motion, thus we do not require multiple frames for discovering the hierarchical structure.

1https://github.com/puschalidou/hierarchical_primitives
Supervised Primitive-Based Representations: Zou et al. [59] exploit LSTMs in combination with a Mixture Density Network (MDN) to learn a cuboid representation from depth maps. Similarly, Niu et al. [38] employ an RNN that iteratively predicts cuboid primitives as well as their symmetry and connectivity relationships from RGB images. More recently, Li et al. [31] utilize PointNet++ [43] for predicting per-point properties that are subsequently used for estimating the primitive parameters, by solving a series of linear least-squares problems. In contrast to [59, 38, 43], which require supervision in terms of the primitive parameters, our model is learned in an unsupervised fashion. In addition, modelling primitives with superquadrics, allows us to exploit a larger shape vocabulary that is not limited to cubes as in [59, 38] or spheres, cones, cylinders and planes as in [31]. Another line of work, complementary to ours, incorporates the principles of constructive solid geometry (CSG) [29] in a learning framework for shape modeling [47, 12, 50, 33]. These works require rich annotations for the primitive parameters and the sequence of predictions.

Unsupervised Shape Abstraction: Closely related to our model are the works of [51, 41] that employ a convolutional neural network (CNN) to regress the parameters of the primitives that best describe the target object, in an unsupervised manner. Primitives can be cuboids [51] or superquadrics [41] and are learned by minimizing the discrepancy between the target and the predicted shape [41]. While these methods learn a flat arrangement of parts, our structure-aware representation decomposes the depicted object into a hierarchical layout of semantic parts. This results in part geometries with different levels of granularity. Our model differs from [51, 41] also wrt. the optimization objective. We empirically observe that for both [51, 41], the proposed loss formulations suffer from various local minima that stem from the nature of their optimization objective. To mitigate this, we use the more robust classification loss proposed in [34, 7, 40] and train our network by learning to classify whether points lie inside or outside the target object. Very recently, [15, 9] explored such a loss for recovering shape elements from 3D objects. Genova et al. [15] leverage a CNN to learn to predict the parameters of a set of axis-aligned 3D Gaussians from a set of depth maps rendered at different viewpoints. Similarly, Deng et al. [9] employ an autoencoder to recover the geometry of an object as a collection of smooth convexes. In contrast to [15, 9], our model goes beyond the local geometry of parts and attempts to recover the underlying hierarchical structure of the object parts.

3. Method

In this section, we describe our novel neural network architecture for inferring structure-aware representations. Given an input \( I \) (e.g., RGB image, voxel grid) our goal is to learn a neural network \( \phi_\theta \), which maps the input to a set of primitives that best describe the target object. The target object is represented as a set of pairs \( \mathcal{X} = \{(x_i, o_i)\}_{i=1}^N \).
where \( x_i \) corresponds to the location of the \( i \)-th point and \( o_i \) denotes its label, namely whether \( x_i \) lies inside \((o_i = 1)\) or outside \((o_i = 0)\) the target object. We acquire these \( N \) pairs by sampling points inside the bounding box of the target mesh and determine their labels using a watertight mesh of the target object. During training, our network learns to predict shapes that contain all internal points from the target mesh \((o_i = 1)\) and none of the external \((o_i = 0)\). We discuss our sampling strategy in our supplementary.

Instead of predicting an unstructured set of primitives, we recover a hierarchical decomposition over parts in the form of a binary tree of maximum depth \( D \) as

\[
\mathcal{P} = \{(p_{k}^{d})_{k=0}^{2^{d}-1} \mid d = \{0 \ldots D\}\}
\]

where \( p_{k}^{d} \) is the \( k \)-th primitive at depth \( d \). Note that for the \( k \)-th node at depth \( d \), its parent is defined as \( p_{d-1}^{k} \) and its two children as \( p_{2k}^{d+1} \) and \( p_{2k+1}^{d+1} \).

At every depth level, \( \mathcal{P} \) reconstructs the target object with \( \{1, 2, \ldots, M\} \) primitives. \( M \) is an upper limit to the maximum number of primitives and is equal to \( 2^{D} \). More specifically, \( \mathcal{P} \) is constructed as follows: the root node is associated with the root primitive that represents the entire shape and is recursively split into two nodes (its children) until reaching the maximum depth \( D \). This recursive partition yields reconstructions that recover the geometry of the target shape using \( 2^{d} \) primitives, where \( d \) denotes the depth level (see Fig. 2). Throughout this paper, the term node is used interchangeably with primitive and always refers to the primitive associated with this particular node.

Every primitive is fully described by a set of parameters \( \lambda_{k}^{d} \) that define its shape, size and position in 3D space. Since not all objects require the same number of primitives, we enable our model to predict unbalanced trees, i.e. stop recursive partitioning if the reconstruction quality is sufficient. To achieve this our network also regresses a reconstruction quality for each primitive denoted as \( q_{k}^{d} \). Based on the value of each \( q_{k}^{d} \) the network dynamically stops the recursive partitioning process resulting in parsimonious representations as illustrated in Fig. 1.

### 3.1. Network Architecture

Our network comprises three main components: (i) the partition network that recursively splits the shape representation into representations of parts, (ii) the structure network that focuses on learning the hierarchical arrangement of primitives, namely assigning parts of the object to the primitives at each depth level and (iii) the geometry network that recovers the primitive parameters. An overview of the proposed pipeline is illustrated in Fig. 3. The first part of our pipeline is a feature encoder, implemented with a ResNet-18 [20], ignoring the final fully connected layer. Instead, we only keep the feature vector of length \( F = 512 \) after average pooling.

#### Partition Network:

The feature encoder maps the input \( \mathbf{I} \) to an intermediate feature representation \( c_{k}^{d} \in \mathbb{R}^{F} \) that describes the root node \( p_{0}^{0} \). The partition network implements a function \( p_{0} : \mathbb{R}^{F} \to \mathbb{R}^{2^{D}} \) that recursively partitions the feature representation \( c_{k}^{d} \) of node \( p_{k}^{d} \) into two feature representations, one for each children \( \{p_{2k}^{d+1}, p_{2k+1}^{d+1}\} \):

\[
p_{0}(c_{k}^{d}) = \{c_{2k}^{d+1}, c_{2k+1}^{d+1}\}.
\]

Each primitive \( p_{k}^{d} \) is directly predicted from \( c_{k}^{d} \) without considering the other intermediate features. This implies that the necessary information for predicting the primitive parameterization is entirely encapsulated in \( c_{k}^{d} \) and not in any other intermediate feature representation.

#### Structure Network:

Due to the lack of ground-truth supervision in terms of the tree structure, we introduce the structure network that seeks to learn a pseudo-ground-truth part-decomposition of the target object. More formally, it learns a function \( s_{\theta} : \mathbb{R}^{F} \to \mathbb{R}^{3} \) that maps each feature representation \( c_{k}^{d} \) to \( h_{k}^{d} \) a spatial location in \( \mathbb{R}^{3} \).

One can think of each \( h_{k}^{d} \) as the (geometric) centroid of a specific part of the target object. We define

\[
\mathcal{H} = \{(h_{k}^{d})_{k=0}^{2^{d}-1} \mid d = \{0 \ldots D\}\}
\]

the set of centroids of all parts of the object at all depth levels. From \( \mathcal{H} \) and \( \mathcal{X} \), we are now able to derive the part decomposition of the target object as the set of points \( \lambda_{k}^{d} \) that are internal to a part with centroid \( h_{k}^{d} \).

Note that, in order to learn \( \mathcal{P} \), we need to be able to partition the target object into \( 2^{d} \) parts at each depth level. At the root level \((d = 0)\), \( h_{0}^{0} \) is the centroid of the target object and \( \lambda_{0}^{0} \) is equal to \( \mathcal{X} \). For \( d = 1 \), \( h_{0}^{1} \) and \( h_{1}^{1} \) are the centroids of the two parts representing the target object. \( \lambda_{0}^{1} \) and \( \lambda_{1}^{1} \) comprise the same points as \( \lambda_{0}^{0} \). For the external points, the labels remain the same. For the internal points, however, the labels are distributed between \( \lambda_{0}^{1} \) and \( \lambda_{1}^{1} \) based on whether \( h_{0}^{1} \) or \( h_{1}^{1} \) is closer. That is, \( \lambda_{0}^{1} \) and \( \lambda_{1}^{1} \) each contain more external labels and less internal labels compared to \( \lambda_{0}^{0} \). The same process is repeated until we reach the maximum depth.

More formally, we define the set of points \( \lambda_{k}^{d} \) corresponding to primitive \( p_{k}^{d} \) implicitly via its centroid \( h_{k}^{d} \):

\[
\lambda_{k}^{d} = \{N_{k}(x, o) \mid \forall(x, o) \in \lambda_{k}^{d-1}\} \quad (4)
\]

Here, \( \lambda_{k}^{d-1} \) denotes the points of the parent. The function \( N_{k}(x, o) \) assigns each \((x, o) \in \lambda_{k}^{d-1}\) to part \( p_{k}^{d} \) if it is closer to \( h_{k}^{d} \) than to \( h_{s(k)}^{d} \) where \( s(k) \) is the sibling of \( k \):

\[
N_{k}(x, o) = \begin{cases} (x, 1) & \|h_{k}^{d} - x\| \leq \|h_{s(k)}^{d} - x\| \wedge o = 1 \\ (x, 0) & \text{otherwise} \end{cases}
\]

\[(5)\]
Figure 4: Structure Network. We visualize the centroids $h_k^d$ and the 3D points $X_k^d$ that correspond to the estimated part $p_k$ for the first three levels of the tree. Fig. 4b explains visually Eq. (4). We color points based on their closest centroid $h_k^o$. Points illustrated with the color associated to a part are labeled “internal” ($o = 1$). Points illustrated with gray are labeled “external” ($o = 0$).

Intuitively, this process recursively associates points to the closest sibling at each level of the binary tree where the association is determined by the label $o$. Fig. 4 illustrates the part decomposition of the target shape using $\mathcal{H}$. We visualize each part with a different color.

Geometry Network: The geometry network learns a function $r_\theta : \mathbb{R}^K \rightarrow \mathbb{R}^K \times [0, 1]$ that maps the feature representation $c_k^d$ to its corresponding primitive parametrization $X_k^d$ and the reconstruction quality prediction $q_k^d$:

$$r_\theta(c_k^d) = \{X_k^d, q_k^d\}. \quad (6)$$

3.2. Primitive Parametrization

For primitives, we use superquadric surfaces. A detailed analysis of the use of superquadrics as geometric primitives is beyond the scope of this paper, thus we refer the reader to [23, 41] for more details. Below, we focus on the properties most relevant to us. For any point $x \in \mathbb{R}^3$, we can determine whether it lies inside or outside a superquadric using its implicit surface function which is commonly referred to as the inside-outside function:

$$f(x; \lambda) = \left( \left( \frac{x}{\alpha_1} \right)^{\frac{2}{d}} + \left( \frac{y}{\alpha_2} \right)^{\frac{2}{d}} + \left( \frac{z}{\alpha_3} \right)^{\frac{2}{d}} \right)^{\frac{d}{2}} \quad (7)$$

where $\alpha = [\alpha_1, \alpha_2, \alpha_3]$ determine the size and $\epsilon = [\epsilon_1, \epsilon_2]$ the shape of the superquadric. If $f(x; \lambda) = 1.0$, the given point $x$ lies on the surface of the superquadric, if $f(x; \lambda) < 1.0$ the corresponding point lies inside and if $f(x; \lambda) > 1.0$ the point lies outside the superquadric. To account for numerical instabilities that arise from the exponentiations in (7), instead of directly using $f(x; \lambda)$, we follow [23] and use $f(x; \lambda)^{\epsilon\cdot\theta}$. Finally, we convert the inside-outside function to an occupancy function, $g : \mathbb{R}^3 \rightarrow [0, 1]$:

$$g(x; \lambda) = \sigma(s(1 - f(x; \lambda)^{\epsilon\cdot\theta})) \quad (8)$$

that results in per-point predictions suitable for the classification problem we want to solve, $\sigma(\cdot)$ is the sigmoid function and $s$ controls the sharpness of the transition of the occupancy function. To account for any rigid body motion transformations, we augment the primitive parameters with a translation vector $t = [t_x, t_y, t_z]$ and a quaternion $q = [q_0, q_1, q_2, q_3]$ [18], which determine the coordinate system transformation $T(x) = R(\lambda) x + t(\lambda)$. Note that in (7), (8) we omit the primitive indexes $k, d$ for clarity. Visualizations of (8) are given in our supplementary.

3.3. Network Losses

Our optimization objective $\mathcal{L}(\mathcal{P}, \mathcal{H}; \mathcal{X})$ is a weighted sum over four terms:

$$\mathcal{L}(\mathcal{P}, \mathcal{H}; \mathcal{X}) = \mathcal{L}_{str}(\mathcal{H}; \mathcal{X}) + \mathcal{L}_{rec}(\mathcal{P}; \mathcal{X})$$

$$+ \mathcal{L}_{comp}(\mathcal{P}; \mathcal{X}) + \mathcal{L}_{prox}(\mathcal{P}) \quad (9)$$

Structure Loss: Using $\mathcal{H}$ and $\mathcal{X}$, we can decompose the target mesh into a hierarchy of disjoint parts. Namely, each $h_k^d$ implicitly defines a set of points $X_k^d$ that describe a specific part of the object as described in (4). To quantify how well $\mathcal{H}$ clusters the input shape $\mathcal{X}$ we minimize the sum of squared distances, similar to classical k-means:

$$\mathcal{L}_{str}(\mathcal{H}; \mathcal{X}) = \sum_{h_k^d \in \mathcal{H}} \frac{1}{2^d - 1} \sum_{(x,o) \in X_k^d} o \left\| x - h_k^d \right\|_2^2 \quad (10)$$

Note that for the loss in (10), we only consider gradients with respect to $\mathcal{H}$ as $X_k^d$ is implicitly defined via $\mathcal{H}$. This results in a procedure resembling Expectation-Maximization (EM) for clustering point clouds, where computing $X_k^d$ is the expectation step and each gradient updated corresponds to the maximization step. In contrast to EM, however, we minimize this loss across all instances of the training set, leading to parsimonious but consistent shape abstractions. An example of this clustering process performed at training-time is shown in Fig. 4.

Reconstruction Loss: The reconstruction loss measures how well the predicted primitives match the target shape. Similar to [15, 9], we formulate our reconstruction loss as a
binary classification problem, where our network learns to predict the surface boundary of the predicted shape by classifying whether points in \( \mathcal{X} \) lie inside or outside the target object. To do this, we first define the occupancy function of the predicted shape at each depth level. Using the occupancy function of each primitive defined in (8), the occupancy function of the overall shape at depth \( d \) becomes:

\[
G^d(x) = \max_{k \in 0...2^d-1} q_k^d(x; \lambda_k^d) \tag{11}
\]

Note that (11) is simply the union of the per-primitive occupancy functions. We formulate our reconstruction loss wrt. the object and wrt. each part of the object as follows

\[
L_{rec}(P; \mathcal{X}) = \sum_{(x,o) \in \mathcal{X}} \sum_{d=0}^{D} L(G^d(x), o) + \sum_{d=0}^{D} \sum_{k=0}^{2^d-1} L(g_k^d(x; \lambda_k^d), o) \tag{12}
\]

\[
L_{comp}(P; \mathcal{X}) = \sum_{d=0}^{D} \sum_{k=0}^{2^d-1} (q_k^d - IoU(p_k^d, \lambda_k^d))^2 \tag{13}
\]

where \( L(\cdot) \) is the binary cross entropy loss. The first term is an object reconstruction loss (12) and measures how well the predicted shape at each depth level matches the target shape. The second term (13) which we refer to as part reconstruction loss measures how accurately each primitive \( p_k^d \) matches the part of the object it represents, defined as the point set \( \lambda_k^d \). Note that the part reconstruction loss enforces non-overlapping primitives, as \( \lambda_k^d \) are non-overlapping by construction. We illustrate our reconstruction loss in Fig. 3.

**Compatibility Loss:** This loss measures how well our model is able to predict the expected reconstruction quality \( q_k^d \) of a primitive \( p_k^d \). A standard metric for measuring the reconstruction quality is the Intersection over Union (IoU). We therefore task our network to predict the reconstruction quality of each primitive \( p_k^d \) in terms of its IoU wrt. the part of the object \( \lambda_k^d \) it represents:

\[
L_{comp}(P; \mathcal{X}) = \sum_{d=0}^{D} \sum_{k=0}^{2^d-1} (q_k^d - IoU(p_k^d, \lambda_k^d))^2 \tag{14}
\]

During inference, \( q_k^d \) allows for further partitioning primitives whose IoU is below a threshold \( q_{th} \) and to stop if the reconstruction quality is high (the primitive fits the object well). As a result, our model predicts an unbalanced tree of primitives where objects can be represented with various number of primitives from 1 to \( 2^D \). This results in parsimonious representations where simple parts are represented with fewer primitives. We empirically observe that the threshold value \( q_{th} \) does not significantly affect our results, thus we empirically set it to 0.6. During training, we do not use the predicted reconstruction quality \( q_k^d \) to dynamically partition the nodes but instead predict the full tree.

**Proximity Loss:** This term is added to counteract vanishing gradients due to the sigmoid in (8). For example, if the initial prediction of a primitive is far away from the target object, the reconstruction loss will be large while its gradients will be small. As a result, it is impossible to “move” this primitive to the right location. Thus, we introduce a proximity loss which encourages the center of each primitive \( p_k^d \) to be close to the centroid of the part it represents:

\[
L_{prox}(P) = \sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \| t(\lambda_k^d) - \mathbf{h}_k^d \|_2 \tag{15}
\]

where \( t(\lambda_k^d) \) is the translation vector of the primitive \( p_k^d \) and \( \mathbf{h}_k^d \) is the centroid of the part it represents. We demonstrate the vanishing gradient problem in our supplementary.

**4. Experiments**

In this section, we provide evidence that our structure-aware representation yields semantic shape abstractions while achieving competitive (or even better results) than various state-of-the-art shape reconstruction methods, such as [34]. Moreover, we also investigate the quality of the learned hierarchies and show that the use of our structure-aware representation yields semantic scene parsings. Im-

---

**Figure 5: Predicted Hierarchies on ShapeNet.** Our model recovers the geometry of an object as an unbalanced hierarchy over primitives, where simpler parts (e.g. base of the lamp) are represented with few primitives and more complex parts (e.g. wings of the plane) with more.
Table 1: Single Image Reconstruction on ShapeNet. Quantitative evaluation of our method against OccNet [34] and primitive-based methods with superquadrics [41] (SQs), SIF [15] and CvxNets [9]. We report the volumetric IoU (higher is better) and the Chamfer-$L_1$ distance (lower is better) wrt. the ground-truth mesh.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>airplane</td>
<td>0.147</td>
<td>0.122</td>
<td>0.065</td>
<td>0.093</td>
<td>0.175</td>
<td>0.571</td>
<td>0.456</td>
<td>0.530</td>
<td>0.598</td>
<td>0.529</td>
</tr>
<tr>
<td>bench</td>
<td>0.155</td>
<td>0.114</td>
<td>0.131</td>
<td>0.133</td>
<td>0.153</td>
<td>0.485</td>
<td>0.202</td>
<td>0.333</td>
<td>0.461</td>
<td>0.437</td>
</tr>
<tr>
<td>cabinet</td>
<td>0.167</td>
<td>0.087</td>
<td>0.102</td>
<td>0.160</td>
<td>0.087</td>
<td>0.733</td>
<td>0.110</td>
<td>0.648</td>
<td>0.709</td>
<td>0.658</td>
</tr>
<tr>
<td>car</td>
<td>0.159</td>
<td>0.117</td>
<td>0.056</td>
<td>0.103</td>
<td>0.141</td>
<td>0.737</td>
<td>0.650</td>
<td>0.657</td>
<td>0.675</td>
<td>0.702</td>
</tr>
<tr>
<td>chair</td>
<td>0.228</td>
<td>0.138</td>
<td>0.192</td>
<td>0.337</td>
<td>0.114</td>
<td>0.501</td>
<td>0.176</td>
<td>0.389</td>
<td>0.491</td>
<td>0.526</td>
</tr>
<tr>
<td>display</td>
<td>0.278</td>
<td>0.106</td>
<td>0.208</td>
<td>0.223</td>
<td>0.137</td>
<td>0.471</td>
<td>0.200</td>
<td>0.491</td>
<td>0.576</td>
<td>0.633</td>
</tr>
<tr>
<td>lamp</td>
<td>0.479</td>
<td>0.189</td>
<td>0.454</td>
<td>0.795</td>
<td>0.169</td>
<td>0.371</td>
<td>0.189</td>
<td>0.260</td>
<td>0.311</td>
<td>0.441</td>
</tr>
<tr>
<td>speaker</td>
<td>0.300</td>
<td>0.132</td>
<td>0.253</td>
<td>0.462</td>
<td>0.108</td>
<td>0.647</td>
<td>0.136</td>
<td>0.577</td>
<td>0.620</td>
<td>0.660</td>
</tr>
<tr>
<td>rifle</td>
<td>0.141</td>
<td>0.127</td>
<td>0.069</td>
<td>0.106</td>
<td>0.203</td>
<td>0.474</td>
<td>0.519</td>
<td>0.463</td>
<td>0.515</td>
<td>0.435</td>
</tr>
<tr>
<td>sofa</td>
<td>0.194</td>
<td>0.106</td>
<td>0.146</td>
<td>0.164</td>
<td>0.128</td>
<td>0.680</td>
<td>0.122</td>
<td>0.606</td>
<td>0.677</td>
<td>0.693</td>
</tr>
<tr>
<td>table</td>
<td>0.189</td>
<td>0.110</td>
<td>0.264</td>
<td>0.358</td>
<td>0.122</td>
<td>0.506</td>
<td>0.180</td>
<td>0.372</td>
<td>0.473</td>
<td>0.491</td>
</tr>
<tr>
<td>phone</td>
<td>0.140</td>
<td>0.112</td>
<td>0.095</td>
<td>0.083</td>
<td>0.149</td>
<td>0.720</td>
<td>0.185</td>
<td>0.658</td>
<td>0.719</td>
<td>0.770</td>
</tr>
<tr>
<td>vessel</td>
<td>0.218</td>
<td>0.125</td>
<td>0.108</td>
<td>0.173</td>
<td>0.178</td>
<td>0.530</td>
<td>0.471</td>
<td>0.502</td>
<td>0.552</td>
<td>0.570</td>
</tr>
<tr>
<td>mean</td>
<td>0.215</td>
<td>0.122</td>
<td>0.165</td>
<td>0.245</td>
<td>0.143</td>
<td>0.571</td>
<td>0.277</td>
<td>0.499</td>
<td>0.567</td>
<td>0.580</td>
</tr>
</tbody>
</table>

Figure 6: Predicted Hierarchies on D-FAUST. We visualize the input RGB image (a), the prediction (b) and the predicted hierarchy (c). We associate each primitive with a color and we observe that our network learns semantic mappings of body parts across different articulations, e.g. node (3, 3) is used for representing the upper part of the left leg, whereas node (1, 1) is used for representing the upper body.

Datasets: First, we use the ShapeNet [6] subset of Choy et al. [8], training our model using the same image renderings and train/test splits as Choy et al. Furthermore, we also experiment with the Dynamic FAUST (D-FAUST) dataset [4], which contains meshes for 129 sequences of 10 humans performing various tasks, such as “running”, “punching” or “shake arms”. We randomly divide these sequences into training (91), test (29) and validation (9).

Baselines: Closely related to our work are the shape parsing methods of [51] and [41] that employ cuboids and superquadric surfaces as primitives. We refer to [41] as SQs and we evaluate using their publicly available code. Moreover, we also compare to the Structured Implicit Function (SIF) [15] that represent the object’s geometry as the isolevel of the sum of a set of Gaussians and to the CvxNets [9] that represent the object parts using smooth convex shapes. Finally, we also report results for OccNet [34], which is the state-of-the-art implicit shape reconstruction technique. Note that in contrast to us, [34] does not consider part decomposition or any form of latent structure.

Evaluation Metrics: Similar to [34, 15, 9], we evaluate our model quantitatively and report the mean Volumetric IoU and the Chamfer-$L_1$ distance. Both metrics are discussed in detail in our supplementary.

4.1. Results on ShapeNet

We evaluate our model on the single-view 3D reconstruction task and compare against various state-of-the-art methods. We follow the standard experimental setup and train a single model for the 13 ShapeNet objects. Both our model and [41] are trained for a maximum number of 64 epochs. Details and ablations on the impact of various components of our model are detailed in the supplementary.
visualizes the predicted hierarchy. Additional qualitative results are provided in the supplementary.

primitives \((D = 6)\). For SIF [15] and CvxNets [9] the reported results are computed using 50 shape elements. The quantitative results are reported in Table 1. We observe that our model outperforms the primitive-based baselines in terms of the IoU as well as the OccNet [34] for the majority of objects \((7/13)\). Regarding Chamfer-\(L_1\), our model is the second best amongst primitive representations, as [41] is optimized for this metric. This also justifies that [41] performs worse in terms of IoU. While our model performs on par with existing state-of-the-art primitive representations in terms of Chamfer-\(L_1\), it also recovers hierarchies, which none of our baselines do. A qualitative comparison of our model with SQs [41] is depicted in Fig. 7. Fig. 5 visualizes the learned hierarchy for this model. We observe that our model recovers unbalanced binary trees that decompose a 3D object into a set of parts. Note that [51, 41] were originally introduced for volumetric 3D reconstruction, thus we provide an experiment on this task in our supplementary.

4.2. Results on D-FAUST

We also demonstrate results on the Dynamic FAUST (D-FAUST) dataset [4], which is very challenging due to the fine structure of the human body. We evaluate our model on the single-view 3D reconstruction task and compare with [41]. Both methods are trained for a maximum number of 32 primitives \((D = 5)\). Fig. 6 illustrates the predicted hierarchy on different humans from the test set. We note that the predicted hierarchies are indeed semantic, as the same nodes are used for modelling the same part of the human body. Fig. 8 compares the predictions of our model with SQs. We observe that while our baseline yields more parsimonious abstractions, their level of detail is limited. On the contrary, our model captures the geometry of the human body with more detail. This is also validated quantitatively, from Table 2. Note that in contrast to ShapeNet, D-FAUST does not contain long, thin (e.g. legs of tables, chairs) or hollow parts (e.g. cars), thus optimizing for either Chamfer-\(L_1\) or IoU leads to similar results. Hence, our method out-

![Figure 7: Single Image 3D Reconstruction. The input image is shown in (a, d), the other columns show the results of our method (c, f) compared to [41] (b, e). Additional qualitative results are provided in the supplementary.](image)

![Figure 8: Single Image 3D Reconstruction. Qualitative comparison of our reconstructions (c, f), to [41] (b, e). The input RGB image is shown in (a, d). Note how our representation yields geometrically more accurate reconstructions, while being semantic, e.g., the primitive colored in blue consistently represents the head of the human while the primitive colored in orange captures the left thigh. Additional qualitative results are provided in the supplementary.](image)

<table>
<thead>
<tr>
<th></th>
<th>IoU</th>
<th>Chamfer-(L_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQs [41]</td>
<td>0.608</td>
<td>0.189</td>
</tr>
<tr>
<td>Ours</td>
<td>0.699</td>
<td>0.098</td>
</tr>
</tbody>
</table>

Table 2: Single Image Reconstruction on D-FAUST. We report the volumetric IoU and the Chamfer-\(L_1\) wrt. the ground-truth mesh for our model compared to [41].

performs [41] also in terms of Chamfer-\(L_1\). Due to lack of space, we only illustrate the predicted hierarchies up to the fourth depth level. The full hierarchies are provided in the supplementary.

5. Conclusion

We propose a learning-based approach that jointly predicts part relationships together with per-part geometries in the form of a binary tree without requiring any part-level annotations for training. Our model yields geometrically accurate primitive-based reconstructions that outperform existing shape abstraction techniques while performing competitively with more flexible implicit shape representations. In future work, we plan to extend our model and predict hierarchical structures that remain consistent in time, such as yielding kinematic trees of objects. Another future direction is to consider more flexible primitives such as general convex shapes and incorporate additional constraints e.g. symmetry to further improve the reconstructions.

Acknowledgments

This research was supported by the Max Planck ETH Center for Learning Systems and a HUAWEI research gift.
References


