

OMAD: Object Model with Articulated Deformations for Pose Estimation and Retrieval

Han Xue*
xiaoxiaoxh@sjtu.edu.cn

Liu Liu*
liuliu1993@sjtu.edu.cn

Wenqiang Xu
vinjohn@sjtu.edu.cn

Haoyuan Fu
simon-fuhaoyuan@sjtu.edu.cn

Cewu Lu[†]
lucewu@sjtu.edu.cn

Shanghai Jiao Tong University
Shanghai, CN

Abstract

Articulated objects are pervasive in daily life. However, due to the intrinsic high-DoF structure, the joint states of the articulated objects are hard to be estimated. To model articulated objects, two kinds of shape deformations namely the geometric and the pose deformation should be considered. In this work, we present a novel category-specific parametric representation called Object Model with Articulated Deformations (OMAD) to explicitly model the articulated objects. In OMAD, a category is associated with a linear shape function with shared shape basis and a non-linear joint function. Both functions can be learned from a large-scale object model dataset and fixed as category-specific priors. Then we propose an OMADNet to predict the shape parameters and the joint states from an object's single observation. With the full representation of the object shape and joint states, we can address several tasks including category-level object pose estimation and the articulated object retrieval. To evaluate these tasks, we create a synthetic dataset based on PartNet-Mobility. Extensive experiments show that our simple OMADNet can serve as a strong baseline for both tasks. Our code and dataset are available in <https://sites.google.com/view/omad-bmvc/>.

1 Introduction

Articulated objects are very common in daily life. A vision system that can model articulated objects from a single observation (*e.g.* depth image, point cloud) can be beneficial for the downstream tasks such as robot manipulation [6], AR/VR [22] and CG modeling [29]. However, visual analysis of articulated objects is challenging due to their high-DoF (Degree of Freedom) nature in 3D space and large intra-class appearance variations. In this work, we

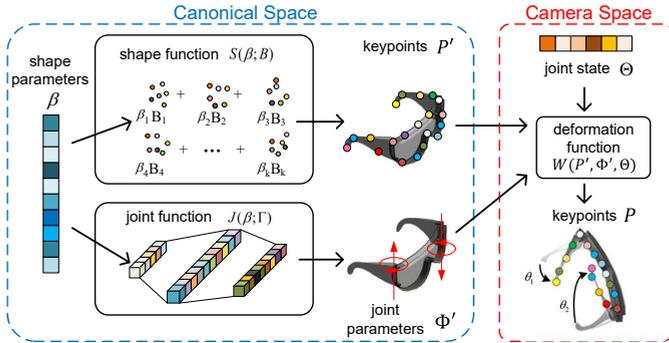


Figure 1: Our OMAD representation for modeling articulated objects. In OMAD, we use shape parameters β to capture shape geometric deformations and use joint states Θ to capture pose deformations. We use $'$ to represent the variables in canonical space (e.g. P'), which distinguishes from variables in camera space (e.g. P).

propose an explicit parametric representation to depict shape and pose of articulated objects in the same category. There are two sources of articulated deformations, namely shape geometric deformation on parts and pose deformations on joints. To handle both deformations, we propose a novel **Object Model with Articulated Deformations (OMAD)** (Fig. 1).

To describe the shape geometric deformations, given an articulated object, we consider its rest structure (i.e. shape geometry, joint parameters) in the axis-aligned, zero-centered canonical space. Then for its shape, which is represented by ordered sparse keypoints, can be derived from the proposed shape function $P' = S(\beta; B)$, where β is a pose-invariant shape parameters in canonical space, and B is the category-specific linear shape basis. For the joint parameters Φ' , we devise a non-linear joint function $\Phi' = J(\beta; \Gamma)$ with parameters Γ . The parameters $\{B, \Gamma\}$ of shape function and joint function can be regarded as category-specific priors, and can be learned from existing object dataset like PartNet-Mobility [23] with proposed OMAD-PriorNet. In this way, the shape geometric deformation can be fully described by (P', Φ') . On the other hand, the pose deformation on joints can simply defined by its joint state Θ in camera space.

To combine both deformations, we devise a deformation function $W(P', \Phi', \Theta)$. It can transform the rest structure in canonical space to camera space. In this sense, we can depict arbitrary articulated object of a known category simply with β and Θ . To predict β and Θ , we propose a novel OMADNet, which is mainly constituted by a neural prediction stage for coarse estimation of β and Θ and an optimization stage to refine the prediction results. To show the applicability of the proposed OMADNet, we apply it into two different tasks, namely articulated object pose estimation and articulated object retrieval. The predicted shape parameters β can be used for pose-oblivious object retrieval [8]. Besides, the predicted joint state Θ and joint parameters Φ' can directly address the object pose estimation task. To evaluate our method, we select articulated object models of 5 categories from PartNet-Mobility[23] to create a dataset named ArtImage. For the articulated object pose estimation task, we achieve better results on joint state error and joint distance error compared to existing method. For the object retrieval task, we achieve 82.3 average mAP on 5 categories.

Our contribution can be summarized as follows:

- A novel category-specific articulated modeling approach named OMAD, which ex-

licitly models both shape deformation and pose deformation at category-level.

- With the proposed OMAD, we provide a simple OMADNet to estimate the shape parameters and joint states required to describe the articulated object from its single observation. The output of OMADNet can be used to address the category-level object pose estimation task and the articulated object retrieval task.

2 Related Work

Articulated Object Shape Signature and Retrieval. The shape parameters in our proposed OMAD which capture the geometric features of articulated objects and summarize them into a vector are mostly like the shape signature adopted in [2, 8, 12, 13, 18, 19]. However, these shape signatures cannot be directly applied to our case for at least two reasons: (1) Their methods are largely relied on the mesh representation, such as spectral-based [12], graph-based [2], geodesic-based [8], SDF-based [18], while our shape is in point cloud format. (2) None of the previous methods are interested in the joint parameters of the articulated objects, while our representation takes both joint states and joint parameters into account. For articulated object retrieval task, the standard benchmarks are the the McGill shape benchmark (MSB) [22] and ISDB dataset [8]. However, the object models for a certain category are very few for both datasets, which cannot support the training of our proposed OMAD-PriorNet and OMADNet.

Low-rank Shape Prior. Our proposed OMAD adopts a low-rank shape prior based approach to model the sparse shape. Such choice has been proved effective in many tasks, such as instance segmentation [23], NRSfM [3, 4, 14, 25], object structure learning [2], human body modeling [17, 21]. To the best of our knowledge, we are the first to apply it to the generic articulated object modeling. It is worth noting that our method differs from SMPL [15] and its variants [24] from at least two aspects: (1) SMPL [15] relies on hand-crafted mesh templates with fixed topology and vertex number, while our OMAD only requires flexible point cloud input. (2) SMPL [15] performs PCA on standard meshes to obtain shape basis, while our proposed OMAD-PriorNet can learn shape basis in an unsupervised manner directly from raw point cloud.

Category-level Articulated Object Pose Estimation. Category-level object pose estimation task aims at predicting poses of unseen objects in the same category, which is firstly introduced by NOCS [27]. A-NCSH [16] extends this task to articulated objects. Later, several works on this task are conducted on videos [11, 18]. Our proposed OMADNet can address the category-level articulated object pose estimation task given a single depth image as input. Overall, our work is closely related to [16] with significant different on methodology. Firstly, [16] takes a part-centered perspective, the joint states are estimated from the predicted part segmentations and poses. Our method takes a joint-centered perspective, the joint states are directly predicted by OMADNet. Secondly, [16] uses dense NOCS coordinates to represent articulated objects, while our method use ordered keypoints to obtain sparse correspondence, which largely reduces optimization complexity. Thirdly, Li et al. [16] models the shape deformation implicitly with segmentation and NOCS coordinates, which cannot explicitly reflects the complete shape deformation by parameterization as our proposed OMAD representation. Thus it cannot support other important tasks like object retrieval.

3 Object Model with Articulated Deformations, OMAD

In this section, we will give a definition about the articulated objects which the OMAD can support (Sec. 3.1). As the representation of pose deformation on the joints are self-evident, we will mainly discuss how to model shape geometric deformations in canonical space (Sec. 3.2) with shape function (Sec. 3.2.1) and joint function (Sec. 3.2.2). A general articulated object in camera space is formed by transforming the keypoints and joint parameters in canonical space via a deformation function (Sec. 3.3).

3.1 Object Articulation

We take a joint-centered perspective to study articulated objects. Kinematic tree is one kind of abstraction of articulated objects (See Fig. 2). We denote the joint that connects one node and its parent node in the kinematic tree as the **reference joint** of this node. For example, if joint k connects node a and its parent node b , then joint k is the reference joint of node a .

Here we consider three types of joints: free joint, revolute joint and prismatic joint (See Fig. 2). (1) Free joint is the joint that connects the world node with the base part (root node) in the kinematic tree. The joint state of free joint can be defined as $\theta^{(free)} = (\mathbf{R}^{(f)}, \mathbf{t}^{(f)})$, which can be seen as the 6D pose of base part in the world coordinate space. (2) For revolute joint, joint parameters can be defined as $\phi^{(r)} = (\mathbf{u}^{(r)}, \mathbf{q}^{(r)})$, where $\mathbf{u}^{(r)}$ represents the direction of joint axis and $\mathbf{q}^{(r)}$ represents the pivot point of joint axis. The joint state is denoted by $\theta^{(r)}$, which is the relative rotation angle compared to a pre-defined rest state in canonical space. (3) For prismatic joint, joint parameters can be defined as ϕ^p , where $\mathbf{u}^{(p)}$ represents the direction of joint axis. The joint state is denoted by $\theta^{(p)}$, which is the relative distance along the joint axis compared to a pre-defined rest state in canonical space.

With the introduction of reference joint and free joint, the part number and the joint number can be equal. Given an articulated object with K joints, the overall joint state is denoted as $\Theta = \{\theta_1, \dots, \theta_K\}$.

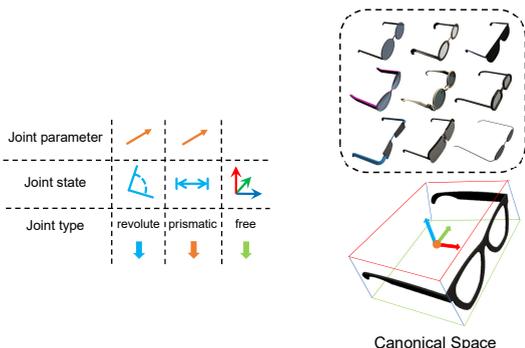
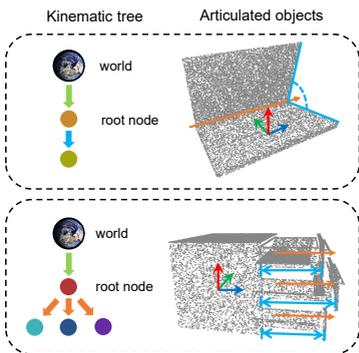


Figure 2: Joint types, parameters and states in consideration. Figure 3: Examples of objects in canonical space.

3.2 Geometric Deformation in Canonical Space

As pose deformation can be independently considered by Θ , we model the shape geometric deformation in the canonical space (See Fig. 3). Objects of the same category in the

canonical space have the following properties: (1) The directions in the joint parameters are aligned. (2) The whole object is zero-centered. (3). The joint states are the same and pre-defined. To note, our canonical space does not normalize the object scale, because object scale is a crucial part of the shape geometry and can be explicitly modeled by the OMAD representation. We use \prime to represent the variables in canonical space (e.g. \mathbf{P}'), which distinguishes from variables in camera space (e.g. \mathbf{P}). As the articulation is associated with shape and joint, thus we model the geometric deformation with a shape function and joint function.

3.2.1 Shape Function

As detailed geometric deformation is hard to be modeled, we adopt the sparse shape representation, which seeks sparse keypoint correspondences among objects from the same category. Inspired by [2], we predict the sparse shape with the low-rank shape prior. Given a complete object point cloud in canonical space, the sparse shape \mathbf{P}' with M keypoints can be generated by shape parameters $\boldsymbol{\beta}$ using shape function $S(\cdot)$:

$$\mathbf{P}' = S(\boldsymbol{\beta}; \mathbf{B}) = \mathbf{B}\boldsymbol{\beta} = \sum_{i=1}^{|\boldsymbol{\beta}|} \beta_i \mathbf{b}_i \quad (1)$$

where $\boldsymbol{\beta} = [\beta_1, \dots, \beta_K]^T$, and the $\mathbf{b}_i \in \mathbb{R}^{3M}$ represents linear shape basis. Let $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_K] \in \mathbb{R}^{3M \times K}$ be the matrix of all such shape basis, which will be shared in the same category.

3.2.2 Joint Function

Given the rest-state object in canonical space, we can define joint parameters $\boldsymbol{\phi}' = (\mathbf{u}', \mathbf{q}')$, $\mathbf{u}' \in \mathbb{R}^3, \mathbf{q}' \in \mathbb{R}^3$ for each joint. As the corresponding joint direction \mathbf{u}' is aligned in the canonical space, and the pivot point \mathbf{q}' for each joint are strongly correlated to the shape geometry, we propose to predict joint parameters in canonical space from shape parameters $\boldsymbol{\beta}$. All the joint parameters $\boldsymbol{\Phi}' = \{\boldsymbol{\phi}'_1, \dots, \boldsymbol{\phi}'_K\}$ of the K joints can be generated by joint function $J(\cdot)$:

$$\boldsymbol{\Phi}' = J(\boldsymbol{\beta}; \boldsymbol{\Gamma}) \quad (2)$$

where $\boldsymbol{\Gamma}$ is the parameters of the joint function $J(\cdot)$. Since the relationship between shape parameters and joint parameters in canonical space are highly non-linear, we design a two-layer MLP with ReLU activation as $J(\cdot)$.

3.3 Deformation function

To combine both shape geometric deformation and pose deformation, we introduce a deformation function $\mathbf{P} = W(\mathbf{P}', \boldsymbol{\Phi}', \boldsymbol{\Theta})$, which can transform the sparse shape \mathbf{P}' in zero-centered canonical space with joint parameters $\boldsymbol{\Phi}'$ according to joint states $\boldsymbol{\Theta}$ to the observed sparse shape in the camera space \mathbf{P} . Each keypoint \mathbf{p}'_i in \mathbf{P}' that belongs to part k is transformed into \mathbf{p}_i in \mathbf{P} according to Eq. 3 and Eq. 4:

$$\bar{\mathbf{p}}_i = G_k(\boldsymbol{\Phi}', \boldsymbol{\Theta}) \bar{\mathbf{p}}'_i \quad (3)$$

$$G_k(\boldsymbol{\Phi}', \boldsymbol{\Theta}) = \prod_{j \in A(k)} F_j(\boldsymbol{\Phi}'_j, \boldsymbol{\Theta}_j) \quad (4)$$

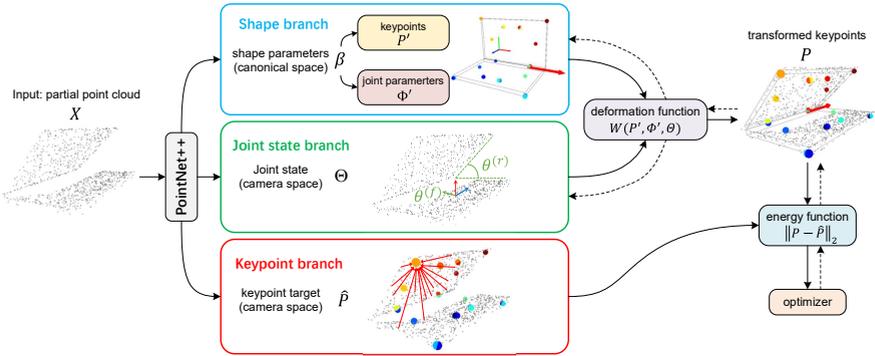


Figure 4: Overall pipeline of our OMADNet. Given a partial point cloud of an observed articulated object, the point features extracted by PointNet++ [20] will be fed into three branches: (1) shape branch (2) joint state branch (3) keypoint branch. We also devise an optimizer to refine final results.

where \bar{p} means the homogeneous form of p , $G_k(\Phi', \Theta)$ is the 4×4 transformation matrix of part k . Finally, $A(k)$ is the ordered set of all the ancestors of part k in the kinematic tree, and $F_j(\Phi'_j, \Theta_j)$ is the relative 4×4 transformation matrix caused by the reference joint of part j . Please refer to the supplementary materials for the details of $F_j(\Phi'_j, \Theta_j)$.

3.4 Learning the OMAD Prior

The parameters of shape function $S(\beta; \mathbf{B})$ and joint function $J(\beta; \Gamma)$ are category-specific, and we call these parameters $\{\mathbf{B}, \Gamma\}$ as **OMAD prior**. We design a network called OMAD-PriorNet to learn OMAD prior for each category in an unsupervised manner. The parameters $\{\mathbf{B}, \Gamma\}$ of shape function $S(\beta; \mathbf{B})$ and joint function $J(\beta; \Gamma)$ will be treated as shared network parameters of OMAD-PriorNet for each category, and they will be updated during training process. After the training, these parameters will be fixed as category-specific **OMAD prior**, then we will predict shape parameters β and keypoints P' for each instance in the training set with the pre-trained OMAD-PriorNet which will be used as ground truth in the training of OMADNet described in Sec. 4.1. Please refer to the supplementary materials for all the loss functions and network structure of OMAD-PriorNet.

Learning of Shape Function. To learn the shape basis \mathbf{B} for shape function $S(\cdot)$, we adopt a similar design in [20]. We make some modifications on the loss functions to adapt to the articulated objects. The tweaks on loss functions are listed in the following: (1) We add an additional L_2 regularization loss on β which can improve the generalization ability. (2) The original chamfer loss, coverage loss and surface loss is changed to part-level. (3) We add a separation loss to ensure the keypoints not collapsed into a cluster.

Learning of Joint Function. The parameters Γ of joint function $J(\cdot)$ is optimized during training according to the proposed **joint loss** defined in Eq. 5.

$$L_{\text{joint}} = L_{\mu} + \lambda L_q \quad (5)$$

where L_{μ} is a cosine-similarity loss that supervises joint direction and L_q is the L_2 loss between the predicted joint pivot point and the ground-truth joint pivot point.

4 OMADNet

With the OMAD representation and the pre-learned OMAD priors, given a point cloud $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$ reprojected from a depth image, we propose a OMADNet to predict the shape parameter $\boldsymbol{\beta}$ and joint states Θ . The overall pipeline is shown in Fig. 4.

4.1 Coarse Prediction of Shape Parameters and Joint States

Feature Extractor. We adopt PointNet++ [40] to extract d -dimensional dense point-wise feature $\mathcal{F}_{\text{dense}} \in \mathbb{R}^{N \times d}$ from \mathbf{X} . Then we can generate global feature $\mathcal{F}_{\text{global}} \in \mathbb{R}^d$ using a max pooling layer with the extracted dense feature.

The Shape Branch. The shape branch learns to predict pose-invariant shape parameters $\boldsymbol{\beta}$ in canonical space. It uses a two layer MLP with global feature $\mathcal{F}_{\text{global}}$ as input to regress a vector as shape parameters $\boldsymbol{\beta}$. We supervise $\boldsymbol{\beta}$ by Eq. 6:

$$L_{\boldsymbol{\beta}} = \|S(\boldsymbol{\beta}) - S(\boldsymbol{\beta}^*)\|_2 + L_{\text{joint}}(J(\boldsymbol{\beta}), J(\boldsymbol{\beta}^*)), \quad (6)$$

where $\boldsymbol{\beta}^*$ is obtained by the pre-trained OMAD-PriorNet described in Sec. 3.4, and L_{joint} is defined in Eq. 5. $L_{\boldsymbol{\beta}}$ indirectly supervises $\boldsymbol{\beta}$ by generating keypoints $\mathbf{P}' = S(\boldsymbol{\beta}; \mathbf{B})$ and joint parameters $\boldsymbol{\Phi}' = J(\boldsymbol{\beta}; \boldsymbol{\Gamma})$ in canonical space. The parameters $\{\mathbf{B}, \boldsymbol{\Gamma}\}$ are obtained from pre-trained OMAD-PriorNet and are fixed during training.

The Joint State Branch. This branch predicts joint states using a two layer MLP with global feature $\mathcal{F}_{\text{global}}$ as input to regress joint states Θ . As there are three joint types in consideration, we treat them separately. For the free joint, $\boldsymbol{\theta}_{\text{base}} = \{\mathbf{R}_{\text{base}}, \mathbf{t}_{\text{base}}\}$, we directly supervise \mathbf{R}_{base} , which is represented by a quaternion vector, via a cosine-similarity loss $\mathcal{L}_{\text{base}(R)}$ while the loss $\mathcal{L}_{\text{base}(t)}$ for \mathbf{t}_{base} is a Root Mean Square (RMS) loss. For the 1D revolute (\mathcal{L}_r) or prismatic (\mathcal{L}_p) joint, we use L_2 loss on $\boldsymbol{\theta}^{(r)}$ or $\boldsymbol{\theta}^{(p)}$. The overall loss for joint state is:

$$L_{\Theta} = \sum_{j=1}^K \mathbb{1}_j^{(\text{free})} (\mathcal{L}_{\text{base}(R)} + \mathcal{L}_{\text{base}(t)}) + \mathbb{1}_j^{(r)} \mathcal{L}_r + \mathbb{1}_j^{(p)} \mathcal{L}_p, \quad (7)$$

where $\mathbb{1}_j^{(\cdot)}$ indicates whether it is the joint type for the joint j .

The Keypoint Branch. The keypoint branch will directly predict ordered keypoint targets $\hat{\mathbf{P}} \in \mathbb{R}^{M \times 3}$ in camera space using dense feature $\mathcal{F}_{\text{dense}}$. These keypoints are for the optimization process described later in Sec. 4.2. Similar to PVN3D [44], we predict $\hat{\mathbf{P}}$ with an offset-voting mechanism, that is every point will regress all the M keypoint offsets with respect to its own location. These offsets are denoted by $\mathcal{O} \in \mathbb{R}^{N \times M}$. However, it is hard to directly regress keypoint offsets in camera space for articulated objects due to large joint deformations and self-occlusion. So we additionally predict an attention map \mathcal{S} of input point cloud \mathbf{X} to focus areas around keypoints. Specifically, the j -th keypoint $\hat{\mathbf{p}}_j \in \hat{\mathbf{P}}$ is generated by Eq. 8:

$$\hat{\mathbf{p}}_j = \frac{1}{N} \sum_{i=1}^N s_{i,j} (\mathbf{x}_i + \mathbf{o}_{i,j}) \quad (8)$$

where $s_{i,j} \in [0, 1]$ is the attention score of i -th point for j -th keypoint, $\mathbf{o}_{i,k} \in \mathcal{O}$ is the 3D offset of j -th keypoint repective to the i -th point, $\mathbf{x}_i \in \mathbf{X}$ is the 3D coordinates of i -th input point. The prediction of $\hat{\mathbf{P}}$ is supervised by:

$$L_{\text{kp}} = \|\hat{\mathbf{P}} - W(S(\boldsymbol{\beta}^*; \mathbf{B}), J(\boldsymbol{\beta}^*; \boldsymbol{\Gamma}), \Theta^*)\|_2 \quad (9)$$

where Θ^* is the ground truth of the joint state, the rest of deformation function $W(\cdot)$ is obtained from pre-trained OMAD-PriorNet in Sec. 3.4.

4.2 Optimization-based Estimator

Since the regression of neural network is known to be noisy, empirically the estimated high-level properties like shape parameters β and joint states Θ are less accurate than the predicted ordered keypoint targets \hat{P} in camera space. Thus, to push the limits of the prediction results in inference, we propose to refine β and Θ with an energy function E defined in Eq. 10:

$$\begin{aligned} E &= \arg \min_{\beta, \Theta} \|W(S(\beta; \mathbf{B}), J(\beta; \Gamma), \Theta) - \hat{P}\|_2 \\ &= \arg \min_{\beta, \Theta} \|\mathbf{P} - \hat{P}\|_2 \end{aligned} \quad (10)$$

where \mathbf{P} is the camera-space keypoints generated by deformation function $W(\cdot)$. The energy function E is differentiable, so we can try to find optimal shape parameters β and joint states Θ to fit the predicted keypoint targets \hat{P} during the optimization process. The initial values of β and Θ are obtained from OMADNet prediction, which vastly reduces the optimization time. Please refer to the supplementary files for further details.

5 Experiments

5.1 Dataset

To the best of our knowledge, there is no public image dataset for articulated object tasks in category level. So we create a synthetic dataset named **ArtImage** to evaluate OMADNet. We select 5 categories of object models from PartNet-Mobility dataset[28] and render depth images under random joint states, object positions and viewpoints for each category with Unity[1]. We will make our dataset public in the future. Please refer to the supplementary materials for further details of ArtImage.

5.2 Experiments on Category-level Pose Estimation

Metrics. We use the following metrics to evaluate the performance of our algorithm on category-level pose estimation. (1) **joint state.** All the joint states error are evaluated in camera space, including mean angle error for revolute joints (max error 90° for missing ones) and mean distance error for prismatic joints (max error 1 for missing ones). (2) **joint parameters.** All the joint parameters are evaluated in camera space. we use mean angle error to evaluate the orientation of joint axis, and use mean distance error to evaluate the distance between two joint axes.

Main Results. Table 1 shows the performance of different methods on category-level pose estimation. We can see that OMAD(refine) surpasses OMAD(initial) by a large margin, which proves the effectiveness of our optimization-based estimator. Compared to state-of-the-art method A-NCSH[15], our OMAD(refine) achieves better results on joint state error and joint distance error on most categories. It is worth noting that the prediction of A-NCSH[15] heavily relies on part segmentation, which could result in missing parts or joints when segmentation fails. While our method could utilize kinematic prior information to

Table 1: Performance comparison on category-level pose estimation task on ArtImage. *OMAD(initial)* represents the initial joint state and joint parameters predicted by OMAD-Net, and *OMAD(refine)* represents the refined results with our optimizer.

Category	Method	Joint State		Joint Parameter		Inference Time per Image ↓
		error ↓		angle error ↓	distance error ↓	
Laptop	A-NCSH [14]	3.5°		1.7°	0.09	9.0s
	OMAD(initial)	5.9°		19.5°	0.14	0.4s
	OMAD(refine)	3.3°		3.7°	0.03	1.6s
Eyeglasses	A-NCSH [14]	12.8°, 14.2°		3.1°, 3.1°	0.07, 0.06	11.9s
	OMAD(initial)	19.1°, 19.4°		17.3°, 17.3°	0.18, 0.17	0.7s
	OMAD(refine)	4.9°, 5.2°		4.2°, 4.6°	0.05, 0.04	2.5s
Dishwasher	A-NCSH [14]	3.8°		6.1°	0.11	5.5s
	OMAD(initial)	17.6°		15.7°	0.17	0.6s
	OMAD(refine)	3.7°		4.6°	0.09	1.6s
Scissors	A-NCSH [14]	4.4°		0.8°	0.04	6.5s
	OMAD(initial)	5.8°		18.7°	0.14	0.7s
	OMAD(refine)	3.2°		2.7°	0.05	1.7s
Drawer	A-NCSH [14]	0.38, 0.45, 0.41		2.6°, 2.7°, 5.2	-	16.5s
	OMAD(initial)	0.18, 0.17, 0.18		15.2°, 15.2°, 15.2°	-	1.0s
	OMAD(refine)	0.11, 0.11, 0.09		3.2°, 3.2°, 3.2°	-	1.9s

prevent missing joints. Besides, our method is much faster than A-NCSH because we adopt sparse keypoints for optimization. We show some qualitative results of our OMAD(refine) in Fig. 5. We can see that the predicted keypoints after optimization can achieve semantic consistency in category, and maintain object structure under heavy self-occlusion.

Ablation Study. Table 2 shows some ablation experiment results. We use optimization-based estimator as default setting in all the ablation experiments. We can see from Table 2 that the use of attention score vastly boosts the performance on all metrics when predicting keypoint targets. We also analyze the upper bound of our method in Table 2. The overall performance improves a lot when we replace keypoint targets \hat{P} predicted by keypoint branch with ground-truth keypoints during optimization process. This indicates that our OMAD representation has great potential and the bottleneck of our method is the quality of predicted keypoint locations in camera space. Please see supplementary files for ablation study on keypoint number, shape parameters β and experiment results on real images.

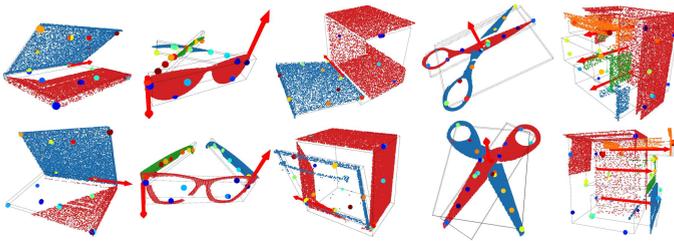


Figure 5: Qualitative results of OMADNet on **unseen** instances for category-level object pose estimation. Keypoint indexes are represented by colors for each category. Ground-truth part segmentation are used for a better view.

5.3 Experiments on Articulated Object Retrieval

Metrics. The pose-invariant shape parameters predicted by OMADNet can be used as shape signatures for articulated object retrieval task. We use the Euclidean distance of shape parameters to estimate the similarity between object instances. We randomly choose 100 depth images as test query set for each category. We report the mAP (mean Average Precision) of top-10 instances on five categories in the test dataset.

Table 2: The effects of using attention score and ground-truth(GT) keypoint targets in OMADNet. Optimization-based estimator are used as default setting.

Category	Ablation		Joint State	Joint Parameter	
	attention score	GT keypoints	error ↓	angle error ↓	distance error ↓
Laptop	✓	✓	4.7°	5.4°	0.04
			3.3°	3.7°	0.03
			1.1°	0.8°	0.01
Eyeglasses	✓	✓	9.8°, 11.2°	9.2°, 9.2°	0.10, 0.09
			4.9°, 5.2°	4.2°, 4.6°	0.05, 0.04
			1.1°, 1.2°	1.2°, 2.1°	0.02, 0.01
Dishwasher	✓	✓	5.0°	5.8°	0.10
			3.7°	4.6°	0.09
			0.4°	0.4°	0.02
Scissors	✓	✓	3.9°	3.9°	0.05
			3.2°	2.7°	0.05
			0.4°	1.8°	0.01
Drawer	✓	✓	0.173, 0.187, 0.181	15.2°, 15.2°, 15.2°	-
			0.110, 0.114, 0.093	3.2°, 3.2°, 3.2°	-
			0.001, 0.001, 0.003	0.5°, 0.5°, 0.5°	-

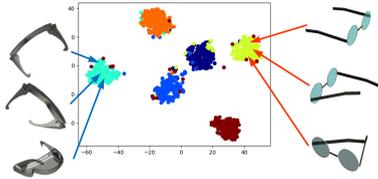


Figure 6: The t-SNE[26] visualization of predicted shape parameters of **unseen** eyeglasses instances under different poses on test dataset of ArtImage. Different instances are distinguished by colors.

Results. To the best of our knowledge, there is no available retrieval method specially designed for our setting. Inspired from [5], we design a simple but effective Siamese Network as baseline which uses PointNet++[20] backbone and contrastive loss[9] for training. Please see supplementary files for further details of the Siamese Network. Table 3 shows some quantitative results. Although our method are not specially designed for retrieval task, we still achieve better performance compared to the baseline on most categories. We find that the Siamese Network can not distinguish some similar instances in drawer category, which vastly harms the performance, while our method can distinguish them very well. Fig. 6 shows t-SNE[26] visualization of shape parameters predicted by OMADNet. We can see that the predicted shape parameters can distinguish different instances well and maintain pose-invariant.

Table 3: Articulated object retrieval results on ArtImage.

Method	mAP					Mean
	Laptop	Eyeglasses	Dishwasher	Scissors	Drawer	
OMAD(Ours)	45.2	93.5	77.7	98.5	96.7	82.3
Siamese Network	43.9	96.2	75.7	96.8	73.6	77.2

6 Conclusion

In this work, we propose a novel category-level representation, OMAD, for articulated object by modeling the two kinds of deformations namely the shape geometric deformation and pose deformation. The OMAD can describe articulated objects with shape parameters and joint states. Then for a naturally observed object, we propose to predict such OMAD information via OMADNet. The predicted OMAD information can be used for multiple tasks, such as category-level object pose estimation and articulated object retrieval.

References

- [1] Unity game engine. <http://www.unity.com/>.
- [2] Alexander Agathos, Ioannis Pratikakis, Panagiotis Papadakis, Stavros Perantonis, Philip Azariadis, and Nickolas S Sapidis. 3d articulated object retrieval using a graph-based representation. *The Visual Computer*, 26(10):1301–1319, 2010.
- [3] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. 2008.
- [4] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 690–696. IEEE, 2000.
- [5] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.
- [6] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins. Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects. *Science Robotics*, 4(30), 2019.
- [7] Clara Fernandez-Labrador, Ajad Chhatkuli, Danda Paudel, Jose Guerrero, Cédric Demonceaux, and Luc Gool. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *16TH EUROPEAN CONFERENCE ON COMPUTER VISION, ECCV 2020*. Springer, 2020.
- [8] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. Pose-oblivious shape signature. *IEEE transactions on visualization and computer graphics*, 13(2):261–271, 2007.
- [9] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.
- [10] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020.
- [11] Ajinkya Jain, Rudolf Lioutikov, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. *arXiv preprint arXiv:2008.10518*, 2020.
- [12] Varun Jain and Hao Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398–407, 2007.
- [13] Duck Hoon Kim, In Kyu Park, Il Dong Yun, and Sang Uk Lee. A new mpeg-7 standard: Perceptual 3-d shape descriptor. In *Pacific-Rim Conference on Multimedia*, pages 238–245. Springer, 2004.

- [14] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1558–1567, 2019.
- [15] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020.
- [16] Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088*, 2020.
- [17] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.
- [18] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiao-long Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv:2104.07645*, 2021.
- [19] Panagiotis Papadakis, Ioannis Pratikakis, Theoharis Theoharis, Georgios Passalis, and Stavros Perantonis. 3d object retrieval using an efficient and compact hybrid shape descriptor. In *Eurographics Workshop on 3D object retrieval*, 2008.
- [20] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.
- [21] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6): 1–17, 2017.
- [22] Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, Sylvain Bouix, and Sven Dickinson. Retrieving articulated 3-d models using medial surfaces. *Machine vision and applications*, 19(4):261–275, 2008.
- [23] Tutian Tang, Wenqiang Xu, Ruolin Ye, Lixin Yang, and Cewu Lu. Learning universal shape dictionary for realtime instance segmentation. *arXiv preprint arXiv:2012.01050*, 2020.
- [24] Catherine Taylor, Robin McNicholas, and Darren Cosker. Transporting real world rigid and articulated objects into egocentric vr experiences. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 623–624. IEEE, 2020.
- [25] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892, 2008.
- [26] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

- [27] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.
- [28] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020.
- [29] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*, 2018.